

Web Service: RfemModel

Web Service: RfemModel

Target Namespace: <http://www.dlubal.com>

Port **RfemModelPort** [Port type](#) [Source code](#)

Location: <http://localhost:8082>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [begin_modification](#) [Detail](#) [Source code](#)
2. [calculate_all](#) [Detail](#) [Source code](#)
3. [cancel_modification](#) [Detail](#) [Source code](#)
4. [delete_all](#) [Detail](#) [Source code](#)
5. [delete_all_results](#) [Detail](#) [Source code](#)
6. [delete_object](#) [Detail](#) [Source code](#)
7. [export_details_of_design_to_csv](#) [Detail](#) [Source code](#)
8. [export_result_tables_to_csv](#) [Detail](#) [Source code](#)
9. [export_result_tables_to_xml](#) [Detail](#) [Source code](#)
10. [export_result_tables_with_detailed_members_results_to_csv](#) [Detail](#) [Source code](#)
11. [export_result_tables_with_detailed_members_results_to_xml](#) [Detail](#) [Source code](#)
12. [finish_modification](#) [Detail](#) [Source code](#)
13. [generate_and_validate_xml_solver_input](#) [Detail](#) [Source code](#)
14. [generate_load_cases_and_combinations](#) [Detail](#) [Source code](#)
15. [generate_parts_lists](#) [Detail](#) [Source code](#)
16. [get_action](#) [Detail](#) [Source code](#)
17. [get_action_combination](#) [Detail](#) [Source code](#)
18. [get_addon_statuses](#) [Detail](#) [Source code](#)
19. [get_all_object_numbers](#) [Detail](#) [Source code](#)
20. [get_building_story](#) [Detail](#) [Source code](#)
21. [get_clipping_box](#) [Detail](#) [Source code](#)
22. [get_clipping_plane](#) [Detail](#) [Source code](#)
23. [get_combination_wizard](#) [Detail](#) [Source code](#)
24. [get_coordinate_system](#) [Detail](#) [Source code](#)
25. [get_cutting_line_setting](#) [Detail](#) [Source code](#)
26. [get_cutting_pattern](#) [Detail](#) [Source code](#)
27. [get_design_overview](#) [Detail](#) [Source code](#)
28. [get_design_situation](#) [Detail](#) [Source code](#)
29. [get_design_support](#) [Detail](#) [Source code](#)
30. [get_dimension](#) [Detail](#) [Source code](#)
31. [get_first_free_number](#) [Detail](#) [Source code](#)
32. [get_formula](#) [Detail](#) [Source code](#)
33. [get_free_circular_load](#) [Detail](#) [Source code](#)
34. [get_free_concentrated_load](#) [Detail](#) [Source code](#)
35. [get_free_line_load](#) [Detail](#) [Source code](#)
36. [get_free_polygon_load](#) [Detail](#) [Source code](#)
37. [get_free_rectangular_load](#) [Detail](#) [Source code](#)
38. [get_global_parameter](#) [Detail](#) [Source code](#)
39. [get_imperfection_case](#) [Detail](#) [Source code](#)
40. [get_imposed_line_deformation](#) [Detail](#) [Source code](#)
41. [get_imposed_nodal_deformation](#) [Detail](#) [Source code](#)
42. [get_intersection](#) [Detail](#) [Source code](#)
43. [get_line](#) [Detail](#) [Source code](#)
44. [get_line_grid](#) [Detail](#) [Source code](#)
45. [get_line_hinge](#) [Detail](#) [Source code](#)
46. [get_line_load](#) [Detail](#) [Source code](#)
47. [get_line_mesh_refinement](#) [Detail](#) [Source code](#)
48. [get_line_set](#) [Detail](#) [Source code](#)
49. [get_line_set_load](#) [Detail](#) [Source code](#)
50. [get_line_support](#) [Detail](#) [Source code](#)
51. [get_line_welded_joint](#) [Detail](#) [Source code](#)
52. [get_list_of_parameters_formula_allowed_for](#) [Detail](#) [Source code](#)
53. [get_load_case](#) [Detail](#) [Source code](#)

54. [get_load_cases_and_combinations](#) [Detail](#) [Source code](#)

55. [get_load_combination](#) [Detail](#) [Source code](#)

56. [get_material](#) [Detail](#) [Source code](#)

57. [get_member](#) [Detail](#) [Source code](#)

58. [get_member_definable_stiffness](#) [Detail](#) [Source code](#)

59. [get_member_eccentricity](#) [Detail](#) [Source code](#)

60. [get_member_hinge](#) [Detail](#) [Source code](#)

61. [get_member_imperfection](#) [Detail](#) [Source code](#)

62. [get_member_load](#) [Detail](#) [Source code](#)

63. [get_member_nonlinearity](#) [Detail](#) [Source code](#)

64. [get_member_representative](#) [Detail](#) [Source code](#)

65. [get_member_result_intermediate_point](#) [Detail](#) [Source code](#)

66. [get_member_set](#) [Detail](#) [Source code](#)

67. [get_member_set_imperfection](#) [Detail](#) [Source code](#)

68. [get_member_set_load](#) [Detail](#) [Source code](#)

69. [get_member_set_representative](#) [Detail](#) [Source code](#)

70. [get_member_stiffness_modification](#) [Detail](#) [Source code](#)

71. [get_member_support](#) [Detail](#) [Source code](#)

72. [get_member_transverse_stiffener](#) [Detail](#) [Source code](#)

73. [get_mesh_settings](#) [Detail](#) [Source code](#)

74. [get_model_info](#) [Detail](#) [Source code](#)

75. [get_model_main_parameters](#) [Detail](#) [Source code](#)

76. [get_model_type](#) [Detail](#) [Source code](#)

77. [get_nodal_load](#) [Detail](#) [Source code](#)

78. [get_nodal_mesh_refinement](#) [Detail](#) [Source code](#)

79. [get_nodal_support](#) [Detail](#) [Source code](#)

80. [get_node](#) [Detail](#) [Source code](#)

81. [get_note](#) [Detail](#) [Source code](#)

82. [get_nth_object_number](#) [Detail](#) [Source code](#)

83. [get_object_count](#) [Detail](#) [Source code](#)

84. [get_object_information](#) [Detail](#) [Source code](#)

85. [get_object_snap](#) [Detail](#) [Source code](#)

86. [get_opening](#) [Detail](#) [Source code](#)

87. [get_opening_load](#) [Detail](#) [Source code](#)

88. [get_optimization_settings](#) [Detail](#) [Source code](#)

89. [get_parts_list_all_by_material](#) [Detail](#) [Source code](#)

90. [get_parts_list_member_representatives_by_material](#) [Detail](#) [Source code](#)

91. [get_parts_list_member_set_representatives_by_material](#) [Detail](#) [Source code](#)

92. [get_parts_list_member_sets_by_material](#) [Detail](#) [Source code](#)

93. [get_parts_list_members_by_material](#) [Detail](#) [Source code](#)

94. [get_parts_list_solids_by_material](#) [Detail](#) [Source code](#)

95. [get_parts_list_surfaces_by_material](#) [Detail](#) [Source code](#)

96. [get_result_combination](#) [Detail](#) [Source code](#)

97. [get_result_section](#) [Detail](#) [Source code](#)

98. [get_rigid_link](#) [Detail](#) [Source code](#)

99. [get_section](#) [Detail](#) [Source code](#)

100. [get_soil_massif](#) [Detail](#) [Source code](#)

101. [get_soil_sample](#) [Detail](#) [Source code](#)

102. [get_solid](#) [Detail](#) [Source code](#)

103. [get_solid_contacts](#) [Detail](#) [Source code](#)

104. [get_solid_gas](#) [Detail](#) [Source code](#)

105. [get_solid_load](#) [Detail](#) [Source code](#)

106. [get_solid_mesh_refinement](#) [Detail](#) [Source code](#)

107. [get_solid_set](#) [Detail](#) [Source code](#)

108. [get_solid_set_load](#) [Detail](#) [Source code](#)

109. [get_spectral_analysis_settings](#) [Detail](#) [Source code](#)

110. [get_static_analysis_settings](#) [Detail](#) [Source code](#)

111. [get_structure_modification](#) [Detail](#) [Source code](#)

112. [get_surface](#) [Detail](#) [Source code](#)

113. [get_surface_eccentricity](#) [Detail](#) [Source code](#)

114. [get_surface_imperfection](#) [Detail](#) [Source code](#)

115. [get_surface_load](#) [Detail](#) [Source code](#)

116. [get_surface_mesh_refinement](#) [Detail](#) [Source code](#)

117. [get_surface_results_adjustment](#) [Detail](#) [Source code](#)

118. [get_surface_set](#) [Detail](#) [Source code](#)

119. [get_surface_set_imperfection](#) [Detail](#) [Source code](#)

120. [get_surface_set_load](#) [Detail](#) [Source code](#)

121. [get_surface_stiffness_modification](#) [Detail](#) [Source code](#)

122. [get_surface_support](#) [Detail](#) [Source code](#)

123. [get_surfaces_contact](#) [Detail](#) [Source code](#)

124. [get_surfaces_contact_type](#) [Detail](#) [Source code](#)

125. [get_terrain](#) [Detail](#) [Source code](#)

126. [get_thickness](#) [Detail](#) [Source code](#)

127. [get_visual_object](#) [Detail](#) [Source code](#)

128. [get_wind_profile](#) [Detail](#) [Source code](#)

129. [get_wind_simulation](#) [Detail](#) [Source code](#)

130. [get_wind_simulation_analysis_settings](#) [Detail](#) [Source code](#)

131. [plausibility_check](#) [Detail](#) [Source code](#)

132. [reset](#) [Detail](#) [Source code](#)

133. [run_script](#) [Detail](#) [Source code](#)

134. [save](#) [Detail](#) [Source code](#)

135. [set_action](#) [Detail](#) [Source code](#)

136. [set_action_combination](#) [Detail](#) [Source code](#)

137. [set_addon_statuses](#) [Detail](#) [Source code](#)

138. [set_building_story](#) [Detail](#) [Source code](#)

139. [set_clipping_box](#) [Detail](#) [Source code](#)

140. [set_clipping_plane](#) [Detail](#) [Source code](#)

141. [set_combination_wizard](#) [Detail](#) [Source code](#)

142. [set_coordinate_system](#) [Detail](#) [Source code](#)

143. [set_cutting_line_setting](#) [Detail](#) [Source code](#)

144. [set_cutting_pattern](#) [Detail](#) [Source code](#)

145. [set_design_situation](#) [Detail](#) [Source code](#)

146. [set_design_support](#) [Detail](#) [Source code](#)

147. [set_dimension](#) [Detail](#) [Source code](#)

148. [set_formula](#) [Detail](#) [Source code](#)

149. [set_free_circular_load](#) [Detail](#) [Source code](#)

150. [set_free_concentrated_load](#) [Detail](#) [Source code](#)

151. [set_free_line_load](#) [Detail](#) [Source code](#)

152. [set_free_polygon_load](#) [Detail](#) [Source code](#)

153. [set_free_rectangular_load](#) [Detail](#) [Source code](#)

154. [set_global_parameter](#) [Detail](#) [Source code](#)

155. [set_imperfection_case](#) [Detail](#) [Source code](#)

156. [set_imposed_line_deformation](#) [Detail](#) [Source code](#)

157. [set_imposed_nodal_deformation](#) [Detail](#) [Source code](#)

158. [set_intersection](#) [Detail](#) [Source code](#)

159. [set_line](#) [Detail](#) [Source code](#)

160. [set_line_grid](#) [Detail](#) [Source code](#)

161. [set_line_hinge](#) [Detail](#) [Source code](#)

162. [set_line_load](#) [Detail](#) [Source code](#)

163. [set_line_mesh_refinement](#) [Detail](#) [Source code](#)

164. [set_line_set](#) [Detail](#) [Source code](#)

165. [set_line_set_load](#) [Detail](#) [Source code](#)

166. [set_line_support](#) [Detail](#) [Source code](#)

167. [set_line_welded_joint](#) [Detail](#) [Source code](#)

168. [set_load_case](#) [Detail](#) [Source code](#)

169. [set_load_cases_and_combinations](#) [Detail](#) [Source code](#)

170. [set_load_combination](#) [Detail](#) [Source code](#)

171. [set_material](#) [Detail](#) [Source code](#)

172. [set_member](#) [Detail](#) [Source code](#)

173. [set_member_definable_stiffness](#) [Detail](#) [Source code](#)

174. [set_member_eccentricity](#) [Detail](#) [Source code](#)

175. [set_member_hinge](#) [Detail](#) [Source code](#)

176. [set_member_imperfection](#) [Detail](#) [Source code](#)

177. [set_member_load](#) [Detail](#) [Source code](#)

178. [set_member_nonlinearity](#) [Detail](#) [Source code](#)

179. [set_member_representative](#) [Detail](#) [Source code](#)

180. [set_member_result_intermediate_point](#) [Detail](#) [Source code](#)

181. [set_member_set](#) [Detail](#) [Source code](#)

182. [set_member_set_imperfection](#) [Detail](#) [Source code](#)

183. [set_member_set_load](#) [Detail](#) [Source code](#)

184. [set_member_set_representative](#) [Detail](#) [Source code](#)
185. [set_member_stiffness_modification](#) [Detail](#) [Source code](#)
186. [set_member_support](#) [Detail](#) [Source code](#)
187. [set_member_transverse_stiffener](#) [Detail](#) [Source code](#)
188. [set_mesh_settings](#) [Detail](#) [Source code](#)
189. [set_model_type](#) [Detail](#) [Source code](#)
190. [set_nodal_load](#) [Detail](#) [Source code](#)
191. [set_nodal_mesh_refinement](#) [Detail](#) [Source code](#)
192. [set_nodal_support](#) [Detail](#) [Source code](#)
193. [set_node](#) [Detail](#) [Source code](#)
194. [set_note](#) [Detail](#) [Source code](#)
195. [set_object_snap](#) [Detail](#) [Source code](#)
196. [set_opening](#) [Detail](#) [Source code](#)
197. [set_opening_load](#) [Detail](#) [Source code](#)
198. [set_optimization_settings](#) [Detail](#) [Source code](#)
199. [set_result_combination](#) [Detail](#) [Source code](#)
200. [set_result_section](#) [Detail](#) [Source code](#)
201. [set_rigid_link](#) [Detail](#) [Source code](#)
202. [set_section](#) [Detail](#) [Source code](#)
203. [set_soil_massif](#) [Detail](#) [Source code](#)
204. [set_soil_sample](#) [Detail](#) [Source code](#)
205. [set_solid](#) [Detail](#) [Source code](#)
206. [set_solid_contacts](#) [Detail](#) [Source code](#)
207. [set_solid_gas](#) [Detail](#) [Source code](#)
208. [set_solid_load](#) [Detail](#) [Source code](#)
209. [set_solid_mesh_refinement](#) [Detail](#) [Source code](#)
210. [set_solid_set](#) [Detail](#) [Source code](#)
211. [set_solid_set_load](#) [Detail](#) [Source code](#)
212. [set_spectral_analysis_settings](#) [Detail](#) [Source code](#)
213. [set_static_analysis_settings](#) [Detail](#) [Source code](#)
214. [set_structure_modification](#) [Detail](#) [Source code](#)
215. [set_surface](#) [Detail](#) [Source code](#)
216. [set_surface_eccentricity](#) [Detail](#) [Source code](#)
217. [set_surface_imperfection](#) [Detail](#) [Source code](#)
218. [set_surface_load](#) [Detail](#) [Source code](#)
219. [set_surface_mesh_refinement](#) [Detail](#) [Source code](#)
220. [set_surface_results_adjustment](#) [Detail](#) [Source code](#)
221. [set_surface_set](#) [Detail](#) [Source code](#)
222. [set_surface_set_imperfection](#) [Detail](#) [Source code](#)
223. [set_surface_set_load](#) [Detail](#) [Source code](#)
224. [set_surface_stiffness_modification](#) [Detail](#) [Source code](#)
225. [set_surface_support](#) [Detail](#) [Source code](#)
226. [set_surfaces_contact](#) [Detail](#) [Source code](#)
227. [set_surfaces_contact_type](#) [Detail](#) [Source code](#)
228. [set_terrain](#) [Detail](#) [Source code](#)
229. [set_thickness](#) [Detail](#) [Source code](#)
230. [set_visual_object](#) [Detail](#) [Source code](#)
231. [set_wind_profile](#) [Detail](#) [Source code](#)
232. [set_wind_simulation](#) [Detail](#) [Source code](#)
233. [set_wind_simulation_analysis_settings](#) [Detail](#) [Source code](#)
234. [unite_nodes_and_supports](#) [Detail](#) [Source code](#)

Operations

Port type **IRfemModel** [Source code](#)

1. begin_modification

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/begin_modification

Input: begin_modification_request (soap:body, use = literal) [Source code](#)

parameters type <i>begin_modification</i> <ul style="list-style-type: none">modification_name type <i>string</i>

Output: begin_modification_response (soap:body, use = literal) [Source code](#)

parameters type [begin_modificationResponse](#)

2. calculate_all

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/calculate_all

Input: calculate_all_request (soap:body, use = literal) [Source code](#)

parameters type [calculate_all](#)

- generateXmlSolverInput type *boolean*

Output: calculate_all_response (soap:body, use = literal) [Source code](#)

parameters type [calculate_allResponse](#)

3. cancel_modification

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/cancel_modification

Input: cancel_modification_request (soap:body, use = literal) [Source code](#)

parameters type [cancel_modification](#)

Output: cancel_modification_response (soap:body, use = literal) [Source code](#)

parameters type [cancel_modificationResponse](#)

4. delete_all

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/delete_all

Input: delete_all_request (soap:body, use = literal) [Source code](#)

parameters type [delete_all](#)

Output: delete_all_response (soap:body, use = literal) [Source code](#)

parameters type [delete_allResponse](#)

5. delete_all_results

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/delete_all_results

Input: delete_all_results_request (soap:body, use = literal) [Source code](#)

parameters type [delete_all_results](#)

- delete_mesh - optional; type *boolean*

Output: delete_all_results_response (soap:body, use = literal) [Source code](#)

parameters type [delete_all_resultsResponse](#)

6. delete_object

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/delete_object

Input: delete_object_request (soap:body, use = literal) [Source code](#)

parameters type [delete_object](#)

- type type *object_types* - type *undefined* with restriction - enum { 'E_OBJECT_TYPE_ACTION', 'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX', 'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD', 'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING', 'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION', 'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD', 'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD', 'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD', 'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE', 'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION', 'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID', 'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT', 'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT', 'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION', 'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS', 'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE', 'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD', 'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET', 'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD', 'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER', 'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT', 'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE', 'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD', 'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK', 'E_OBJECT_TYPE_SECTION', 'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE', 'E_OBJECT_TYPE_SOLID', 'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS', 'E_OBJECT_TYPE_SOLID_LOAD', 'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET',

'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS',
'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION',
'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT',
'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY',
'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD',
'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT',
'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION',
'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS',
'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION',
'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
▪ no type *int*
▪ parent_no - optional; type *int*

Output: delete_object_response (soap:body, use = literal) [Source code](#)

parameters type *delete_objectResponse*

7. export_details_of_design_to_csv

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/export_details_of_design_to_csv

Input: export_details_of_design_to_csv_request (soap:body, use = literal) [Source code](#)

parameters type *export_details_of_design_to_csv*
▪ targetDirectoryPath type *string*

Output: export_details_of_design_to_csv_response (soap:body, use = literal) [Source code](#)

parameters type *export_details_of_design_to_csvResponse*

8. export_result_tables_to_csv

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/export_result_tables_to_csv

Input: export_result_tables_to_csv_request (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_to_csv*
▪ target_directory_path type *string*

Output: export_result_tables_to_csv_response (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_to_csvResponse*

9. export_result_tables_to_xml

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/export_result_tables_to_xml

Input: export_result_tables_to_xml_request (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_to_xml*
▪ target_file_path type *string*

Output: export_result_tables_to_xml_response (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_to_xmlResponse*

10. export_result_tables_with_detailed_members_results_to_csv

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/export_result_tables_with_detailed_members_results_to_csv

Input: export_result_tables_with_detailed_members_results_to_csv_request (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_with_detailed_members_results_to_csv*
▪ target_directory_path type *string*

Output: export_result_tables_with_detailed_members_results_to_csv_response (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_with_detailed_members_results_to_csvResponse*

11. export_result_tables_with_detailed_members_results_to_xml

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/export_result_tables_with_detailed_members_results_to_xml

Input: export_result_tables_with_detailed_members_results_to_xml_request (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_with_detailed_members_results_to_xml*
▪ target_file_path type *string*

Output: export_result_tables_with_detailed_members_results_to_xml_response (soap:body, use = literal) [Source code](#)

parameters type *export_result_tables_with_detailed_members_results_to_xmlResponse*

12. finish_modification

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/finish_modification

Input: [finish_modification_request](#) (soap:body, use = literal) [Source code](#)

parameters type [finish_modification](#)

Output: [finish_modification_response](#) (soap:body, use = literal) [Source code](#)

parameters type [finish_modificationResponse](#)

13. [generate_and_validate_xml_solver_input](#)

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/generate_and_validate_xml_solver_input

Input: [generate_and_validate_xml_solver_input_request](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_and_validate_xml_solver_input](#)

- solver_input_file_path type *String*

Output: [generate_and_validate_xml_solver_input_response](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_and_validate_xml_solver_inputResponse](#)

- value type [generate_and_validate_xml_solver_input_result](#)
 - succeeded - optional; type *boolean*
 - result - optional; type *string*
 - plausibility_check_messages - optional; type [plausibility_check_messages](#)
 - message - optional, unbounded; type [plausibility_check_message](#)
 - message_type type [plausibility_check_message_type](#) - type *undefined* with restriction - enum { 'ERROR', 'INFORMATION', 'QUESTION', 'WARNING' }
 - object - optional; type *string*
 - input_field - optional; type *string*
 - current_value - optional; type *string*
 - message - optional; type *string*
 - result - optional; type *boolean*

14. [generate_load_cases_and_combinations](#)

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/generate_load_cases_and_combinations

Input: [generate_load_cases_and_combinations_request](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_load_cases_and_combinations](#)

Output: [generate_load_cases_and_combinations_response](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_load_cases_and_combinationsResponse](#)

15. [generate_parts_lists](#)

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/generate_parts_lists

Input: [generate_parts_lists_request](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_parts_lists](#)

Output: [generate_parts_lists_response](#) (soap:body, use = literal) [Source code](#)

parameters type [generate_parts_listsResponse](#)

16. [get_action](#)

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_action

Input: [get_action_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get_action](#)

- no type *int*

Output: [get_action_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get_actionResponse](#)

- value type [action](#)
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - is_active - optional; type *boolean*
 - action_category - optional; type *string*
 - action_type - optional; type [action_action_type](#) - type *undefined* with restriction - enum { 'ACTING_ALTERNATIVELY', 'ACTING_DIFFERENTLY', 'ACTING_SIMULTANEOUSLY', 'DYNAMIC_LOAD_CASE' }
 - comment - optional; type *string*
 - is_generated - optional; type *boolean*
 - generating_object_info - optional; type *string*
 - items - optional; type *array_of_action_items*
 - action_items - optional, unbounded; type [action_items](#)
 - no - optional; type *int*
 - load_case_item - optional; type *int*
 - acting_group_number - optional; type *int*
 - has_short_duration - optional; type *boolean*
 - has_duration_shorter_than_one_month - optional; type *boolean*
 - imposed_load_category - optional; type [action_imposed_load_category](#) - type *undefined* with restriction - enum { 'IMPOSED_LOADS_CATEGORY_A', 'IMPOSED_LOADS_CATEGORY_B', 'IMPOSED_LOADS_CATEGORY_C',

'IMPOSED_LOADS_CATEGORY_D', 'IMPOSED_LOADS_CATEGORY_E')

- has_short_duration_according_to_5132 - optional; type *boolean*
- for_temperature_apply_coefficients - optional; type *boolean*
- short_time_variable_action - optional; type *boolean*
- has_inclusive_action - optional; type *boolean*
- inclusive_action - optional; type *int*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

17. get_action_combination

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_action_combination

Input: get_action_combination_request (soap:body, use = literal) [Source code](#)

parameters type *get_action_combination*

- no type *int*

Output: get_action_combination_response (soap:body, use = literal) [Source code](#)

parameters type *get_action_combinationResponse*

- value type *action_combination*
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - attribute_always_editable - optional; type *string*
 - comment - optional; type *string*
 - design_situation - optional; type *int*
 - items - optional; type *array_of_action_combination_items*
 - action_combination_items - optional, unbounded; type *action_combination_items*
 - no - optional; type *int*
 - action_item - optional; type *int*
 - operator_type - optional; type *operator_type* - type *undefined* with restriction - enum { 'OPERATOR_AND', 'OPERATOR_NONE', 'OPERATOR_OR' }
 - left_parenthesis - optional; type *boolean*
 - right_parenthesis - optional; type *boolean*
 - group_factor - optional; type *double*
 - action_factor - optional; type *double*
 - action_load_type - optional; type *action_load_type* - type *undefined* with restriction - enum { 'LOAD_TYPE_PERMANENT', 'LOAD_TYPE_TRANSIENT' }
 - group_load_type - optional; type *group_load_type* - type *undefined* with restriction - enum { 'LOAD_TYPE_PERMANENT', 'LOAD_TYPE_TRANSIENT' }
 - action - optional; type *int*
 - is_leading - optional; type *boolean*
 - gamma - optional; type *double*
 - psi - optional; type *double*
 - xi - optional; type *double*
 - k_fi - optional; type *double*
 - c_esl - optional; type *double*
 - k_def - optional; type *double*
 - psi_0 - optional; type *double*
 - psi_1 - optional; type *double*
 - psi_2 - optional; type *double*
 - fi - optional; type *double*
 - gamma_0 - optional; type *double*
 - alfa - optional; type *double*
 - k_f - optional; type *double*
 - phi - optional; type *double*
 - rho - optional; type *double*
 - omega_0 - optional; type *double*
 - gamma_l_1 - optional; type *double*
 - k_creep - optional; type *double*
 - active - optional; type *boolean*
 - construction_stage - optional; type *int*
 - combination_type - optional; type *action_combination_combination_type* - type *undefined* with restriction - enum { 'ENVELOPE_PERMANENT', 'ENVELOPE_TRANSIENT', 'GENERAL', 'SUPERPOSITION' }
 - generated_load_combinations - optional; type *array_of_int*
 - generated_result_combinations - optional; type *array_of_int*
 - is_generated - optional; type *boolean*
 - generating_object_info - optional; type *string*
 - id_for_export_import - optional; type *string*
 - metadata_for_export_import - optional; type *string*

18. get_addon_statuses

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_addon_statuses

Input: get_addon_statuses_request (soap:body, use = literal) [Source code](#)

parameters type *get_addon_statuses*

Output: get_addon_statuses_response (soap:body, use = literal) [Source code](#)

parameters type *get_addon_statusesResponse*

- value type *addon_list_type*
 - design_addons - optional; type *addon_list_design_addons_list_type*
 - stress_analysis_active - optional; type *boolean*
 - concrete_design_active - optional; type *boolean*
 - steel_design_active - optional; type *boolean*
 - timber_design_active - optional; type *boolean*
 - aluminum_design_active - optional; type *boolean*
 - steel_joints_active - optional; type *boolean*

- timber_joints_active - optional; type *boolean*
- craneway_design_active - optional; type *boolean*
- masonry_design_active - optional; type *boolean*
- multilayer_surfaces_design_active - optional; type *boolean*
- analysis - optional; type *addon_list_analysis_list_type*
 - material_nonlinear_analysis_active - optional; type *boolean*
 - structure_stability_active - optional; type *boolean*
 - construction_stages_active - optional; type *boolean*
 - time_dependent_active - optional; type *boolean*
 - form_finding_active - optional; type *boolean*
 - cutting_patterns_active - optional; type *boolean*
 - torsional_warping_active - optional; type *boolean*
 - cost_estimation_active - optional; type *boolean*
- dynamic_analysis_settings - optional; type *addon_list_dynamic_analysis_settings_list_type*
 - modal_active - optional; type *boolean*
 - spectral_active - optional; type *boolean*
 - time_history_active - optional; type *boolean*
 - pushover_active - optional; type *boolean*
 - harmonic_response_active - optional; type *boolean*
- special_solutions - optional; type *addon_list_special_solutions_list_type*
 - building_model_active - optional; type *boolean*
 - wind_simulation_active - optional; type *boolean*
 - geotechnical_analysis_active - optional; type *boolean*

19. get_all_object_numbers

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: `http://localhost:8082/get_all_object_numbers`

Input: `get_all_object_numbers_request` (soap:body, use = literal) [Source code](#)

parameters type *get_all_object_numbers*

- type type *object_types* - type *undefined* with restriction - enum { 'E_OBJECT_TYPE_ACTION', 'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX', 'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD', 'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING', 'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION', 'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD', 'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD', 'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD', 'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE', 'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION', 'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID', 'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT', 'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT', 'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION', 'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS', 'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE', 'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD', 'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET', 'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD', 'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER', 'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT', 'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE', 'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD', 'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK', 'E_OBJECT_TYPE_SECTION', 'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE', 'E_OBJECT_TYPE_SOLID', 'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS', 'E_OBJECT_TYPE_SOLID_LOAD', 'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET', 'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION', 'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT', 'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY', 'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD', 'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT', 'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS', 'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION', 'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
- parent_no - optional; type *int*

Output: `get_all_object_numbers_response` (soap:body, use = literal) [Source code](#)

parameters type *get_all_object_numbersResponse*

- value type *array_of_int*

20. get_building_story

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: `http://localhost:8082/get_building_story`

Input: `get_building_story_request` (soap:body, use = literal) [Source code](#)

parameters type *get_building_story*

- no type *int*

Output: `get_building_story_response` (soap:body, use = literal) [Source code](#)

parameters type *get_building_storyResponse*

- value type *building_story*
 - no type *int*
 - type - optional; type *building_story_type* - type *undefined* with restriction - enum { 'TYPE_STANDARD' }
 - user_defined_name_enabled - optional; type *boolean*

- name - optional; type *string*
- story_no - optional; type *int*
- elevation - optional; type *double*
- bottom_elevation - optional; type *double*
- height - optional; type *double*
- modified_height - optional; type *double*
- thickness - optional; type *double*
- comment - optional; type *string*
- mass - optional; type *double*
- center_of_gravity_x - optional; type *double*
- center_of_gravity_y - optional; type *double*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

21. get_clipping_box

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_clipping_box

Input: get_clipping_box_request (soap:body, use = literal) [Source code](#)

parameters type *get_clipping_box*

- no type *int*

Output: get_clipping_box_response (soap:body, use = literal) [Source code](#)

parameters type *get_clipping_boxResponse*

- value type *clipping_box*
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - coordinate_system - optional; type *int*
 - origin_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
 - origin_coordinate_x - optional; type *double*
 - origin_coordinate_y - optional; type *double*
 - origin_coordinate_z - optional; type *double*
 - dimensions - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
 - dimension_x - optional; type *double*
 - dimension_y - optional; type *double*
 - dimension_z - optional; type *double*
 - comment - optional; type *string*
 - id_for_export_import - optional; type *string*
 - metadata_for_export_import - optional; type *string*

22. get_clipping_plane

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_clipping_plane

Input: get_clipping_plane_request (soap:body, use = literal) [Source code](#)

parameters type *get_clipping_plane*

- no type *int*

Output: get_clipping_plane_response (soap:body, use = literal) [Source code](#)

parameters type *get_clipping_planeResponse*

- value type *clipping_plane*
 - no type *int*
 - type - optional; type *clipping_plane_type* - type *undefined* with restriction - enum { 'TYPE_2_POINTS_AND_ANGLE', 'TYPE_3_POINTS', 'TYPE_OFFSET_XYZ', 'TYPE_POINT_AND_3_ANGLES' }
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - coordinate_system - optional; type *int*
 - invert_clipping_side - optional; type *boolean*
 - origin_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
 - origin_coordinate_x - optional; type *double*
 - origin_coordinate_y - optional; type *double*
 - origin_coordinate_z - optional; type *double*
 - u_axis_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
 - u_axis_point_coordinate_x - optional; type *double*
 - u_axis_point_coordinate_y - optional; type *double*
 - u_axis_point_coordinate_z - optional; type *double*
 - clipping_plane_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
 - clipping_plane_point_coordinate_x - optional; type *double*
 - clipping_plane_point_coordinate_y - optional; type *double*
 - clipping_plane_point_coordinate_z - optional; type *double*
 - clipping_plane_angle - optional; type *double*

- rotation_angles_sequence - optional; type *clipping_plane_rotation_angles_sequence* - type *undefined* with restriction - enum { 'SEQUENCE_UVW', 'SEQUENCE_UWV', 'SEQUENCE_VUW', 'SEQUENCE_VWU', 'SEQUENCE_WUV', 'SEQUENCE_WVU', 'SEQUENCE_XYZ', 'SEQUENCE_XZY', 'SEQUENCE_YXZ', 'SEQUENCE_YZX', 'SEQUENCE_ZXY', 'SEQUENCE_ZYX' }
- rotation_angle_1 - optional; type *double*
- rotation_angle_2 - optional; type *double*
- rotation_angle_3 - optional; type *double*
- orientation - optional; type *clipping_plane_orientation* - type *undefined* with restriction - enum { 'ORIENTATION_PARALLEL_TO_XY', 'ORIENTATION_PARALLEL_TO_XZ', 'ORIENTATION_PARALLEL_TO_YZ' }
- comment - optional; type *string*
- is_generated - optional; type *boolean*
- generating_object_info - optional; type *string*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

23. get_combination_wizard

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_combination_wizard

Input: get_combination_wizard_request (soap:body, use = literal) [Source code](#)

parameters type *get_combination_wizard*

- no type *int*

Output: get_combination_wizard_response (soap:body, use = literal) [Source code](#)

parameters type *get_combination_wizardResponse*

- value type *combination_wizard*
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - static_analysis_settings - optional; type *int*
 - generate_combinations - optional; type *combination_wizard_generate_combinations* - type *undefined* with restriction - enum { 'GENERATE_LOAD_COMBINATIONS', 'GENERATE_RESULT_COMBINATIONS' }
 - has_stability_analysis - optional; type *boolean*
 - stability_analysis_settings - optional; type *int*
 - consider_imperfection_case - optional; type *boolean*
 - generate_same_CO_without_IC - optional; type *boolean*
 - consider_construction_stages - optional; type *boolean*
 - user_defined_action_combinations - optional; type *boolean*
 - favorable_permanent_actions - optional; type *boolean*
 - consider_inclusive_or_exclusive_load_cases - optional; type *boolean*
 - reduce_number_of_generated_combinations - optional; type *boolean*
 - auxiliary_combinations - optional; type *boolean*
 - generate_subcombinations_of_type_superposition - optional; type *boolean*
 - comment - optional; type *string*
 - consider_initial_state - optional; type *boolean*
 - initial_state_case - optional; type *int*
 - initial_state_definition_type - optional; type *combination_wizard_initial_state_definition_type* - type *undefined* with restriction - enum { 'DEFINITION_TYPE_FINAL_STATE', 'DEFINITION_TYPE_STIFFNESS', 'DEFINITION_TYPE_STRAINS', 'DEFINITION_TYPE_STRAINS_WITH_USER_DEFINED_FACTORS' }
 - individual_factors_of_selected_objects_table - optional; type *array_of_combination_wizard_individual_factors_of_selected_objects_table*
 - combination_wizard_individual_factors_of_selected_objects_table - optional, unbounded; type *combination_wizard_individual_factors_of_selected_objects_table*
 - no - optional; type *int*
 - object_type - optional; type *object_type* - type *undefined* with restriction - enum { 'INITIAL_STATE_FACTORS_OBJECT_TYPE_LINE_HINGE', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_LINE_WITH_SUPPORT', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_MEMBER', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_MEMBER_HINGE', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_NODE_WITH_SUPPORT', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_SOLID', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_SURFACE' }
 - object_list - optional; type *array_of_int*
 - strain_type - optional; type *strain_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG_X', 'ALONG_Y', 'ALONG_Z', 'AROUND_X', 'AROUND_Y', 'AROUND_Z' }
 - factor - optional; type *double*
 - comment - optional; type *string*
 - structure_modification_enabled - optional; type *boolean*
 - structure_modification - optional; type *int*
 - id_for_export_import - optional; type *string*
 - metadata_for_export_import - optional; type *string*

24. get_coordinate_system

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_coordinate_system

Input: get_coordinate_system_request (soap:body, use = literal) [Source code](#)

parameters type *get_coordinate_system*

- no type *int*

Output: get_coordinate_system_response (soap:body, use = literal) [Source code](#)

parameters type *get_coordinate_systemResponse*

- value type *coordinate_system*
 - no type *int*
 - type - optional; type *coordinate_system_type* - type *undefined* with restriction - enum { 'TYPE_2_POINTS_AND_ANGLE', 'TYPE_3_POINTS', 'TYPE_GLOBAL_XYZ', 'TYPE_OFFSET_XYZ', 'TYPE_POINT_AND_3_ANGLES' }
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*

- origin_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- origin_coordinate_x - optional; type *double*
- origin_coordinate_y - optional; type *double*
- origin_coordinate_z - optional; type *double*
- u_axis_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- u_axis_point_coordinate_x - optional; type *double*
- u_axis_point_coordinate_y - optional; type *double*
- u_axis_point_coordinate_z - optional; type *double*
- uw_plane_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- uw_plane_point_coordinate_x - optional; type *double*
- uw_plane_point_coordinate_y - optional; type *double*
- uw_plane_point_coordinate_z - optional; type *double*
- uw_plane_angle - optional; type *double*
- rotation_angles_sequence - optional; type *coordinate_system_rotation_angles_sequence* - type *undefined* with restriction - enum { 'SEQUENCE_XYZ', 'SEQUENCE_XZY', 'SEQUENCE_YXZ', 'SEQUENCE_YZX', 'SEQUENCE_ZXY', 'SEQUENCE_ZYX' }
- rotation_angle_1 - optional; type *double*
- rotation_angle_2 - optional; type *double*
- rotation_angle_3 - optional; type *double*
- comment - optional; type *string*
- is_generated - optional; type *boolean*
- generating_object_info - optional; type *string*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

25. get_cutting_line_setting

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_cutting_line_setting

Input: get_cutting_line_setting_request (soap:body, use = literal) [Source code](#)

parameters type *get_cutting_line_setting*

- no type *int*

Output: get_cutting_line_setting_response (soap:body, use = literal) [Source code](#)

parameters type *get_cutting_line_settingResponse*

- value type *cutting_line_setting*
 - no type *int*
 - type - optional; type *cutting_line_setting_type* - type *undefined* with restriction - enum { 'TYPE_BOUNDARY_LINE', 'TYPE_WELDING_LINE' }
 - comment - optional; type *string*
 - cutting_line_compensation - optional; type *double*
 - cutting_line_compensation_enabled - optional; type *boolean*
 - cutting_pattern_allowance - optional; type *double*
 - name - optional; type *string*
 - weld_allowance_assignment - optional; type *cutting_line_setting_weld_allowance_assignment* - type *undefined* with restriction - enum { 'WELD_ALLOWANCE_TO_LEFT_PATTERN', 'WELD_ALLOWANCE_TO_RIGHT_PATTERN', 'WELD_ALLOWANCE_TO_NEIGHBOURING_PATTERNS' }
 - id_for_export_import - optional; type *string*
 - metadata_for_export_import - optional; type *string*

26. get_cutting_pattern

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_cutting_pattern

Input: get_cutting_pattern_request (soap:body, use = literal) [Source code](#)

parameters type *get_cutting_pattern*

- no type *int*

Output: get_cutting_pattern_response (soap:body, use = literal) [Source code](#)

parameters type *get_cutting_patternResponse*

- value type *cutting_pattern*
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - material_change_enabled - optional; type *boolean*
 - material - optional; type *int*
 - pattern_orientation_category - optional; type *cutting_pattern_pattern_orientation_category* - type *undefined* with restriction - enum { 'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_ANGULAR_ROTATION', 'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_COORDINATE_SYSTEM', 'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_LINES' }
 - angular_rotation - optional; type *double*
 - axis - optional; type *cutting_pattern_axis* - type *undefined* with restriction - enum { 'AXIS_X', 'AXIS_Y' }
 - parallel_to_lines - optional; type *array_of_int*
 - coordinate_system - optional; type *int*
 - boundary_lines - optional; type *array_of_int*
 - cutting_line_settings_table - optional; type *array_of_cutting_pattern_cutting_line_settings_table*
 - cutting_pattern_cutting_line_settings_table - optional, unbounded; type *cutting_pattern_cutting_line_settings_table*
 - no - optional; type *int*
 - line - optional; type *int*

- cutting_line_settings - optional; type *int*
- comment - optional; type *string*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

27. get_design_overview

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_design_overview

Input: get_design_overview_request (soap:body, use = literal) [Source code](#)

parameters type *get_design_overview*

Output: get_design_overview_response (soap:body, use = literal) [Source code](#)

parameters type *get_design_overviewResponse*

- value type *array_of_design_overview*
 - design_overview - optional, unbounded; type *design_overview*
 - no - optional; type *int*
 - addon - optional; type *string*
 - object_type - optional; type *string*
 - objects_no - optional; type *array_of_int*
 - location - optional; type *string*
 - construction_stage - optional; type *string*
 - design_situation - optional; type *string*
 - loading - optional; type *string*
 - design_ratio - optional; type *double*
 - design_check_type - optional; type *string*
 - description - optional; type *string*

28. get_design_situation

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_design_situation

Input: get_design_situation_request (soap:body, use = literal) [Source code](#)

parameters type *get_design_situation*

- no type *int*

Output: get_design_situation_response (soap:body, use = literal) [Source code](#)

parameters type *get_design_situationResponse*

- value type *design_situation*
 - no type *int*
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - design_situation_type - optional; type *int*
 - active - optional; type *boolean*
 - comment - optional; type *string*
 - is_generated - optional; type *boolean*
 - generating_object_info - optional; type *string*
 - combination_wizard - optional; type *int*
 - case_objects - optional; type *array_of_design_situation_case_objects*
 - design_situation_case_objects - optional, unbounded; type *int*
 - id_for_export_import - optional; type *string*
 - metadata_for_export_import - optional; type *string*

29. get_design_support

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_design_support

Input: get_design_support_request (soap:body, use = literal) [Source code](#)

parameters type *get_design_support*

- no type *int*

Output: get_design_support_response (soap:body, use = literal) [Source code](#)

parameters type *get_design_supportResponse*

- value type *design_support*
 - no type *int*
 - type - optional; type *design_support_type* - type *undefined* with restriction - enum { 'DESIGN_SUPPORT_TYPE_CONCRETE', 'DESIGN_SUPPORT_TYPE_GENERAL', 'DESIGN_SUPPORT_TYPE_TIMBER' }
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - assigned_to_members - optional; type *array_of_int*
 - assigned_to_member_sets - optional; type *array_of_int*
 - assigned_to_nodes - optional; type *array_of_int*
 - assigned_to_objects - optional; type *string*
 - comment - optional; type *string*
 - is_generated - optional; type *boolean*
 - generating_object_info - optional; type *string*
 - activate_in_y - optional; type *boolean*
 - activate_in_z - optional; type *boolean*
 - concrete_monolithic_connection_y_enabled - optional; type *boolean*
 - concrete_monolithic_connection_z_enabled - optional; type *boolean*
 - concrete_ratio_of_moment_redistribution_y - optional; type *double*
 - concrete_ratio_of_moment_redistribution_z - optional; type *double*
 - design_support_orientation_y - optional; type *design_support_design_support_orientation_y* - type *undefined* with restriction - enum { 'DESIGN_SUPPORT_ORIENTATION_YAXIS_BOTH',

- 'DESIGN_SUPPORT_ORIENTATION_Y_AXIS_NEGATIVE', 'DESIGN_SUPPORT_ORIENTATION_Y_AXIS_POSITIVE' }
- design_support_orientation_z - optional; type *design_support_design_support_orientation_z* - type *undefined* with restriction - enum { 'DESIGN_SUPPORT_ORIENTATION_Z_AXIS_BOTH', 'DESIGN_SUPPORT_ORIENTATION_Z_AXIS_NEGATIVE', 'DESIGN_SUPPORT_ORIENTATION_Z_AXIS_POSITIVE' }
- direct_support_y_enabled - optional; type *boolean*
- direct_support_z_enabled - optional; type *boolean*
- inner_support_y_enabled - optional; type *boolean*
- inner_support_z_enabled - optional; type *boolean*
- limit_of_high_bending_stresses_y - optional; type *double*
- limit_of_high_bending_stresses_z - optional; type *double*
- support_depth_by_section_width_of_member_y_enabled - optional; type *boolean*
- support_depth_by_section_width_of_member_z_enabled - optional; type *boolean*
- support_depth_y - optional; type *double*
- support_depth_z - optional; type *double*
- support_width_y - optional; type *double*
- support_width_z - optional; type *double*
- timber_allow_higher_deformation_y_enabled - optional; type *boolean*
- timber_allow_higher_deformation_z_enabled - optional; type *boolean*
- timber_calculation_method_y - optional; type *design_support_timber_calculation_method_y* - type *undefined* with restriction - enum { 'CALCULATION_METHOD_ACC_TO_4_2_2_2', 'CALCULATION_METHOD_ACC_TO_ANNEX_C' }
- timber_calculation_method_z - optional; type *design_support_timber_calculation_method_z* - type *undefined* with restriction - enum { 'CALCULATION_METHOD_ACC_TO_4_2_2_2', 'CALCULATION_METHOD_ACC_TO_ANNEX_C' }
- timber_check_critical_bearing_y_enabled - optional; type *boolean*
- timber_check_critical_bearing_z_enabled - optional; type *boolean*
- timber_compression_design_value_y - optional; type *design_support_timber_compression_design_value_y* - type *undefined* with restriction - enum { 'COMPRESSION_DESIGN_VALUE_0_0_2', 'COMPRESSION_DESIGN_VALUE_0_0_4' }
- timber_compression_design_value_z - optional; type *design_support_timber_compression_design_value_z* - type *undefined* with restriction - enum { 'COMPRESSION_DESIGN_VALUE_0_0_2', 'COMPRESSION_DESIGN_VALUE_0_0_4' }
- timber_factor_of_compression_y - optional; type *double*
- timber_factor_of_compression_z - optional; type *double*
- timber_load_bearing_capacity_of_screw_y - optional; type *double*
- timber_load_bearing_capacity_of_screw_z - optional; type *double*
- timber_number_of_screws_in_grain_direction_y - optional; type *int*
- timber_number_of_screws_in_grain_direction_z - optional; type *int*
- timber_number_of_screws_in_vertical_direction_y - optional; type *int*
- timber_number_of_screws_in_vertical_direction_z - optional; type *int*
- timber_spacing_of_rows_y - optional; type *double*
- timber_spacing_of_rows_z - optional; type *double*
- timber_spacing_of_screws_y - optional; type *double*
- timber_spacing_of_screws_z - optional; type *double*
- timber_stiffener_parameters_specification_type_y - optional; type *design_support_timber_stiffener_parameters_specification_type_y* - type *undefined* with restriction - enum { 'STIFFENER_PARAMETERS_SPECIFICATION_TYPE_AUTOMATICALLY', 'STIFFENER_PARAMETERS_SPECIFICATION_TYPE_MANUALLY' }
- timber_stiffener_parameters_specification_type_z - optional; type *design_support_timber_stiffener_parameters_specification_type_z* - type *undefined* with restriction - enum { 'STIFFENER_PARAMETERS_SPECIFICATION_TYPE_AUTOMATICALLY', 'STIFFENER_PARAMETERS_SPECIFICATION_TYPE_MANUALLY' }
- timber_stiffening_elements_type_y - optional; type *design_support_timber_stiffening_elements_type_y* - type *undefined* with restriction - enum { 'STIFFENING_ELEMENTS_TYPE_SCREWS' }
- timber_stiffening_elements_type_z - optional; type *design_support_timber_stiffening_elements_type_z* - type *undefined* with restriction - enum { 'STIFFENING_ELEMENTS_TYPE_SCREWS' }
- timber_stiffening_elements_y_enabled - optional; type *boolean*
- timber_stiffening_elements_z_enabled - optional; type *boolean*
- timber_ultimate_strength_of_screw_y - optional; type *double*
- timber_ultimate_strength_of_screw_z - optional; type *double*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

30. get_dimension

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_dimension

Input: get_dimension_request (soap:body, use = literal) [Source code](#)

parameters type *get_dimension*

- no type *int*

Output: get_dimension_response (soap:body, use = literal) [Source code](#)

parameters type *get_dimensionResponse*

- value type *dimension*
 - no type *int*
 - type - optional; type *dimension_type* - type *undefined* with restriction - enum { 'DIMENSION_TYPE_ANGULAR', 'DIMENSION_TYPE_ARC_LENGTH', 'DIMENSION_TYPE_DIAMETER', 'DIMENSION_TYPE_ELEVATION', 'DIMENSION_TYPE_LINEAR', 'DIMENSION_TYPE_RADIUS', 'DIMENSION_TYPE_SLOPE' }
 - user_defined_name_enabled - optional; type *boolean*
 - name - optional; type *string*
 - display_properties_index - optional; type *int*
 - dimension_line_offset - optional; type *double*
 - is_custom_vertical_position - optional; type *boolean*
 - custom_vertical_position - optional; type *dimension_custom_vertical_position* - type *undefined* with restriction - enum { 'VERTICAL_POSITION_ABOVE', 'VERTICAL_POSITION_CENTER', 'VERTICAL_POSITION_UNDER' }
 - is_custom_horizontal_position - optional; type *boolean*
 - custom_horizontal_position - optional; type *dimension_custom_horizontal_position* - type *undefined* with restriction - enum { 'HORIZONTAL_POSITION_CENTER', 'HORIZONTAL_POSITION_LEFT', 'HORIZONTAL_POSITION_RIGHT' }
 - symbol - optional; type *string*
 - comment - optional; type *string*
 - measured_length - optional; type *double*
 - measured_lengths - optional; type *array_of_int*

- measured_angle - optional; type *double*
- measured_angles - optional; type *array_of_int*
- measured_slope - optional; type *double*
- is_global_dimension_line_offset - optional; type *boolean*
- linear_coordinate_system - optional; type *int*
- linear_reference - optional; type *dimension_linear_reference* - type *undefined* with restriction - enum { 'REFERENCE_LENGTH', 'REFERENCE_PROJECTION_X', 'REFERENCE_PROJECTION_Y', 'REFERENCE_PROJECTION_Z' }
- linear_plane - optional; type *dimension_linear_plane* - type *undefined* with restriction - enum { 'PLANE_FIRST', 'PLANE_SECOND' }
- linear_global_dimension_line_offset - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- linear_global_dimension_line_offset_x - optional; type *double*
- linear_global_dimension_line_offset_y - optional; type *double*
- linear_global_dimension_line_offset_z - optional; type *double*
- linear_reference_table - optional; type *array_of_dimension_linear_reference_table*
 - dimension_linear_reference_table - optional, unbounded; type *dimension_linear_reference_table*
 - no - optional; type *int*
 - reference_object_type - optional; type *reference_object_type* - type *undefined* with restriction - enum { 'REFERENCE_TYPE_CONTROL_POINT', 'REFERENCE_TYPE_LINE', 'REFERENCE_TYPE_MEMBER', 'REFERENCE_TYPE_NODE', 'REFERENCE_TYPE_POINT', 'REFERENCE_TYPE_POINT_ON_LINE', 'REFERENCE_TYPE_SURFACE' }
 - reference_object - optional; type *int*
 - line_relative_distance - optional; type *double*
 - coordinate_x - optional; type *double*
 - coordinate_y - optional; type *double*
 - coordinate_z - optional; type *double*
- arc_length_reference_line - optional; type *int*
- arc_length_reference_table - optional; type *array_of_dimension_arc_length_reference_table*
 - dimension_arc_length_reference_table - optional, unbounded; type *dimension_arc_length_reference_table*
 - no - optional; type *int*
 - reference_object_type - optional; type *reference_object_type* - type *undefined* with restriction - enum { 'REFERENCE_TYPE_CONTROL_POINT', 'REFERENCE_TYPE_LINE', 'REFERENCE_TYPE_MEMBER', 'REFERENCE_TYPE_NODE', 'REFERENCE_TYPE_POINT', 'REFERENCE_TYPE_POINT_ON_LINE', 'REFERENCE_TYPE_SURFACE' }
 - reference_object - optional; type *int*
 - line_relative_distance - optional; type *double*
 - coordinate_x - optional; type *double*
 - coordinate_y - optional; type *double*
 - coordinate_z - optional; type *double*
- arc_length_angle_greater_than_180 - optional; type *boolean*
- angular_reference_table - optional; type *array_of_dimension_angular_reference_table*
 - dimension_angular_reference_table - optional, unbounded; type *dimension_angular_reference_table*
 - no - optional; type *int*
 - reference_object_type - optional; type *reference_object_type* - type *undefined* with restriction - enum { 'REFERENCE_TYPE_CONTROL_POINT', 'REFERENCE_TYPE_LINE', 'REFERENCE_TYPE_MEMBER', 'REFERENCE_TYPE_NODE', 'REFERENCE_TYPE_POINT', 'REFERENCE_TYPE_POINT_ON_LINE', 'REFERENCE_TYPE_SURFACE' }
 - reference_object - optional; type *int*
 - line_relative_distance - optional; type *double*
 - coordinate_x - optional; type *double*
 - coordinate_y - optional; type *double*
 - coordinate_z - optional; type *double*
- angular_quadrant - optional; type *dimension_angular_quadrant* - type *undefined* with restriction - enum { 'ANGULAR_QUADRANT_LEFT', 'ANGULAR_QUADRANT_NEGATIVE', 'ANGULAR_QUADRANT_POSITIVE', 'ANGULAR_QUADRANT_RIGHT' }
- angular_angle_greater_than_180 - optional; type *boolean*
- radius_diameter_reference_line - optional; type *int*
- radius_diameter_is_target_point - optional; type *boolean*
- radius_diameter_target_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- radius_diameter_target_point_coordinate_x - optional; type *double*
- radius_diameter_target_point_coordinate_y - optional; type *double*
- radius_diameter_target_point_coordinate_z - optional; type *double*
- radius_diameter_position_on_line - optional; type *double*
- slope_coordinate_system - optional; type *int*
- slope_plane - optional; type *dimension_slope_plane* - type *undefined* with restriction - enum { 'SLOPE_PLANE_XY', 'SLOPE_PLANE_XZ', 'SLOPE_PLANE_YZ' }
- slope_reference_line - optional; type *int*
- slope_direction - optional; type *dimension_slope_direction* - type *undefined* with restriction - enum { 'SLOPE_DIRECTION_DOWNWARD', 'SLOPE_DIRECTION_UPWARD' }
- slope_refer_distance_from_line_end - optional; type *boolean*
- slope_position_is_relative - optional; type *boolean*
- slope_position_absolute - optional; type *double*
- slope_position_relative - optional; type *double*
- elevation_reference_object_type - optional; type *dimension_elevation_reference_object_type* - type *undefined* with restriction - enum { 'ELEVATION_REFERENCE_OBJECT_TYPE_CONTROL_POINT', 'ELEVATION_REFERENCE_OBJECT_TYPE_NODE', 'ELEVATION_REFERENCE_OBJECT_TYPE_POINT', 'ELEVATION_REFERENCE_OBJECT_TYPE_SURFACE' }
- elevation_reference_surface - optional; type *int*
- elevation_reference_node - optional; type *int*
- elevation_reference_control_point - optional; type *int*
- elevation_reference_point_coordinates - optional; type *vector_3d*
 - x type *double*
 - y type *double*
 - z type *double*
- elevation_reference_point_coordinate_x - optional; type *double*
- elevation_reference_point_coordinate_y - optional; type *double*
- elevation_reference_point_coordinate_z - optional; type *double*
- elevation_distance_from_picked_position - optional; type *double*
- elevation_rotation_around_z - optional; type *double*
- elevation_reference_level_height - optional; type *double*
- elevation_is_altitude - optional; type *boolean*

- elevation_altitude - optional; type *double*
- is_generated - optional; type *boolean*
- generating_object_info - optional; type *string*
- id_for_export_import - optional; type *string*
- metadata_for_export_import - optional; type *string*

31. get_first_free_number

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_first_free_number

Input: get_first_free_number_request (soap:body, use = literal) [Source code](#)

parameters type *get_first_free_number*

- type type *object_types* - type *undefined* with restriction - enum { 'E_OBJECT_TYPE_ACTION', 'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX', 'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD', 'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING', 'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION', 'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD', 'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD', 'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD', 'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE', 'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION', 'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID', 'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT', 'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT', 'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION', 'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS', 'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE', 'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD', 'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET', 'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD', 'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER', 'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT', 'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE', 'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD', 'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK', 'E_OBJECT_TYPE_SECTION', 'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE', 'E_OBJECT_TYPE_SOLID', 'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS', 'E_OBJECT_TYPE_SOLID_LOAD', 'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET', 'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION', 'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT', 'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY', 'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD', 'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT', 'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS', 'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION', 'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
- parent_no - optional; type *int*

Output: get_first_free_number_response (soap:body, use = literal) [Source code](#)

parameters type *get_first_free_numberResponse*

- value type *int*

32. get_formula

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/get_formula

Input: get_formula_request (soap:body, use = literal) [Source code](#)

parameters type *get_formula*

- object_location type *object_location*
 - type type *object_types* - type *undefined* with restriction - enum { 'E_OBJECT_TYPE_ACTION', 'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX', 'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD', 'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING', 'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION', 'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD', 'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD', 'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD', 'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE', 'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION', 'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID', 'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT', 'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT', 'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION', 'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS', 'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE', 'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD', 'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET', 'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD', 'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION', 'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER', 'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT', 'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE', 'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD', 'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK', 'E_OBJECT_TYPE_SECTION',


```
'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE', 'E_OBJECT_TYPE_SOLID',
'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS', 'E_OBJECT_TYPE_SOLID_LOAD',
'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET',
'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS',
'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION',
'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT',
'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY',
'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD',
'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT',
'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION',
'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS',
'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION',
'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
```

- no *type int*
- parent_no - optional; *type int*
- object_parameter_location *type object_parameter_location_type*
 - attribute *type string*
 - parameter_path_in_nested_models_hierarchy - optional; *type parameter_path_in_nested_models_hierarchy_type*
 - node - optional, unbounded; *type node_of_parameter_path_in_nested_models_hierarchy_type*
 - row_path *type string*
 - column_string_id *type string*

Output: get_formula_response (soap:body, use = literal) [Source code](#)

- ```
parameters type get_formulaResponse
 value type formula_data
 formula type string
 is_valid type boolean
 error_message - optional; type string
 calculated_value - optional; type double
```

### 33. get\_free\_circular\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_free\_circular\_load

**Input:** get\_free\_circular\_load\_request (soap:body, use = literal) [Source code](#)

- ```
parameters type get_free_circular_load
  no type int
  load_case_no type int
```

Output: get_free_circular_load_response (soap:body, use = literal) [Source code](#)

- ```
parameters type get_free_circular_loadResponse
 value type free_circular_load
 no type int
 surfaces - optional; type array_of_int
 load_case - optional; type int
 coordinate_system - optional; type int
 load_projection - optional; type free_circular_load_load_projection - type undefined with restriction - enum {
 'LOAD_PROJECTION_XY_OR_UV', 'LOAD_PROJECTION_XZ_OR_UW', 'LOAD_PROJECTION_YZ_OR_VW' }
 load_direction - optional; type free_circular_load_load_direction - type undefined with restriction - enum {
 'LOAD_DIRECTION_GLOBAL_X_PROJECTED', 'LOAD_DIRECTION_GLOBAL_X_TRUE',
 'LOAD_DIRECTION_GLOBAL_Y_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Y_TRUE',
 'LOAD_DIRECTION_GLOBAL_Z_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Z_TRUE',
 'LOAD_DIRECTION_LOCAL_X', 'LOAD_DIRECTION_LOCAL_Y', 'LOAD_DIRECTION_LOCAL_Z',
 'LOAD_DIRECTION_USER_DEFINED_U_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_U_TRUE',
 'LOAD_DIRECTION_USER_DEFINED_V_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_V_TRUE',
 'LOAD_DIRECTION_USER_DEFINED_W_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_W_TRUE' }
 load_acting_region_from - optional; type double
 load_acting_region_to - optional; type double
 load_distribution - optional; type free_circular_load_load_distribution - type undefined with restriction - enum {
 'LOAD_DISTRIBUTION_LINEAR', 'LOAD_DISTRIBUTION_UNIFORM' }
 magnitude_uniform - optional; type double
 magnitude_center - optional; type double
 magnitude_radius - optional; type double
 load_location_x - optional; type double
 load_location_y - optional; type double
 load_location_radius - optional; type double
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

### 34. get\_free\_concentrated\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_free\_concentrated\_load

**Input:** get\_free\_concentrated\_load\_request (soap:body, use = literal) [Source code](#)

- ```
parameters type get_free_concentrated_load
  no type int
  load_case_no type int
```

Output: get_free_concentrated_load_response (soap:body, use = literal) [Source code](#)

- ```
parameters type get_free_concentrated_loadResponse
 value type free_concentrated_load
 no type int
 load_type - optional; type free_concentrated_load_load_type - type undefined with restriction - enum {
 'LOAD_TYPE_FORCE', 'LOAD_TYPE_MOMENT' }
```

- surfaces - optional; type *array\_of\_int*
- load\_case - optional; type *int*
- coordinate\_system - optional; type *int*
- load\_projection - optional; type *free\_concentrated\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
- load\_direction - optional; type *free\_concentrated\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X', 'LOAD\_DIRECTION\_GLOBAL\_Y', 'LOAD\_DIRECTION\_GLOBAL\_Z', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U', 'LOAD\_DIRECTION\_USER\_DEFINED\_V', 'LOAD\_DIRECTION\_USER\_DEFINED\_W' }
- load\_acting\_region\_from - optional; type *double*
- load\_acting\_region\_to - optional; type *double*
- magnitude - optional; type *double*
- load\_location\_x - optional; type *double*
- load\_location\_y - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 35. get\_free\_line\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_free\_line\_load

**Input:** get\_free\_line\_load\_request (soap:body, use = literal) [Source code](#)

- parameters type *get\_free\_line\_load*
- no type *int*
  - load\_case\_no type *int*

**Output:** get\_free\_line\_load\_response (soap:body, use = literal) [Source code](#)

- parameters type *get\_free\_line\_loadResponse*
- value type *free\_line\_load*
    - no type *int*
    - surfaces - optional; type *array\_of\_int*
    - load\_case - optional; type *int*
    - coordinate\_system - optional; type *int*
    - load\_projection - optional; type *free\_line\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
    - load\_direction - optional; type *free\_line\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }
    - load\_acting\_region\_from - optional; type *double*
    - load\_acting\_region\_to - optional; type *double*
    - load\_distribution - optional; type *free\_line\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_UNIFORM' }
    - magnitude\_uniform - optional; type *double*
    - magnitude\_first - optional; type *double*
    - magnitude\_second - optional; type *double*
    - load\_location\_first\_x - optional; type *double*
    - load\_location\_first\_y - optional; type *double*
    - load\_location\_second\_x - optional; type *double*
    - load\_location\_second\_y - optional; type *double*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

### 36. get\_free\_polygon\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_free\_polygon\_load

**Input:** get\_free\_polygon\_load\_request (soap:body, use = literal) [Source code](#)

- parameters type *get\_free\_polygon\_load*
- no type *int*
  - load\_case\_no type *int*

**Output:** get\_free\_polygon\_load\_response (soap:body, use = literal) [Source code](#)

- parameters type *get\_free\_polygon\_loadResponse*
- value type *free\_polygon\_load*
    - no type *int*
    - surfaces - optional; type *array\_of\_int*
    - load\_case - optional; type *int*
    - coordinate\_system - optional; type *int*
    - load\_projection - optional; type *free\_polygon\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
    - load\_direction - optional; type *free\_polygon\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }

'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE',  
'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }

- load\_acting\_region\_from - optional; type *double*
- load\_acting\_region\_to - optional; type *double*
- load\_distribution - optional; type *free\_polygon\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_LINEAR\_FIRST', 'LOAD\_DISTRIBUTION\_LINEAR\_SECOND', 'LOAD\_DISTRIBUTION\_UNIFORM' }
- magnitude\_uniform - optional; type *double*
- magnitude\_linear\_1 - optional; type *double*
- magnitude\_linear\_2 - optional; type *double*
- magnitude\_linear\_3 - optional; type *double*
- magnitude\_linear\_location\_1 - optional; type *int*
- magnitude\_linear\_location\_2 - optional; type *int*
- magnitude\_linear\_location\_3 - optional; type *int*
- load\_location - optional; type *array\_of\_free\_polygon\_load\_load\_location*
  - free\_polygon\_load\_load\_location - optional, unbounded; type *free\_polygon\_load\_load\_location*
    - no - optional; type *int*
    - first\_coordinate - optional; type *double*
    - second\_coordinate - optional; type *double*
    - magnitude - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 37. get\_free\_rectangular\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_free\_rectangular\_load

**Input:** get\_free\_rectangular\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_free\_rectangular\_load*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_free\_rectangular\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_free\_rectangular\_loadResponse*

- value type *free\_rectangular\_load*
  - no type *int*
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_rectangular\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_rectangular\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }
  - load\_acting\_region\_from - optional; type *double*
  - load\_acting\_region\_to - optional; type *double*
  - load\_distribution - optional; type *free\_rectangular\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR\_FIRST', 'LOAD\_DISTRIBUTION\_LINEAR\_SECOND', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_VARYING\_ALONG\_PERIMETER', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z\_AND\_ALONG\_PERIMETER' }
  - magnitude\_uniform - optional; type *double*
  - magnitude\_linear\_first - optional; type *double*
  - magnitude\_linear\_second - optional; type *double*
  - load\_location\_first\_x - optional; type *double*
  - load\_location\_first\_y - optional; type *double*
  - load\_location\_second\_x - optional; type *double*
  - load\_location\_second\_y - optional; type *double*
  - load\_location\_rectangle - optional; type *free\_rectangular\_load\_load\_location\_rectangle* - type *undefined* with restriction - enum { 'LOAD\_LOCATION\_RECTANGLE\_CENTER\_AND\_SIDES', 'LOAD\_LOCATION\_RECTANGLE\_CORNER\_POINTS' }
  - load\_location\_center\_x - optional; type *double*
  - load\_location\_center\_y - optional; type *double*
  - load\_location\_center\_side\_a - optional; type *double*
  - load\_location\_center\_side\_b - optional; type *double*
  - load\_location\_rotation - optional; type *double*
  - load\_varying\_in\_z\_parameters - optional; type *array\_of\_free\_rectangular\_load\_load\_varying\_in\_z\_parameters*
    - free\_rectangular\_load\_load\_varying\_in\_z\_parameters - optional, unbounded; type *free\_rectangular\_load\_load\_varying\_in\_z\_parameters*
      - no - optional; type *int*
      - distance - optional; type *double*
      - recalculated\_magnitude - optional; type *double*
      - factor - optional; type *double*
      - note - optional; type *string*
  - load\_varying\_along\_perimeter\_parameters - optional; type *array\_of\_free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters*
    - free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters - optional, unbounded; type *free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters*
      - no - optional; type *int*
      - alpha - optional; type *double*
      - recalculated\_magnitude - optional; type *double*
      - factor - optional; type *double*
      - note - optional; type *string*
  - load\_varying\_in\_z\_parameters\_sorted - optional; type *boolean*
  - load\_varying\_along\_perimeter\_parameters\_sorted - optional; type *boolean*

- load\_varying\_along\_perimeter\_z\_index - optional; type *int*
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_start\_angle - optional; type *double*
- comment - optional; type *string*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 38. get\_global\_parameter

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_global\_parameter

**Input:** get\_global\_parameter\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_global\_parameter*

- no type *int*

**Output:** get\_global\_parameter\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_global\_parameterResponse*

- value type *global\_parameter*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - attribute\_always\_editable - optional; type *string*
  - symbol - optional; type *string*
  - unit\_group - optional; type *global\_parameter\_unit\_group* - type *undefined* with restriction - enum { 'ANGLE', 'AREA', 'DENSITY', 'DIMENSIONLESS', 'DYNAMIC\_INCREASE\_FACTOR', 'EG\_MODULE', 'FRICTION\_COEFFICIENT', 'GEOGRAPHIC\_COORDINATES', 'GRAVITATIONAL\_ACCELERATION', 'LENGTH', 'LOADS\_ANGULAR\_ACCELERATION', 'LOADS\_ANGULAR\_VELOCITY', 'LOADS\_AREA\_MASS', 'LOADS\_AXIAL\_STRAIN', 'LOADS\_DENSITY', 'LOADS\_DISPLACEMENT', 'LOADS\_DISPLACEMENT\_PER\_UNIT\_LENGTH', 'LOADS\_FORCE', 'LOADS\_FORCE\_PER\_UNIT\_LENGTH', 'LOADS\_IMPOSED\_DISPLACEMENT', 'LOADS\_IMPOSED\_ROTATION', 'LOADS\_KINEMATIC\_VISCOSITY', 'LOADS\_KINETIC\_ENERGY', 'LOADS\_LENGTH', 'LOADS\_MASS', 'LOADS\_MOMENT', 'LOADS\_MOMENT\_PER\_UNIT\_LENGTH', 'LOADS\_PRECAMBER', 'LOADS\_PRESSURE', 'LOADS\_RELATIVE\_LENGTH', 'LOADS\_ROTATION', 'LOADS\_ROTATION\_PER\_UNIT\_LENGTH', 'LOADS\_SOLID\_TYPE\_LOAD', 'LOADS\_SPECIFIC\_ENERGY', 'LOADS\_SURFACE\_TYPE\_LOAD', 'LOADS\_TEMPERATURE', 'LOADS\_TEMPERATURE\_CHANGE', 'LOADS\_TURBULENCE\_DISSIPATION\_RATE', 'LOADS\_VELOCITY', 'MASS', 'MASS\_MOMENT\_PER\_UNIT\_AREA', 'MATERIAL\_ANGLE', 'MATERIAL\_DEFORMATION', 'MATERIAL\_FACTOR', 'MATERIAL\_QUANTITY\_FLOAT\_PRECISION\_2', 'MATERIAL\_QUANTITY\_INTEGER', 'MATERIAL\_SPECIFIC\_WEIGHT', 'MATERIAL\_THICKNESS', 'PARTIAL\_FACTOR', 'POISSONS\_RATIO', 'PRECISION\_FACTOR', 'QUANTITY', 'QUANTITY\_INTEGER', 'RATIO', 'RELATIVE\_LENGTH', 'SECTION\_ANGLE', 'SECTION\_AREA', 'SECTION\_BIMOMENT', 'SECTION\_COMPLIANCE', 'SECTION\_DIMENSION', 'SECTION\_EFFECTIVE\_AREA', 'SECTION\_EFFECTIVE\_SECOND\_MOMENT\_OF\_AREA', 'SECTION\_FORCE', 'SECTION\_MOMENT', 'SECTION\_MOMENT\_OF\_INERTIA', 'SECTION\_NORMALIZED\_WARPING\_CONSTANT', 'SECTION\_PERIMETER', 'SECTION\_QUANTITY', 'SECTION\_SECTION\_FACTOR', 'SECTION\_SECTION\_MODULUS', 'SECTION\_STATIC\_MOMENT\_OF\_AREA', 'SECTION\_TENSION\_FIELD\_COEFFICIENT\_1', 'SECTION\_TENSION\_FIELD\_COEFFICIENT\_2', 'SECTION\_UNIT\_STRESSES', 'SECTION\_UNIT\_WARPING\_FUNCTION', 'SECTION\_WARPING\_CONSTANT', 'SECTION\_WARPING\_STATIC\_MOMENT', 'SELF\_WEIGHT\_FACTOR', 'STIFFNESS\_MULTIPLICATION\_FACTOR', 'STRAIN', 'STRAIN\_RATE', 'STRESSES', 'THERMAL\_EXPANSION\_COEFFICIENT', 'THICKNESS', 'TIME', 'VOLUME', 'WEIGHT\_AND\_KNOT' }
  - definition\_type - optional; type *global\_parameter\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_FORMULA', 'DEFINITION\_TYPE\_VALUE' }
  - value - optional; type *double*
  - unit - optional; type *string*
  - formula - optional; type *string*
  - min - optional; type *double*
  - max - optional; type *double*
  - increment - optional; type *double*
  - steps - optional; type *int*
  - comment - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 39. get\_imperfection\_case

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_imperfection\_case

**Input:** get\_imperfection\_case\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_imperfection\_case*

- no type *int*

**Output:** get\_imperfection\_case\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_imperfection\_caseResponse*

- value type *imperfection\_case*
  - no type *int*

- type - optional; type *imperfection\_case\_type* - type *undefined* with restriction - enum { 'IMPERFECTION\_TYPE\_BUCKLING\_MODE', 'IMPERFECTION\_TYPE\_DYNAMIC\_EIGENMODE', 'IMPERFECTION\_TYPE\_IMPERFECTION\_CASES\_GROUP', 'IMPERFECTION\_TYPE\_INITIAL\_SWAY\_VIA\_TABLE', 'IMPERFECTION\_TYPE\_LOCAL\_IMPERFECTIONS', 'IMPERFECTION\_TYPE\_NOTIONAL\_LOADS\_FROM\_LOAD\_CASE', 'IMPERFECTION\_TYPE\_STATIC\_DEFORMATION' }
- user\_defined\_name\_enabled - optional; type *boolean*
- name - optional; type *string*
- assigned\_to\_load\_cases - optional; type *array\_of\_int*
- assigned\_to\_load\_combinations - optional; type *array\_of\_int*
- is\_active - optional; type *boolean*
- assign\_to\_combinations\_without\_assigned\_imperfection\_case - optional; type *boolean*
- direction - optional; type *imperfection\_case\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_NEGATIVE', 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_NEGATIVE', 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_NEGATIVE', 'IMPERFECTION\_CASE\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'IMPERFECTION\_CASE\_DIRECTION\_LOCAL\_X', 'IMPERFECTION\_CASE\_DIRECTION\_LOCAL\_Y', 'IMPERFECTION\_CASE\_DIRECTION\_LOCAL\_Y\_NEGATIVE', 'IMPERFECTION\_CASE\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_CASE\_DIRECTION\_LOCAL\_Z\_NEGATIVE', 'IMPERFECTION\_CASE\_DIRECTION\_SPATIAL' }
- direction\_for\_level\_direction - optional; type *imperfection\_case\_direction\_for\_level\_direction* - type *undefined* with restriction - enum { 'DIRECTION\_X', 'DIRECTION\_XY', 'DIRECTION\_XZ', 'DIRECTION\_Y', 'DIRECTION\_YZ', 'DIRECTION\_Z' }
- coordinate\_system - optional; type *int*
- load\_case\_for\_notional\_loads - optional; type *int*
- sway\_coefficients\_reciprocal - optional; type *boolean*
- level\_imperfections - optional; type *array\_of\_imperfection\_case\_level\_imperfections*
  - imperfection\_case\_level\_imperfections - optional, unbounded; type *imperfection\_case\_level\_imperfections*
    - no - optional; type *int*
    - level - optional; type *double*
    - e\_1 - optional; type *double*
    - theta\_1 - optional; type *double*
    - e\_2 - optional; type *double*
    - theta\_2 - optional; type *double*
    - comment - optional; type *string*
- source - optional; type *imperfection\_case\_source* - type *undefined* with restriction - enum { 'SOURCE\_TYPE\_AUTOMATICALLY', 'SOURCE\_TYPE\_LOAD\_CASE', 'SOURCE\_TYPE\_LOAD\_COMBINATION', 'SOURCE\_TYPE\_OWN\_LOAD\_CASE\_OR\_COMBINATION' }
- shape\_from\_load\_case - optional; type *int*
- shape\_from\_load\_combination - optional; type *int*
- buckling\_shape - optional; type *int*
- delta\_zero - optional; type *double*
- magnitude\_assignment\_type - optional; type *imperfection\_case\_magnitude\_assignment\_type* - type *undefined* with restriction - enum { 'MAGNITUDE\_ASSIGNMENT\_LOCATION\_WITH\_LARGEST\_DISPLACEMENT', 'MAGNITUDE\_ASSIGNMENT\_SPECIFIC\_NODE' }
- reference\_node - optional; type *int*
- amount\_of\_modes\_to\_investigate - optional; type *int*
- eigenmode\_automatically - optional; type *boolean*
- imperfection\_cases\_items - optional; type *array\_of\_imperfection\_case\_imperfection\_cases\_items*
  - imperfection\_case\_imperfection\_cases\_items - optional, unbounded; type *imperfection\_case\_imperfection\_cases\_items*
    - no - optional; type *int*
    - name - optional; type *int*
    - factor - optional; type *double*
    - operator\_type - optional; type *operator\_type* - type *undefined* with restriction - enum { 'OPERATOR\_AND', 'OPERATOR\_NONE', 'OPERATOR\_OR' }
    - comment - optional; type *string*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 40. get\_imposed\_line\_deformation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_imposed\_line\_deformation

**Input:** get\_imposed\_line\_deformation\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_imposed\_line\_deformation*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_imposed\_line\_deformation\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_imposed\_line\_deformationResponse*

- value type *imposed\_line\_deformation*
  - no type *int*
  - lines - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - imposed\_displacement\_line\_start - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - imposed\_displacement\_line\_start\_x - optional; type *double*
  - imposed\_displacement\_line\_start\_y - optional; type *double*
  - imposed\_displacement\_line\_start\_z - optional; type *double*
  - imposed\_displacement\_line\_end - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - imposed\_displacement\_line\_end\_x - optional; type *double*
  - imposed\_displacement\_line\_end\_y - optional; type *double*

- imposed\_displacement\_line\_end\_z - optional; type *double*
- imposed\_rotation\_line\_start - optional; type *double*
- imposed\_rotation\_line\_end - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 41. get\_imposed\_nodal\_deformation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_imposed\_nodal\_deformation

**Input:** get\_imposed\_nodal\_deformation\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_imposed\_nodal\_deformation*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_imposed\_nodal\_deformation\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_imposed\_nodal\_deformationResponse*

- value type *imposed\_nodal\_deformation*
  - no type *int*
  - nodes - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - imposed\_displacement - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - imposed\_displacement\_x - optional; type *double*
  - imposed\_displacement\_y - optional; type *double*
  - imposed\_displacement\_z - optional; type *double*
  - imposed\_rotation - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - imposed\_rotation\_x - optional; type *double*
  - imposed\_rotation\_y - optional; type *double*
  - imposed\_rotation\_z - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

#### 42. get\_intersection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_intersection

**Input:** get\_intersection\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_intersection*

- no type *int*

**Output:** get\_intersection\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_intersectionResponse*

- value type *intersection*
  - no type *int*
  - comment - optional; type *string*
  - generated\_lines - optional; type *array\_of\_int*
  - generated\_nodes - optional; type *array\_of\_int*
  - surface\_a - optional; type *int*
  - surface\_b - optional; type *int*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

#### 43. get\_line

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line

**Input:** get\_line\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_line*

- no type *int*

**Output:** get\_line\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_lineResponse*

- value type *line*
  - no type *int*
  - type - optional; type *line\_type* - type *undefined* with restriction - enum { 'TYPE\_ARC', 'TYPE\_CIRCLE', 'TYPE\_CUT\_VIA\_SECTION', 'TYPE\_CUT\_VIA\_TWO\_LINES', 'TYPE\_ELLIPSE', 'TYPE\_ELLIPTICAL\_ARC', 'TYPE\_NURBS', 'TYPE\_PARABOLA', 'TYPE\_POLYLINE', 'TYPE\_SPLINE' }
  - definition\_nodes - optional; type *array\_of\_int*
  - length - optional; type *double*
  - comment - optional; type *string*
  - arc\_first\_node - optional; type *int*
  - arc\_second\_node - optional; type *int*
  - arc\_control\_point - optional; type *vector\_3d*

- x type *double*
- y type *double*
- z type *double*
- arc\_control\_point\_x - optional; type *double*
- arc\_control\_point\_y - optional; type *double*
- arc\_control\_point\_z - optional; type *double*
- arc\_center - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- arc\_center\_x - optional; type *double*
- arc\_center\_y - optional; type *double*
- arc\_center\_z - optional; type *double*
- arc\_radius - optional; type *double*
- arc\_height - optional; type *double*
- arc\_alpha - optional; type *double*
- arc\_alpha\_adjustment\_target - optional; type *line\_arc\_alpha\_adjustment\_target* - type *undefined* with restriction - enum { 'ALPHA\_ADJUSTMENT\_TARGET\_ARC\_CONTROL\_POINT', 'ALPHA\_ADJUSTMENT\_TARGET\_BEGINNING\_OF\_ARC', 'ALPHA\_ADJUSTMENT\_TARGET\_END\_OF\_ARC' }
- circle\_center - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- circle\_center\_coordinate\_1 - optional; type *double*
- circle\_center\_coordinate\_2 - optional; type *double*
- circle\_center\_coordinate\_3 - optional; type *double*
- circle\_normal - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- circle\_normal\_coordinate\_1 - optional; type *double*
- circle\_normal\_coordinate\_2 - optional; type *double*
- circle\_normal\_coordinate\_3 - optional; type *double*
- circle\_rotation - optional; type *double*
- circle\_node - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- circle\_radius - optional; type *double*
- ellipse\_first\_node - optional; type *int*
- ellipse\_second\_node - optional; type *int*
- ellipse\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- ellipse\_control\_point\_x - optional; type *double*
- ellipse\_control\_point\_y - optional; type *double*
- ellipse\_control\_point\_z - optional; type *double*
- elliptical\_arc\_first\_node - optional; type *int*
- elliptical\_arc\_second\_node - optional; type *int*
- elliptical\_arc\_alpha - optional; type *double*
- elliptical\_arc\_beta - optional; type *double*
- elliptical\_arc\_normal - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_normal\_x - optional; type *double*
- elliptical\_arc\_normal\_y - optional; type *double*
- elliptical\_arc\_normal\_z - optional; type *double*
- elliptical\_arc\_major\_radius - optional; type *double*
- elliptical\_arc\_minor\_radius - optional; type *double*
- elliptical\_arc\_center - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_center\_x - optional; type *double*
- elliptical\_arc\_center\_y - optional; type *double*
- elliptical\_arc\_center\_z - optional; type *double*
- elliptical\_arc\_focus\_1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_focus\_1\_x - optional; type *double*
- elliptical\_arc\_focus\_1\_y - optional; type *double*
- elliptical\_arc\_focus\_1\_z - optional; type *double*
- elliptical\_arc\_focus\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_focus\_2\_x - optional; type *double*
- elliptical\_arc\_focus\_2\_y - optional; type *double*
- elliptical\_arc\_focus\_2\_z - optional; type *double*
- elliptical\_arc\_first\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_first\_control\_point\_x - optional; type *double*
- elliptical\_arc\_first\_control\_point\_y - optional; type *double*
- elliptical\_arc\_first\_control\_point\_z - optional; type *double*
- elliptical\_arc\_second\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_second\_control\_point\_x - optional; type *double*
- elliptical\_arc\_second\_control\_point\_y - optional; type *double*

- elliptical\_arc\_second\_control\_point\_z - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_perimeter\_control\_point\_x - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point\_y - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point\_z - optional; type *double*
- parabola\_first\_node - optional; type *int*
- parabola\_second\_node - optional; type *int*
- parabola\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- parabola\_control\_point\_x - optional; type *double*
- parabola\_control\_point\_y - optional; type *double*
- parabola\_control\_point\_z - optional; type *double*
- parabola\_alpha - optional; type *double*
- parabola\_focus\_directrix\_distance - optional; type *double*
- parabola\_focus - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- parabola\_focus\_x - optional; type *double*
- parabola\_focus\_y - optional; type *double*
- parabola\_focus\_z - optional; type *double*
- nurbs\_order - optional; type *int*
- nurbs\_control\_points\_by\_components - optional; type *array\_of\_line\_nurbs\_control\_points\_by\_components*
  - line\_nurbs\_control\_points\_by\_components - optional, unbounded; type *line\_nurbs\_control\_points\_by\_components*
    - no - optional; type *int*
    - global\_coordinate\_x - optional; type *double*
    - global\_coordinate\_y - optional; type *double*
    - global\_coordinate\_z - optional; type *double*
    - weight - optional; type *double*
- nurbs\_control\_points - optional; type *array\_of\_line\_nurbs\_control\_points*
  - line\_nurbs\_control\_points - optional, unbounded; type *line\_nurbs\_control\_points*
    - no - optional; type *int*
    - global\_coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - weight type *double*
- nurbs\_knots - optional; type *array\_of\_line\_nurbs\_knots*
  - line\_nurbs\_knots - optional, unbounded; type *double*
- rotation\_specification\_type - optional; type *line\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE' }
- rotation\_angle - optional; type *double*
- rotation\_help\_node - optional; type *int*
- rotation\_plane - optional; type *line\_rotation\_plane* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- is\_rotated - optional; type *boolean*
- member - optional; type *int*
- support - optional; type *int*
- mesh\_refinement - optional; type *int*
- line\_weld\_assignment - optional; type *array\_of\_line\_line\_weld\_assignment*
  - line\_line\_weld\_assignment - optional, unbounded; type *line\_line\_weld\_assignment*
    - no - optional; type *int*
    - weld - optional; type *int*
    - surface1 - optional; type *int*
    - surface2 - optional; type *int*
    - surface3 - optional; type *int*
- has\_line\_welds - optional; type *boolean*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 44. get\_line\_grid

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line\_grid

**Input:** get\_line\_grid\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_grid*

- no type *int*

**Output:** get\_line\_grid\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_gridResponse*

- value type *line\_grid*
  - no type *int*
  - type - optional; type *line\_grid\_type* - type *undefined* with restriction - enum { 'TYPE\_STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - coordinate\_system\_type - optional; type *line\_grid\_coordinate\_system\_type* - type *undefined* with restriction - enum { 'CARTESIAN', 'CYLINDRICAL', 'INCLINED', 'SPHERICAL' }
  - lock\_line\_grid - optional; type *boolean*
  - include\_in\_view - optional; type *boolean*



- line\_grid\_points - optional; type *boolean*
- line\_grid\_lines - optional; type *boolean*
- label\_text - optional; type *boolean*
- dimensions - optional; type *boolean*
- lock\_points\_to\_line\_grid - optional; type *boolean*
- origin\_type - optional; type *line\_grid\_origin\_type* - type *undefined* with restriction - enum { 'ORIGIN\_FREE\_POINT', 'ORIGIN\_NODE' }
- origin\_node - optional; type *int*
- origin\_free\_point\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- origin\_free\_point\_coordinate\_x - optional; type *double*
- origin\_free\_point\_coordinate\_y - optional; type *double*
- origin\_free\_point\_coordinate\_z - optional; type *double*
- alpha\_ux - optional; type *double*
- alpha\_vy - optional; type *double*
- alpha\_wz - optional; type *double*
- x\_input\_type - optional; type *line\_grid\_x\_input\_type* - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
- x\_direction - optional; type *line\_grid\_x\_direction* - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
- x\_assignment - optional; type *array\_of\_line\_grid\_x\_assignment*
  - line\_grid\_x\_assignment - optional, unbounded; type *line\_grid\_x\_assignment*
    - no - optional; type *int*
    - name - optional; type *string*
    - coordinates - optional; type *double*
    - spans - optional; type *double*
    - spans\_count - optional; type *int*
- y\_input\_type - optional; type *line\_grid\_y\_input\_type* - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
- y\_direction - optional; type *line\_grid\_y\_direction* - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
- y\_assignment - optional; type *array\_of\_line\_grid\_y\_assignment*
  - line\_grid\_y\_assignment - optional, unbounded; type *line\_grid\_y\_assignment*
    - no - optional; type *int*
    - name - optional; type *string*
    - coordinates - optional; type *double*
    - spans - optional; type *double*
    - spans\_count - optional; type *int*
- z\_input\_type - optional; type *line\_grid\_z\_input\_type* - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
- z\_direction - optional; type *line\_grid\_z\_direction* - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
- z\_assignment - optional; type *array\_of\_line\_grid\_z\_assignment*
  - line\_grid\_z\_assignment - optional, unbounded; type *line\_grid\_z\_assignment*
    - no - optional; type *int*
    - name - optional; type *string*
    - coordinates - optional; type *double*
    - spans - optional; type *double*
    - spans\_count - optional; type *int*
- coordinate\_system - optional; type *int*
- rotation\_coordinate\_system - optional; type *int*
- has\_specific\_direction - optional; type *boolean*
- specific\_direction\_type - optional; type *line\_grid\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
- axes\_sequence - optional; type *line\_grid\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
- rotated\_about\_angle\_x - optional; type *double*
- rotated\_about\_angle\_y - optional; type *double*
- rotated\_about\_angle\_z - optional; type *double*
- rotated\_about\_angle\_1 - optional; type *double*
- rotated\_about\_angle\_2 - optional; type *double*
- rotated\_about\_angle\_3 - optional; type *double*
- directed\_to\_node\_direction\_node - optional; type *int*
- directed\_to\_node\_plane\_node - optional; type *int*
- directed\_to\_node\_first\_axis - optional; type *line\_grid\_directed\_to\_node\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- directed\_to\_node\_second\_axis - optional; type *line\_grid\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *line\_grid\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_axis - optional; type *line\_grid\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 45. get\_line\_hinge

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line\_hinge

**Input:** get\_line\_hinge\_request (soap:body, use = literal) [Source code](#)

- parameters type *get\_line\_hinge*
- no type *int*

Output: `get_line_hinge_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_hingeResponse
 value type line_hinge
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 assigned_to - optional; type string
 translational_release_u_x - optional; type double
 translational_release_u_x_nonlinearity - optional; type line_hinge_translational_release_u_x_nonlinearity - type undefined with restriction - enum { 'NONLINEARITY_TYPE_DIAGRAM', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE', 'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE', 'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_2', 'NONLINEARITY_TYPE_NONE', 'NONLINEARITY_TYPE_PARTIAL_ACTIVITY', 'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM' }
 translational_release_u_y - optional; type double
 translational_release_u_y_nonlinearity - optional; type line_hinge_translational_release_u_y_nonlinearity - type undefined with restriction - enum { 'NONLINEARITY_TYPE_DIAGRAM', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE', 'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE', 'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_2', 'NONLINEARITY_TYPE_NONE', 'NONLINEARITY_TYPE_PARTIAL_ACTIVITY', 'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM' }
 translational_release_u_z - optional; type double
 translational_release_u_z_nonlinearity - optional; type line_hinge_translational_release_u_z_nonlinearity - type undefined with restriction - enum { 'NONLINEARITY_TYPE_DIAGRAM', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE', 'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE', 'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_2', 'NONLINEARITY_TYPE_NONE', 'NONLINEARITY_TYPE_PARTIAL_ACTIVITY', 'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM' }
 rotational_release_phi_x - optional; type double
 rotational_release_phi_x_nonlinearity - optional; type line_hinge_rotational_release_phi_x_nonlinearity - type undefined with restriction - enum { 'NONLINEARITY_TYPE_DIAGRAM', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE', 'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE', 'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE', 'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2', 'NONLINEARITY_TYPE_FRICTION_DIRECTION_2', 'NONLINEARITY_TYPE_NONE', 'NONLINEARITY_TYPE_PARTIAL_ACTIVITY', 'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM' }
 force_moment_diagram_around_x_table - optional; type array_of_line_hinge_force_moment_diagram_around_x_table
 line_hinge_force_moment_diagram_around_x_table - optional, unbounded; type line_hinge_force_moment_diagram_around_x_table
 no - optional; type int
 force - optional; type double
 max_moment - optional; type double
 min_moment - optional; type double
 note - optional; type string
 force_moment_diagram_around_x_symmetric - optional; type boolean
 force_moment_diagram_around_x_is_sorted - optional; type boolean
 force_moment_diagram_around_x_start - optional; type line_hinge_force_moment_diagram_around_x_start - type undefined with restriction - enum { 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_CONTINUOUS', 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_TEARING', 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_YIELDING' }
 force_moment_diagram_around_x_end - optional; type line_hinge_force_moment_diagram_around_x_end - type undefined with restriction - enum { 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_CONTINUOUS', 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_TEARING', 'FORCE_MOMENT_DIAGRAM_ENDING_TYPE_YIELDING' }
 force_moment_diagram_around_x_depends_on - optional; type line_hinge_force_moment_diagram_around_x_depends_on - type undefined with restriction - enum { 'FORCE_MOMENT_DIAGRAM_DEPENDS_ON_N', 'FORCE_MOMENT_DIAGRAM_DEPENDS_ON_VY', 'FORCE_MOMENT_DIAGRAM_DEPENDS_ON_VZ' }
 slab_wall_connection - optional; type boolean
 slab_wall_with_slab_edge_block - optional; type boolean
 slab_wall_connection_offset - optional; type double
 slab_edge_block_width - optional; type double
 generated_line_hinges - optional; type array_of_line_hinge_generated_line_hinges
 line_hinge_generated_line_hinges - optional, unbounded; type line_hinge_generated_line_hinges
 no - optional; type int
 generated_by - optional; type int
 generated_line_hinge - optional; type int
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

#### 46. `get_line_load`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_line_load`

**Input:** `get_line_load_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_load
 no type int
 load_case_no type int
```

**Output:** `get_line_load_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_loadResponse
 value type line_load
```

- no *type int*
- load\_type - optional; type *line\_load\_load\_type* - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MOMENT' }
- lines - optional; type *array\_of\_int*
- load\_case - optional; type *int*
- coordinate\_system - optional; type *string*
- load\_distribution - optional; type *line\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING' }
- load\_direction - optional; type *line\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
- load\_direction\_orientation - optional; type *line\_load\_load\_direction\_orientation* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
- magnitude - optional; type *double*
- magnitude\_1 - optional; type *double*
- magnitude\_2 - optional; type *double*
- magnitude\_3 - optional; type *double*
- individual\_mass\_components - optional; type *boolean*
- mass\_global - optional; type *double*
- mass\_x - optional; type *double*
- mass\_y - optional; type *double*
- mass\_z - optional; type *double*
- distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_a\_absolute - optional; type *double*
- distance\_a\_relative - optional; type *double*
- distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_b\_absolute - optional; type *double*
- distance\_b\_relative - optional; type *double*
- distance\_c\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_c\_absolute - optional; type *double*
- distance\_c\_relative - optional; type *double*
- count\_n - optional; type *int*
- varying\_load\_parameters\_are\_defined\_as\_relative - optional; type *boolean*
- varying\_load\_parameters - optional; type *array\_of\_line\_load\_varying\_load\_parameters*
  - line\_load\_varying\_load\_parameters - optional, unbounded; type *line\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
  - varying\_load\_parameters\_sorted - optional; type *boolean*
- reference\_to\_list\_of\_lines - optional; type *boolean*
- distance\_from\_line\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 47. get\_line\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line\_mesh\_refinement

**Input:** get\_line\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_line\_mesh\_refinement*

- no *type int*

**Output:** get\_line\_mesh\_refinement\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_line\_mesh\_refinementResponse*

- value type *line\_mesh\_refinement*
  - no *type int*
  - type - optional; type *line\_mesh\_refinement\_type* - type *undefined* with restriction - enum { 'TYPE\_ELEMENTS', 'TYPE\_GRADUAL', 'TYPE\_LENGTH' }
  - comment - optional; type *string*
  - elements\_finite\_elements - optional; type *int*
  - generating\_object\_info - optional; type *string*
  - gradual\_rows - optional; type *int*
  - is\_generated - optional; type *boolean*
  - lines - optional; type *array\_of\_int*
  - name - optional; type *string*
  - number\_of\_layers - optional; type *int*
  - target\_length - optional; type *double*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

#### 48. get\_line\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line\_set

**Input:** `get_line_set_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_set
 ■ no type int
```

**Output:** `get_line_set_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_setResponse
 ■ value type line_set
 ■ no type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ set_type - optional; type line_set_set_type - type undefined with restriction - enum { 'SET_TYPE_CONTINUOUS', 'SET_TYPE_GROUP' }
 ■ lines - optional; type array_of_int
 ■ length - optional; type double
 ■ center_of_gravity - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ center_of_gravity_x - optional; type double
 ■ center_of_gravity_y - optional; type double
 ■ center_of_gravity_z - optional; type double
 ■ position - optional; type string
 ■ position_short - optional; type string
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

#### 49. `get_line_set_load`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_line_set_load`

**Input:** `get_line_set_load_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_set_load
 ■ no type int
 ■ load_case_no type int
```

**Output:** `get_line_set_load_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_line_set_loadResponse
 ■ value type line_set_load
 ■ no type int
 ■ load_type - optional; type line_set_load_load_type - type undefined with restriction - enum { 'E_TYPE_MASS', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_MOMENT' }
 ■ line_sets - optional; type array_of_int
 ■ load_case - optional; type int
 ■ coordinate_system - optional; type string
 ■ load_distribution - optional; type line_set_load_load_distribution - type undefined with restriction - enum { 'LOAD_DISTRIBUTION_CONCENTRATED_1', 'LOAD_DISTRIBUTION_CONCENTRATED_2', 'LOAD_DISTRIBUTION_CONCENTRATED_2x2', 'LOAD_DISTRIBUTION_CONCENTRATED_N', 'LOAD_DISTRIBUTION_CONCENTRATED_VARYING', 'LOAD_DISTRIBUTION_PARABOLIC', 'LOAD_DISTRIBUTION_TAPERED', 'LOAD_DISTRIBUTION_TRAPEZOIDAL', 'LOAD_DISTRIBUTION_UNIFORM', 'LOAD_DISTRIBUTION_UNIFORM_TOTAL', 'LOAD_DISTRIBUTION_VARYING' }
 ■ load_direction - optional; type line_set_load_load_direction - type undefined with restriction - enum { 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_PROJECTED', 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE', 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE', 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE', 'LOAD_DIRECTION_LOCAL_X', 'LOAD_DIRECTION_LOCAL_Y', 'LOAD_DIRECTION_LOCAL_Z' }
 ■ load_direction_orientation - optional; type line_set_load_load_direction_orientation - type undefined with restriction - enum { 'LOAD_DIRECTION_FORWARD', 'LOAD_DIRECTION_REVERSED' }
 ■ magnitude - optional; type double
 ■ magnitude_1 - optional; type double
 ■ magnitude_2 - optional; type double
 ■ magnitude_3 - optional; type double
 ■ mass_global - optional; type double
 ■ mass_x - optional; type double
 ■ mass_y - optional; type double
 ■ mass_z - optional; type double
 ■ count_n - optional; type int
 ■ varying_load_parameters_are_defined_as_relative - optional; type boolean
 ■ varying_load_parameters - optional; type array_of_line_set_load_varying_load_parameters
 ■ line_set_load_varying_load_parameters - optional, unbounded; type line_set_load_varying_load_parameters
 ■ no - optional; type int
 ■ distance - optional; type double
 ■ delta_distance - optional; type double
 ■ magnitude - optional; type double
 ■ note - optional; type string
 ■ varying_load_parameters_sorted - optional; type boolean
 ■ distance_a_is_defined_as_relative - optional; type boolean
 ■ distance_a_absolute - optional; type double
 ■ distance_a_relative - optional; type double
 ■ distance_b_is_defined_as_relative - optional; type boolean
 ■ distance_b_absolute - optional; type double
 ■ distance_b_relative - optional; type double
 ■ distance_c_is_defined_as_relative - optional; type boolean
 ■ distance_c_absolute - optional; type double
 ■ distance_c_relative - optional; type double
 ■ reference_to_list_of_line_sets - optional; type boolean
```

- distance\_from\_line\_set\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- individual\_mass\_components - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 50. get\_line\_support

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_line\\_support](http://localhost:8082/get_line_support)

**Input:** [get\\_line\\_support\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_support*

- no type *int*

**Output:** [get\\_line\\_support\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_supportResponse*

- value type *line\_support*
  - no type *int*
  - comment - optional; type *string*
  - coordinate\_system - optional; type *line\_support\_coordinate\_system* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_GLOBAL', 'COORDINATE\_SYSTEM\_LOCAL' }
  - diagram\_along\_x\_end - optional; type *line\_support\_diagram\_along\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_x\_is\_sorted - optional; type *boolean*
  - diagram\_along\_x\_start - optional; type *line\_support\_diagram\_along\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_x\_symmetric - optional; type *boolean*
  - diagram\_along\_x\_table - optional; type *array\_of\_line\_support\_diagram\_along\_x\_table*
    - line\_support\_diagram\_along\_x\_table - optional, unbounded; type *line\_support\_diagram\_along\_x\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_along\_y\_end - optional; type *line\_support\_diagram\_along\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_y\_is\_sorted - optional; type *boolean*
  - diagram\_along\_y\_start - optional; type *line\_support\_diagram\_along\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_y\_symmetric - optional; type *boolean*
  - diagram\_along\_y\_table - optional; type *array\_of\_line\_support\_diagram\_along\_y\_table*
    - line\_support\_diagram\_along\_y\_table - optional, unbounded; type *line\_support\_diagram\_along\_y\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_along\_z\_end - optional; type *line\_support\_diagram\_along\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_z\_is\_sorted - optional; type *boolean*
  - diagram\_along\_z\_start - optional; type *line\_support\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_z\_symmetric - optional; type *boolean*
  - diagram\_along\_z\_table - optional; type *array\_of\_line\_support\_diagram\_along\_z\_table*
    - line\_support\_diagram\_along\_z\_table - optional, unbounded; type *line\_support\_diagram\_along\_z\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_around\_x\_end - optional; type *line\_support\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_around\_x\_is\_sorted - optional; type *boolean*
  - diagram\_around\_x\_start - optional; type *line\_support\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_around\_x\_symmetric - optional; type *boolean*
  - diagram\_around\_x\_table - optional; type *array\_of\_line\_support\_diagram\_around\_x\_table*
    - line\_support\_diagram\_around\_x\_table - optional, unbounded; type *line\_support\_diagram\_around\_x\_table*
      - no - optional; type *int*
      - rotation - optional; type *double*
      - moment - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_around\_y\_end - optional; type *line\_support\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_around\_y\_is\_sorted - optional; type *boolean*
  - diagram\_around\_y\_start - optional; type *line\_support\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_around\_y\_symmetric - optional; type *boolean*
  - diagram\_around\_y\_table - optional; type *array\_of\_line\_support\_diagram\_around\_y\_table*



- `partial_activity_around_z_negative_rotation` - optional; type *double*
- `partial_activity_around_z_negative_slippage` - optional; type *double*
- `partial_activity_around_z_negative_type` - optional; type *line\_support\_partial\_activity\_around\_z\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_z_positive_moment` - optional; type *double*
- `partial_activity_around_z_positive_rotation` - optional; type *double*
- `partial_activity_around_z_positive_slippage` - optional; type *double*
- `partial_activity_around_z_positive_type` - optional; type *line\_support\_partial\_activity\_around\_z\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `rotational_restraint` - optional; type *vector\_3d*
  - `x` type *double*
  - `y` type *double*
  - `z` type *double*
- `rotational_restraint_x` - optional; type *double*
- `rotational_restraint_x_nonlinearity` - optional; type *line\_support\_rotational\_restraint\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `rotational_restraint_y` - optional; type *double*
- `rotational_restraint_y_nonlinearity` - optional; type *line\_support\_rotational\_restraint\_y\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `rotational_restraint_z` - optional; type *double*
- `rotational_restraint_z_nonlinearity` - optional; type *line\_support\_rotational\_restraint\_z\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `spring` - optional; type *vector\_3d*
  - `x` type *double*
  - `y` type *double*
  - `z` type *double*
- `spring_x` - optional; type *double*
- `spring_x_nonlinearity` - optional; type *line\_support\_spring\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `spring_y` - optional; type *double*
- `spring_y_nonlinearity` - optional; type *line\_support\_spring\_y\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `spring_z` - optional; type *double*
- `spring_z_nonlinearity` - optional; type *line\_support\_spring\_z\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `user_defined_name_enabled` - optional; type *boolean*
- `x_axis_rotation` - optional; type *double*
- `is_generated` - optional; type *boolean*
- `generating_object_info` - optional; type *string*
- `fictionous_wall_enabled` - optional; type *boolean*
- `fictionous_wall_width` - optional; type *double*
- `fictionous_wall_height` - optional; type *double*
- `fictionous_wall_head_support_type` - optional; type *line\_support\_fictionous\_wall\_head\_support\_type* - type *undefined* with restriction - enum { 'HEAD\_SUPPORT\_TYPE\_HINGED', 'HEAD\_SUPPORT\_TYPE\_SEMI\_RIGID' }
- `fictionous_wall_base_support_type` - optional; type *line\_support\_fictionous\_wall\_base\_support\_type* - type *undefined* with restriction - enum { 'BASE\_SUPPORT\_TYPE\_ELASTIC', 'BASE\_SUPPORT\_TYPE\_HINGED', 'BASE\_SUPPORT\_TYPE\_RIGID' }
- `fictionous_wall_base_elastic` - optional; type *double*
- `fictionous_wall_shear_stiffness` - optional; type *boolean*
- `fictionous_wall_material` - optional; type *int*
- `fictionous_wall_spring_x` - optional; type *double*
- `fictionous_wall_spring_y` - optional; type *double*
- `fictionous_wall_spring_z` - optional; type *double*
- `fictionous_wall_rotational_restraint_about_line_axis` - optional; type *double*
- `id_for_export_import` - optional; type *string*
- `metadata_for_export_import` - optional; type *string*

## 51. get\_line\_welded\_joint

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_line\_welded\_joint

**Input:** get\_line\_welded\_joint\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_welded\_joint*

- no type *int*

**Output:** get\_line\_welded\_joint\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_line\_welded\_jointResponse*

- value type *line\_welded\_joint*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - joint\_type - optional; type *line\_welded\_joint\_joint\_type* - type *undefined* with restriction - enum { 'BUTT\_JOINT', 'CORNER\_JOINT', 'LAP\_JOINT', 'TEE\_JOINT' }
  - weld\_type - optional; type *line\_welded\_joint\_weld\_type* - type *undefined* with restriction - enum { 'WELD\_BEVEL\_AND\_FILLET', 'WELD\_DOUBLE\_BEVEL', 'WELD\_DOUBLE\_FILLET', 'WELD\_DOUBLE\_J', 'WELD\_DOUBLE\_SQUARE', 'WELD\_DOUBLE\_U', 'WELD\_DOUBLE\_V', 'WELD\_J\_AND\_FILLET', 'WELD\_SINGLE\_BEVEL', 'WELD\_SINGLE\_FILLET', 'WELD\_SINGLE\_J', 'WELD\_SINGLE\_SQUARE', 'WELD\_SINGLE\_U', 'WELD\_SINGLE\_V', 'WELD\_V\_AND\_FILLET' }
  - longitudinal\_arrangement - optional; type *line\_welded\_joint\_longitudinal\_arrangement* - type *undefined* with restriction - enum { 'CHAIN\_INTERMITTENT', 'CONTINUOUS', 'STAGGERED\_INTERMITTENT' }
  - weld\_size\_a1 - optional; type *double*
  - weld\_size\_a2 - optional; type *double*
  - weld\_length - optional; type *double*
  - pitch - optional; type *double*
  - first\_weld\_position - optional; type *double*
  - stress\_analysis\_configuration - optional; type *int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 52. get\_list\_of\_parameters\_formula\_allowed\_for

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_list\_of\_parameters\_formula\_allowed\_for

**Input:** get\_list\_of\_parameters\_formula\_allowed\_for\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_list\_of\_parameters\_formula\_allowed\_for*

- object\_location type *object\_location*
  - type type *object\_types* - type *undefined* with restriction - enum { 'E\_OBJECT\_TYPE\_ACTION', 'E\_OBJECT\_TYPE\_ACTION\_COMBINATION', 'E\_OBJECT\_TYPE\_BUILDING\_STORY', 'E\_OBJECT\_TYPE\_CLIPPING\_BOX', 'E\_OBJECT\_TYPE\_CLIPPING\_PLANE', 'E\_OBJECT\_TYPE\_COMBINATION\_WIZARD', 'E\_OBJECT\_TYPE\_COORDINATE\_SYSTEM', 'E\_OBJECT\_TYPE\_CUTTING\_LINE\_SETTING', 'E\_OBJECT\_TYPE\_CUTTING\_PATTERN', 'E\_OBJECT\_TYPE\_DESIGN\_SITUATION', 'E\_OBJECT\_TYPE\_DESIGN\_SUPPORT', 'E\_OBJECT\_TYPE\_DIMENSION', 'E\_OBJECT\_TYPE\_FREE\_CIRCULAR\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_CONCENTRATED\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_POLYGON\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_RECTANGULAR\_LOAD', 'E\_OBJECT\_TYPE\_GLOBAL\_PARAMETER', 'E\_OBJECT\_TYPE\_IMPERFECTION\_CASE', 'E\_OBJECT\_TYPE\_IMPOSED\_LINE\_DEFORMATION', 'E\_OBJECT\_TYPE\_IMPOSED\_NODAL\_DEFORMATION', 'E\_OBJECT\_TYPE\_INTERSECTION', 'E\_OBJECT\_TYPE\_LINE', 'E\_OBJECT\_TYPE\_LINE\_GRID', 'E\_OBJECT\_TYPE\_LINE\_HINGE', 'E\_OBJECT\_TYPE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_LINE\_SET', 'E\_OBJECT\_TYPE\_LINE\_SET\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_SUPPORT', 'E\_OBJECT\_TYPE\_LINE\_WELDED\_JOINT', 'E\_OBJECT\_TYPE\_LOAD\_CASE', 'E\_OBJECT\_TYPE\_LOAD\_COMBINATION', 'E\_OBJECT\_TYPE\_MATERIAL', 'E\_OBJECT\_TYPE\_MEMBER', 'E\_OBJECT\_TYPE\_MEMBER\_DEFINABLE\_STIFFNESS', 'E\_OBJECT\_TYPE\_MEMBER\_ECCENTRICITY', 'E\_OBJECT\_TYPE\_MEMBER\_HINGE', 'E\_OBJECT\_TYPE\_MEMBER\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_NONLINEARITY', 'E\_OBJECT\_TYPE\_MEMBER\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_RESULT\_INTERMEDIATE\_POINT', 'E\_OBJECT\_TYPE\_MEMBER\_SET', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_STIFFNESS\_MODIFICATION', 'E\_OBJECT\_TYPE\_MEMBER\_SUPPORT', 'E\_OBJECT\_TYPE\_MEMBER\_TRANSVERSE\_STIFFENER', 'E\_OBJECT\_TYPE\_NODAL\_LOAD', 'E\_OBJECT\_TYPE\_NODAL\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_NODAL\_SUPPORT', 'E\_OBJECT\_TYPE\_NODE', 'E\_OBJECT\_TYPE\_NOTE', 'E\_OBJECT\_TYPE\_OBJECT\_SNAP', 'E\_OBJECT\_TYPE\_OPENING', 'E\_OBJECT\_TYPE\_OPENING\_LOAD', 'E\_OBJECT\_TYPE\_RESULT\_COMBINATION', 'E\_OBJECT\_TYPE\_RESULT\_SECTION', 'E\_OBJECT\_TYPE\_RIGID\_LINK', 'E\_OBJECT\_TYPE\_SECTION', 'E\_OBJECT\_TYPE\_SOIL\_MASSIF', 'E\_OBJECT\_TYPE\_SOIL\_SAMPLE', 'E\_OBJECT\_TYPE\_SOLID', 'E\_OBJECT\_TYPE\_SOLID\_CONTACTS', 'E\_OBJECT\_TYPE\_SOLID\_GAS', 'E\_OBJECT\_TYPE\_SOLID\_LOAD', 'E\_OBJECT\_TYPE\_SOLID\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_SOLID\_SET', 'E\_OBJECT\_TYPE\_SOLID\_SET\_LOAD', 'E\_OBJECT\_TYPE\_SPECTRAL\_ANALYSIS\_SETTINGS', 'E\_OBJECT\_TYPE\_STATIC\_ANALYSIS\_SETTINGS', 'E\_OBJECT\_TYPE\_STRUCTURE\_MODIFICATION', 'E\_OBJECT\_TYPE\_SURFACE', 'E\_OBJECT\_TYPE\_SURFACES\_CONTACT', 'E\_OBJECT\_TYPE\_SURFACES\_CONTACT\_TYPE', 'E\_OBJECT\_TYPE\_SURFACE\_ECCENTRICITY', 'E\_OBJECT\_TYPE\_SURFACE\_IMPERFECTION', 'E\_OBJECT\_TYPE\_SURFACE\_LOAD', 'E\_OBJECT\_TYPE\_SURFACE\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_SURFACE\_RESULTS\_ADJUSTMENT', 'E\_OBJECT\_TYPE\_SURFACE\_SET', 'E\_OBJECT\_TYPE\_SURFACE\_SET\_IMPERFECTION', 'E\_OBJECT\_TYPE\_SURFACE\_SET\_LOAD', 'E\_OBJECT\_TYPE\_SURFACE\_STIFFNESS\_MODIFICATION', 'E\_OBJECT\_TYPE\_SURFACE\_SUPPORT', 'E\_OBJECT\_TYPE\_TERRAIN', 'E\_OBJECT\_TYPE\_THICKNESS', 'E\_OBJECT\_TYPE\_VISUAL\_OBJECT', 'E\_OBJECT\_TYPE\_WIND\_PROFILE', 'E\_OBJECT\_TYPE\_WIND\_SIMULATION', 'E\_OBJECT\_TYPE\_WIND\_SIMULATION\_ANALYSIS\_SETTINGS' }
  - no type *int*
  - parent\_no - optional; type *int*

**Output:** get\_list\_of\_parameters\_formula\_allowed\_for\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_list\_of\_parameters\_formula\_allowed\_forResponse*

- value type *object\_parameter\_location\_array\_type*
  - object\_parameter\_location - optional, unbounded; type *object\_parameter\_location\_type*
    - attribute type *string*



- `parameter_path_in_nested_models_hierarchy` - optional; type `parameter_path_in_nested_models_hierarchy_type`
  - `node` - optional, unbounded; type `node_of_parameter_path_in_nested_models_hierarchy_type`
    - `row_path` type `string`
    - `column_string_id` type `string`

### 53. `get_load_case`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_load_case`

**Input:** `get_load_case_request` (soap:body, use = literal) [Source code](#)

**parameters** type `get_load_case`

- no type `int`

**Output:** `get_load_case_response` (soap:body, use = literal) [Source code](#)

**parameters** type `get_load_caseResponse`

- value type `load_case`
  - no type `int`
  - `analysis_type` - optional; type `load_case_analysis_type` - type `undefined` with restriction - enum { 'ANALYSIS\_TYPE\_CREEP\_AND\_SHRINKAGE', 'ANALYSIS\_TYPE\_CUTTING\_PATTERN', 'ANALYSIS\_TYPE\_MODAL', 'ANALYSIS\_TYPE\_RESPONSE\_SPECTRUM', 'ANALYSIS\_TYPE\_STATIC', 'ANALYSIS\_TYPE\_TIME\_DEPENDENT', 'ANALYSIS\_TYPE\_TIME\_HISTORY', 'ANALYSIS\_TYPE\_WIND\_SIMULATION' }
  - `name` - optional; type `string`
  - `static_analysis_settings` - optional; type `int`
  - `stability_analysis_settings` - optional; type `int`
  - `modal_analysis_settings` - optional; type `int`
  - `spectral_analysis_settings` - optional; type `int`
  - `calculate_critical_load` - optional; type `boolean`
  - `consider_imperfection` - optional; type `boolean`
  - `imperfection_case` - optional; type `int`
  - `consider_initial_state` - optional; type `boolean`
  - `initial_state_case` - optional; type `int`
  - `initial_state_definition_type` - optional; type `load_case_initial_state_definition_type` - type `undefined` with restriction - enum { 'DEFINITION\_TYPE\_FINAL\_STATE', 'DEFINITION\_TYPE\_STIFFNESS', 'DEFINITION\_TYPE\_STRAINS', 'DEFINITION\_TYPE\_STRAINS\_WITH\_USER\_DEFINED\_FACTORS' }
  - `creep_loading_case` - optional; type `int`
  - `individual_factors_of_selected_objects_table` - optional; type `array_of_load_case_individual_factors_of_selected_objects_table`
    - `load_case_individual_factors_of_selected_objects_table` - optional, unbounded; type `load_case_individual_factors_of_selected_objects_table`
      - no - optional; type `int`
      - `object_type` - optional; type `object_type` - type `undefined` with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
      - `object_list` - optional; type `array_of_int`
      - `strain_type` - optional; type `strain_type` - type `undefined` with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
      - `factor` - optional; type `double`
      - `comment` - optional; type `string`
  - `consider_construction_stage` - optional; type `boolean`
  - `construction_stage_case` - optional; type `int`
  - `to_solve` - optional; type `boolean`
  - `action_category` - optional; type `string`
  - `self_weight_active` - optional; type `boolean`
  - `self_weight_factors` - optional; type `vector_3d`
    - `x` type `double`
    - `y` type `double`
    - `z` type `double`
  - `self_weight_factor_x` - optional; type `double`
  - `self_weight_factor_y` - optional; type `double`
  - `self_weight_factor_z` - optional; type `double`
  - `factor_phi` - optional; type `load_case_factor_phi` - type `undefined` with restriction - enum { 'FACTOR\_PHI\_1', 'FACTOR\_PHI\_2' }
  - `load_duration` - optional; type `int`
  - `loading_start` - optional; type `double`
  - `time_being_investigated` - optional; type `double`
  - `has_inclusive_load_cases` - optional; type `boolean`
  - `inclusive_load_cases` - optional; type `array_of_load_case_inclusive_load_cases`
    - `load_case_inclusive_load_cases` - optional, unbounded; type `int`
  - `has_exclusive_load_cases` - optional; type `boolean`
  - `exclusive_load_cases` - optional; type `array_of_load_case_exclusive_load_cases`
    - `load_case_exclusive_load_cases` - optional, unbounded; type `int`
  - `import_masses_from` - optional; type `int`
  - `import_modal_analysis_from` - optional; type `int`
  - `response_spectrum_is_enabled_in_any_direction` - optional; type `boolean`
  - `response_spectrum_is_enabled_in_direction_x` - optional; type `boolean`
  - `response_spectrum_is_enabled_in_direction_y` - optional; type `boolean`
  - `response_spectrum_is_enabled_in_direction_z` - optional; type `boolean`
  - `response_spectrum_in_direction_x` - optional; type `int`
  - `response_spectrum_in_direction_y` - optional; type `int`
  - `response_spectrum_in_direction_z` - optional; type `int`
  - `response_spectrum_scale_factor_in_direction_x` - optional; type `double`
  - `response_spectrum_scale_factor_in_direction_y` - optional; type `double`
  - `response_spectrum_scale_factor_in_direction_z` - optional; type `double`
  - `response_spectrum_rotation_angle` - optional; type `double`
  - `response_spectrum_consider_accidental_torsion` - optional; type `boolean`
  - `response_spectrum_eccentricity_for_x_direction_absolute` - optional; type `double`
  - `response_spectrum_eccentricity_for_y_direction_absolute` - optional; type `double`

- response\_spectrum\_eccentricity\_for\_x\_direction\_relative - optional; type *double*
- response\_spectrum\_eccentricity\_for\_y\_direction\_relative - optional; type *double*
- response\_spectrum\_building\_length\_in\_x - optional; type *double*
- response\_spectrum\_building\_length\_in\_y - optional; type *double*
- response\_spectrum\_eccentricity\_for\_x\_direction\_is\_defined\_as\_relative - optional; type *boolean*
- response\_spectrum\_eccentricity\_for\_y\_direction\_is\_defined\_as\_relative - optional; type *boolean*
- response\_spectrum\_user\_defined\_building\_lengths - optional; type *boolean*
- selection\_of\_modes\_mode\_activation\_table - optional; type *array\_of\_load\_case\_selection\_of\_modes\_mode\_activation\_table*
  - load\_case\_selection\_of\_modes\_mode\_activation\_table - optional, unbounded; type *load\_case\_selection\_of\_modes\_mode\_activation\_table*
    - no - optional; type *int*
    - to\_generate - optional; type *boolean*
    - natural\_period\_t - optional; type *double*
    - natural\_frequency\_f - optional; type *double*
    - acceleration\_s\_ax - optional; type *double*
    - acceleration\_s\_ay - optional; type *double*
    - acceleration\_s\_az - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mex - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mey - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mez - optional; type *double*
    - damping - optional; type *double*
- selection\_of\_modes\_deselect\_modes\_according\_to\_criterion\_is\_enabled - optional; type *boolean*
- selection\_of\_modes\_deselect\_modes\_according\_to\_criterion\_value - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mex - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mey - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mez - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- structure\_modification\_enabled - optional; type *boolean*
- structure\_modification - optional; type *int*
- possibility\_of\_crowds - optional; type *boolean*
- specification\_for\_load\_case\_gr1a - optional; type *load\_case\_specification\_for\_load\_case\_gr1a* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR1A\_PEDESTRIAN\_AND\_CYCLE\_TRACK', 'SPECIFICATION\_GR1A\_TS1', 'SPECIFICATION\_GR1A\_UDL' }
- specification\_for\_load\_case\_gr2 - optional; type *load\_case\_specification\_for\_load\_case\_gr2* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR2\_HORIZONTAL\_FORCES\_BRAKING\_AND\_ACCELERATION', 'SPECIFICATION\_GR2\_HORIZONTAL\_FORCES\_CENTRIFUGAL\_FORCES' }
- specification\_for\_load\_case\_gr5 - optional; type *load\_case\_specification\_for\_load\_case\_gr5* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR5\_SPECIAL\_VEHICLE', 'SPECIFICATION\_GR5\_TS', 'SPECIFICATION\_GR5\_UDL' }
- specification\_for\_load\_case\_gr6 - optional; type *load\_case\_specification\_for\_load\_case\_gr6* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR6\_HORIZONTAL\_FORCES\_BRAKING\_AND\_ACCELERATION', 'SPECIFICATION\_GR6\_HORIZONTAL\_FORCES\_CENTRIFUGAL\_FORCES', 'SPECIFICATION\_GR6\_PEDESTRIAN\_AND\_CYCLE\_TRACK', 'SPECIFICATION\_GR6\_TS', 'SPECIFICATION\_GR6\_UDL' }
- liveload\_less\_equal\_100 - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- liveload\_consider\_design\_situation\_seismic\_weight\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_seismic\_weight\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_seismic\_weight\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LOADS\_ABOVE\_3', 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LOADS\_UP\_TO\_3' }
- liveload\_consider\_design\_situation\_seismic\_mass\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_seismic\_mass\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_seismic\_mass\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_ACTUAL\_CONDITIONS', 'EFFECTIVE\_SEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_EQUIVALENT\_UNIFORM\_LIBRARY\_ARCHIVES', 'EFFECTIVE\_SEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_EQUIVALENT\_UNIFORM\_OTHER\_CIVIL\_BUILDINGS' }
- consider\_design\_situation\_effective\_seismic\_weight - optional; type *boolean*
- liveload\_less\_equal\_48 - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- liveload\_less\_equal\_100\_ibc - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- cranehookload\_nocombined\_roofliveload - optional; type *boolean*
- roof\_configurations\_that\_do\_not\_shed\_snow - optional; type *boolean*
- flat\_roof\_snow\_load - optional; type *boolean*
- flat\_roof\_snow\_load\_category - optional; type *load\_case\_flat\_roof\_snow\_load\_category* - type *undefined* with restriction - enum { 'ROOF\_CAT\_SNOW\_LOADS\_EXCEEDING\_30PSF', 'ROOF\_CAT\_SNOW\_LOADS\_OF\_30PSF\_OR\_LESS' }
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- deadload\_factor\_shall\_be\_increased - optional; type *boolean*
- deadload\_factor\_shall\_be\_increased1 - optional; type *boolean*
- deadload\_factor\_shall\_be\_increased2 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight\_NBC05 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight\_NBC15 - optional; type *boolean*
- liveload\_principal\_factor\_may\_be\_reduced1 - optional; type *boolean*
- liveload\_companion\_factor\_may\_be\_increased1 - optional; type *boolean*

- liveload\_principal\_factor\_may\_be\_reduced2 - optional; type *boolean*
- liveload\_companion\_factor\_may\_be\_increased2 - optional; type *boolean*
- consider\_design\_situation\_effective\_seismic\_weight\_NBC05 - optional; type *boolean*
- consider\_design\_situation\_effective\_seismic\_weight\_NBC15 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight - optional; type *boolean*
- longtermload\_tp\_considered\_in\_limit\_state\_of\_serviceability - optional; type *boolean*
- greater\_intensity\_than\_4 - optional; type *boolean*
- distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_check - optional; type *boolean*
- distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_combo - optional; type *load\_case\_distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_combo* - type *undefined* with restriction - enum { 'EOVERSTRENGTH\_FACTOR\_DIRECTION\_X', 'EOVERSTRENGTH\_FACTOR\_DIRECTION\_Y' }
- consider\_design\_situation\_seismic\_mass\_combination - optional; type *boolean*
- wind\_simulation\_wind\_direction\_angle - optional; type *double*
- wind\_simulation\_terrain\_offset - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_minus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_plus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_minus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_plus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_plus\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_depth\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_height\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_width\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_coefficient - optional; type *double*
- wind\_simulation\_wind\_profile - optional; type *int*
- wind\_simulation\_analysis\_settings - optional; type *int*
- wind\_simulation\_wind\_tunnel\_depth\_minus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_plus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_minus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_plus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_plus\_length - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_depth - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_height - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_width - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height - optional; type *double*
- geotechnical\_analysis\_reset\_small\_strain\_state\_variables - optional; type *boolean*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 54. get\_load\_cases\_and\_combinations

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_load\_cases\_and\_combinations

**Input:** get\_load\_cases\_and\_combinations\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_load\_cases\_and\_combinations*

**Output:** get\_load\_cases\_and\_combinations\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_load\_cases\_and\_combinationsResponse*

- value type *load\_cases\_and\_combinations*
  - current\_standard\_for\_combination\_wizard - optional; type *int*
  - activate\_combination\_wizard\_and\_classification - optional; type *boolean*
  - activate\_combination\_wizard - optional; type *boolean*
  - result\_combinations\_active - optional; type *boolean*
  - result\_combinations\_parentheses\_active - optional; type *boolean*
  - result\_combinations\_consider\_sub\_results - optional; type *boolean*
  - combination\_name\_according\_to\_action\_category - optional; type *boolean*

#### 55. get\_load\_combination

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_load\_combination

**Input:** get\_load\_combination\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_load\_combination*

- no type *int*

**Output:** get\_load\_combination\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_load\_combinationResponse*

- value type *load\_combination*
  - no type *int*
  - analysis\_type - optional; type *load\_combination\_analysis\_type* - type *undefined* with restriction - enum { 'ANALYSIS\_TYPE\_HARMONIC\_RESPONSE\_ANALYSIS', 'ANALYSIS\_TYPE\_STATIC', 'ANALYSIS\_TYPE\_STATIC\_CREEP\_AND\_SHRINKAGE', 'ANALYSIS\_TYPE\_STATIC\_TIME\_DEPENDENCE' }
  - design\_situation - optional; type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - static\_analysis\_settings - optional; type *int*
  - import\_modal\_analysis\_load\_case - optional; type *int*
  - calculate\_critical\_load - optional; type *boolean*
  - stability\_analysis\_settings - optional; type *int*
  - consider\_imperfection - optional; type *boolean*
  - imperfection\_case - optional; type *int*
  - consider\_initial\_state - optional; type *boolean*
  - initial\_state\_case - optional; type *int*
  - consider\_construction\_stage - optional; type *boolean*
  - construction\_stage - optional; type *int*

- `consider_creep_caused_by_permanent_loading` - optional; type *boolean*
- `creep_caused_by_permanent_loading_case` - optional; type *int*
- `sustained_load_enabled` - optional; type *boolean*
- `sustained_load` - optional; type *int*
- `sway_load_enabled` - optional; type *boolean*
- `sway_load` - optional; type *int*
- `structure_modification_enabled` - optional; type *boolean*
- `structure_modification` - optional; type *int*
- `to_solve` - optional; type *boolean*
- `comment` - optional; type *string*
- `load_duration` - optional; type *int*
- `items` - optional; type *array\_of\_load\_combination\_items*
  - `load_combination_items` - optional, unbounded; type *load\_combination\_items*
    - `no` - optional; type *int*
    - `factor` - optional; type *double*
    - `load_case` - optional; type *int*
    - `action` - optional; type *int*
    - `is_leading` - optional; type *boolean*
    - `gamma` - optional; type *double*
    - `psi` - optional; type *double*
    - `xi` - optional; type *double*
    - `k_fi` - optional; type *double*
    - `c_esl` - optional; type *double*
    - `k_def` - optional; type *double*
    - `psi_0` - optional; type *double*
    - `psi_1` - optional; type *double*
    - `psi_2` - optional; type *double*
    - `fi` - optional; type *double*
    - `gamma_0` - optional; type *double*
    - `alfa` - optional; type *double*
    - `k_f` - optional; type *double*
    - `phi` - optional; type *double*
    - `rho` - optional; type *double*
    - `omega_0` - optional; type *double*
    - `gamma_l_1` - optional; type *double*
    - `k_creep` - optional; type *double*
    - `shift` - optional; type *double*
    - `amplitude_function_type` - optional; type *amplitude\_function\_type* - type *undefined* with restriction - enum { 'CONSTANT', 'LINEAR', 'QUADRATIC' }
- `loading_start` - optional; type *double*
- `time_being_investigated` - optional; type *double*
- `is_generated` - optional; type *boolean*
- `generating_object_info` - optional; type *string*
- `initial_state_definition_type` - optional; type *load\_combination\_initial\_state\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_FINAL\_STATE', 'DEFINITION\_TYPE\_STIFFNESS', 'DEFINITION\_TYPE\_STRAINS', 'DEFINITION\_TYPE\_STRAINS\_WITH\_USER\_DEFINED\_FACTORS' }
- `individual_factors_of_selected_objects_table` - optional; type *array\_of\_load\_combination\_individual\_factors\_of\_selected\_objects\_table*
  - `load_combination_individual_factors_of_selected_objects_table` - optional, unbounded; type *load\_combination\_individual\_factors\_of\_selected\_objects\_table*
    - `no` - optional; type *int*
    - `object_type` - optional; type *object\_type* - type *undefined* with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
    - `object_list` - optional; type *array\_of\_int*
    - `strain_type` - optional; type *strain\_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
    - `factor` - optional; type *double*
    - `comment` - optional; type *string*
- `geotechnical_analysis_reset_small_strain_state_variables` - optional; type *boolean*
- `id_for_export_import` - optional; type *string*
- `metadata_for_export_import` - optional; type *string*

## 56. `get_material`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_material`

**Input:** `get_material_request` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_material*

- `no` type *int*

**Output:** `get_material_response` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_materialResponse*

- value type *material*
  - `no` type *int*
  - `material_type` - optional; type *material\_material\_type* - type *undefined* with restriction - enum { 'TYPE\_ALUMINUM', 'TYPE\_BASIC', 'TYPE\_CONCRETE', 'TYPE\_FABRIC', 'TYPE\_FOIL', 'TYPE\_GAS', 'TYPE\_GLASS', 'TYPE\_MASONRY', 'TYPE\_METAL', 'TYPE\_REINFORCING\_STEEL', 'TYPE\_SOIL', 'TYPE\_STEEL', 'TYPE\_TIMBER' }
  - `material_model` - optional; type *material\_material\_model* - type *undefined* with restriction - enum { 'MODEL\_ISOTROPIC\_DAMAGE\_2D\_3D', 'MODEL\_ISOTROPIC\_LINEAR\_ELASTIC', 'MODEL\_ISOTROPIC\_MASONRY\_PLASTIC\_2D', 'MODEL\_ISOTROPIC\_NONLINEAR\_ELASTIC\_1D', 'MODEL\_ISOTROPIC\_NONLINEAR\_ELASTIC\_2D\_3D', 'MODEL\_ISOTROPIC\_PLASTIC\_1D', 'MODEL\_ISOTROPIC\_PLASTIC\_2D\_3D', 'MODEL\_ISOTROPIC\_SOIL\_NONLINEAR\_ELASTIC\_3D', 'MODEL\_ISOTROPIC\_SOIL\_PLASTIC\_3D', 'MODEL\_ISOTROPIC\_TIMBER\_LINEAR\_ELASTIC\_MEMBERS', 'MODEL\_ORTHOTROPIC\_2D', 'MODEL\_ORTHOTROPIC\_3D', 'MODEL\_ORTHOTROPIC\_MASONRY\_PLASTIC\_2D', 'MODEL\_ORTHOTROPIC\_PLASTIC\_2D', 'MODEL\_ORTHOTROPIC\_PLASTIC\_3D', 'MODEL\_ORTHOTROPIC\_TIMBER\_LINEAR\_ELASTIC\_SURFACES' }

- application\_context - optional; type *material\_application\_context* - type *undefined* with restriction - enum { 'ALUMINUM\_DESIGN', 'COMBINATION\_WIZARD', 'CONCRETE\_DESIGN', 'CONCRETE\_FOUNDATION\_DESIGN', 'CRANEWAY\_DESIGN', 'DYNAMIC\_ANALYSIS', 'GEOTECHNICAL\_ANALYSIS', 'GLASS\_DESIGN', 'INVALID', 'INVALID\_DESIGN', 'LOAD\_WIZARD', 'MASONRY\_DESIGN', 'PIPING\_DESIGN', 'STEEL\_DESIGN', 'STEEL\_JOINT\_DESIGN', 'TIMBER\_DESIGN', 'TIMBER\_JOINT\_DESIGN', 'TOWER\_DESIGN' }
- diagram\_type - optional; type *material\_diagram\_type* - type *undefined* with restriction - enum { 'DIAGRAM\_TYPE\_BASIC', 'DIAGRAM\_TYPE\_BILINEAR', 'DIAGRAM\_TYPE\_STRESS\_STRAIN\_DIAGRAM' }
- user\_defined\_name\_enabled - optional; type *boolean*
- name - optional; type *string*
- user\_defined - optional; type *boolean*
- definition\_type - optional; type *material\_definition\_type* - type *undefined* with restriction - enum { 'DERIVED\_G', 'DERIVED\_NU', 'E\_G\_NO\_NU', 'E\_G\_NU', 'NONE' }
- stress\_failure\_hypothesis - optional; type *material\_stress\_failure\_hypothesis* - type *undefined* with restriction - enum { 'E\_STRESS\_FAILURE\_HYPOTHESIS\_OEDOMETRIC\_CONDITIONS\_WITH\_SMALL\_STRAIN\_STIFFNESS', 'STRESS\_FAILURE\_HYPOTHESIS\_DRUCKER\_PRAGER', 'STRESS\_FAILURE\_HYPOTHESIS\_MOHR\_COULOMB', 'STRESS\_FAILURE\_HYPOTHESIS\_OEDOMETRIC\_CONDITIONS', 'STRESS\_FAILURE\_HYPOTHESIS\_TRESCA', 'STRESS\_FAILURE\_HYPOTHESIS\_VON\_MISES' }
- is\_temperature\_dependent - optional; type *boolean*
- is\_dynamic\_increase\_factor - optional; type *boolean*
- has\_cost\_estimation - optional; type *boolean*
- optimization - optional; type *boolean*
- has\_emissions\_estimation - optional; type *boolean*
- reference\_temperature - optional; type *double*
- temperature\_properties\_sorted - optional; type *boolean*
- stiffness\_matrix\_editable - optional; type *boolean*
- stiffness\_modification - optional; type *boolean*
- stiffness\_modification\_type - optional; type *material\_stiffness\_modification\_type* - type *undefined* with restriction - enum { 'STIFFNESS\_MODIFICATION\_TYPE\_DIVISION', 'STIFFNESS\_MODIFICATION\_TYPE\_MULTIPLICATION' }
- has\_linear\_elastic\_with\_nonlinear\_criteria - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- members\_weight\_active - optional; type *boolean*
- members\_weight\_unit\_cost - optional; type *double*
- members\_weight\_unit - optional; type *material\_members\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- members\_weight\_quantity - optional; type *double*
- members\_weight\_cost - optional; type *double*
- members\_volume\_active - optional; type *boolean*
- members\_volume\_unit\_cost - optional; type *double*
- members\_volume\_unit - optional; type *material\_members\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- members\_volume\_quantity - optional; type *double*
- members\_volume\_cost - optional; type *double*
- members\_surface\_active - optional; type *boolean*
- members\_surface\_unit\_cost - optional; type *double*
- members\_surface\_unit - optional; type *material\_members\_surface\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- members\_surface\_quantity - optional; type *double*
- members\_surface\_cost - optional; type *double*
- surfaces\_weight\_active - optional; type *boolean*
- surfaces\_weight\_unit\_cost - optional; type *double*
- surfaces\_weight\_unit - optional; type *material\_surfaces\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- surfaces\_weight\_quantity - optional; type *double*
- surfaces\_weight\_cost - optional; type *double*
- surfaces\_volume\_active - optional; type *boolean*
- surfaces\_volume\_unit\_cost - optional; type *double*
- surfaces\_volume\_unit - optional; type *material\_surfaces\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- surfaces\_volume\_quantity - optional; type *double*
- surfaces\_volume\_cost - optional; type *double*
- surfaces\_top\_face\_active - optional; type *boolean*
- surfaces\_top\_face\_unit\_cost - optional; type *double*
- surfaces\_top\_face\_unit - optional; type *material\_surfaces\_top\_face\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_top\_face\_quantity - optional; type *double*
- surfaces\_top\_face\_cost - optional; type *double*
- surfaces\_area\_active - optional; type *boolean*
- surfaces\_area\_unit\_cost - optional; type *double*
- surfaces\_area\_unit - optional; type *material\_surfaces\_area\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_area\_quantity - optional; type *double*
- surfaces\_area\_cost - optional; type *double*
- surfaces\_bottom\_face\_active - optional; type *boolean*
- surfaces\_bottom\_face\_unit\_cost - optional; type *double*
- surfaces\_bottom\_face\_unit - optional; type *material\_surfaces\_bottom\_face\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_bottom\_face\_quantity - optional; type *double*
- surfaces\_bottom\_face\_cost - optional; type *double*
- solids\_weight\_active - optional; type *boolean*
- solids\_weight\_unit\_cost - optional; type *double*
- solids\_weight\_unit - optional; type *material\_solids\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- solids\_weight\_quantity - optional; type *double*
- solids\_weight\_cost - optional; type *double*

- solids\_volume\_active - optional; type *boolean*
- solids\_volume\_unit\_cost - optional; type *double*
- solids\_volume\_unit - optional; type *material\_solids\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- solids\_volume\_quantity - optional; type *double*
- solids\_volume\_cost - optional; type *double*
- solids\_area\_active - optional; type *boolean*
- solids\_area\_unit\_cost - optional; type *double*
- solids\_area\_unit - optional; type *material\_solids\_area\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- solids\_area\_quantity - optional; type *double*
- solids\_area\_cost - optional; type *double*
- sum\_weight - optional; type *double*
- weight\_percentage - optional; type *double*
- cost\_percentage - optional; type *double*
- sum\_cost - optional; type *double*
- total\_cost - optional; type *double*
- emissions\_members\_weight\_active - optional; type *boolean*
- emissions\_members\_weight\_unit\_cost - optional; type *double*
- emissions\_members\_weight\_unit - optional; type *material\_emissions\_members\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG', 'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }
- emissions\_members\_weight\_quantity - optional; type *double*
- emissions\_members\_weight\_cost - optional; type *double*
- emissions\_members\_volume\_active - optional; type *boolean*
- emissions\_members\_volume\_unit\_cost - optional; type *double*
- emissions\_members\_volume\_unit - optional; type *material\_emissions\_members\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMP GAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_members\_volume\_quantity - optional; type *double*
- emissions\_members\_volume\_cost - optional; type *double*
- emissions\_members\_surface\_active - optional; type *boolean*
- emissions\_members\_surface\_unit\_cost - optional; type *double*
- emissions\_members\_surface\_unit - optional; type *material\_emissions\_members\_surface\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_members\_surface\_quantity - optional; type *double*
- emissions\_members\_surface\_cost - optional; type *double*
- emissions\_surfaces\_weight\_active - optional; type *boolean*
- emissions\_surfaces\_weight\_unit\_cost - optional; type *double*
- emissions\_surfaces\_weight\_unit - optional; type *material\_emissions\_surfaces\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG', 'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }
- emissions\_surfaces\_weight\_quantity - optional; type *double*
- emissions\_surfaces\_weight\_cost - optional; type *double*
- emissions\_surfaces\_volume\_active - optional; type *boolean*
- emissions\_surfaces\_volume\_unit\_cost - optional; type *double*
- emissions\_surfaces\_volume\_unit - optional; type *material\_emissions\_surfaces\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMP GAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_surfaces\_volume\_quantity - optional; type *double*
- emissions\_surfaces\_volume\_cost - optional; type *double*
- emissions\_surfaces\_top\_face\_active - optional; type *boolean*
- emissions\_surfaces\_top\_face\_unit\_cost - optional; type *double*
- emissions\_surfaces\_top\_face\_unit - optional; type *material\_emissions\_surfaces\_top\_face\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_top\_face\_quantity - optional; type *double*
- emissions\_surfaces\_top\_face\_cost - optional; type *double*
- emissions\_surfaces\_area\_active - optional; type *boolean*
- emissions\_surfaces\_area\_unit\_cost - optional; type *double*
- emissions\_surfaces\_area\_unit - optional; type *material\_emissions\_surfaces\_area\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_area\_quantity - optional; type *double*
- emissions\_surfaces\_area\_cost - optional; type *double*
- emissions\_surfaces\_bottom\_face\_active - optional; type *boolean*
- emissions\_surfaces\_bottom\_face\_unit\_cost - optional; type *double*
- emissions\_surfaces\_bottom\_face\_unit - optional; type *material\_emissions\_surfaces\_bottom\_face\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_bottom\_face\_quantity - optional; type *double*
- emissions\_surfaces\_bottom\_face\_cost - optional; type *double*
- emissions\_solids\_weight\_active - optional; type *boolean*
- emissions\_solids\_weight\_unit\_cost - optional; type *double*
- emissions\_solids\_weight\_unit - optional; type *material\_emissions\_solids\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG', 'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }
- emissions\_solids\_weight\_quantity - optional; type *double*
- emissions\_solids\_weight\_cost - optional; type *double*
- emissions\_solids\_volume\_active - optional; type *boolean*
- emissions\_solids\_volume\_unit\_cost - optional; type *double*

- emissions\_solid\_volume\_unit - optional; type *material\_emissions\_solid\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMPGAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_solid\_volume\_quantity - optional; type *double*
- emissions\_solid\_volume\_cost - optional; type *double*
- emissions\_solid\_area\_active - optional; type *boolean*
- emissions\_solid\_area\_unit\_cost - optional; type *double*
- emissions\_solid\_area\_unit - optional; type *material\_emissions\_solid\_area\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_solid\_area\_quantity - optional; type *double*
- emissions\_solid\_area\_cost - optional; type *double*
- emissions\_percentage - optional; type *double*
- emissions\_sum - optional; type *double*
- emissions\_total - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 57. get\_member

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_member

**Input:** get\_member\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_member*

- no type *int*

**Output:** get\_member\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_memberResponse*

- value type *member*
  - no type *int*
  - type - optional; type *member\_type* - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE', 'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }
  - is\_deactivated\_for\_calculation - optional; type *boolean*
  - line - optional; type *int*
  - section\_distribution\_type - optional; type *member\_section\_distribution\_type* - type *undefined* with restriction - enum { 'SECTION\_DISTRIBUTION\_TYPE\_LINEAR', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
  - reference\_type - optional; type *member\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - nodes - optional; type *array\_of\_int*
  - node\_start - optional; type *int*
  - node\_end - optional; type *int*
  - analytical\_length - optional; type *double*
  - analytical\_volume - optional; type *double*
  - analytical\_surface\_of\_coating - optional; type *double*
  - analytical\_mass - optional; type *double*
  - surface\_of\_coating - optional; type *double*
  - analytical\_center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - analytical\_center\_of\_gravity\_x - optional; type *double*
  - analytical\_center\_of\_gravity\_y - optional; type *double*
  - analytical\_center\_of\_gravity\_z - optional; type *double*
  - length - optional; type *double*
  - volume - optional; type *double*
  - mass - optional; type *double*
  - center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_of\_gravity\_x - optional; type *double*
  - center\_of\_gravity\_y - optional; type *double*
  - center\_of\_gravity\_z - optional; type *double*
  - member\_representative - optional; type *int*
  - design\_properties\_via\_member - optional; type *boolean*
  - design\_properties\_via\_parent\_member\_set - optional; type *boolean*
  - design\_properties\_parent\_member\_set - optional; type *int*
  - comment - optional; type *string*
  - member\_type\_rib\_alignment - optional; type *member\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
  - member\_rib\_first\_surface - optional; type *int*
  - member\_rib\_second\_surface - optional; type *int*
  - member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
  - align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
  - flange\_dimensions - optional; type *array\_of\_member\_flange\_dimensions*
    - member\_flange\_dimensions - optional, unbounded; type *member\_flange\_dimensions*
      - no - optional; type *int*
      - end\_ordinate - optional; type *double*
      - length - optional; type *double*
      - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH',

'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED'  
}

- reference\_length - optional; type *double*
- reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
- width\_minus\_y\_integrative - optional; type *double*
- width\_minus\_y\_effective - optional; type *double*
- width\_minus\_y\_maximal - optional; type *double*
- width\_plus\_y\_integrative - optional; type *double*
- width\_plus\_y\_effective - optional; type *double*
- width\_plus\_y\_maximal - optional; type *double*
- distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- synchronize\_width\_mode - optional; type *boolean*
- relative\_ordinates\_mode - optional; type *boolean*
- member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_type\_definable\_stiffness - optional; type *int*
- result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- result\_beam\_y\_z - optional; type *double*
- result\_beam\_y\_plus - optional; type *double*
- result\_beam\_z\_plus - optional; type *double*
- result\_beam\_y\_minus - optional; type *double*
- result\_beam\_z\_minus - optional; type *double*
- result\_beam\_radius - optional; type *double*
- result\_beam\_include\_surfaces - optional; type *array\_of\_int*
- result\_beam\_include\_all\_surfaces - optional; type *boolean*
- result\_beam\_include\_solids - optional; type *array\_of\_int*
- result\_beam\_include\_all\_solids - optional; type *boolean*
- result\_beam\_include\_members - optional; type *array\_of\_int*
- result\_beam\_include\_all\_members - optional; type *boolean*
- result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- result\_beam\_exclude\_members - optional; type *array\_of\_int*
- projected\_length - optional; type *double*
- section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_start\_absolute - optional; type *double*
- section\_distance\_from\_start\_relative - optional; type *double*
- section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_end\_absolute - optional; type *double*
- section\_distance\_from\_end\_relative - optional; type *double*
- section\_alignment - optional; type *member\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- section\_start - optional; type *int*
- section\_end - optional; type *int*
- section\_internal - optional; type *int*
- section\_material - optional; type *int*
- curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- curved\_member\_is\_cantilevers - optional; type *boolean*
- curved\_member\_cantilevers\_type - optional; type *member\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- rotation\_specification\_type - optional; type *member\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- rotation\_angle - optional; type *double*
- rotation\_help\_node - optional; type *int*
- rotation\_plane\_type - optional; type *member\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- rotation\_surface - optional; type *int*
- rotation\_surface\_plane\_type - optional; type *member\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- is\_rotated - optional; type *boolean*
- member\_hinge\_start - optional; type *int*
- member\_hinge\_end - optional; type *int*
- member\_eccentricity\_start - optional; type *int*
- member\_eccentricity\_end - optional; type *int*
- support - optional; type *int*
- member\_transverse\_stiffener - optional; type *int*
- member\_nonlinearity - optional; type *int*
- member\_result\_intermediate\_point - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- aluminum\_effective\_lengths - optional; type *int*
- aluminum\_boundary\_conditions - optional; type *int*
- aluminum\_member\_local\_section\_reduction - optional; type *int*
- aluminum\_member\_transverse\_weld - optional; type *int*
- aluminum\_member\_shear\_panel - optional; type *int*
- aluminum\_member\_rotational\_restraint - optional; type *int*
- member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_aluminum\_design\_sls\_configuration - optional; type *int*
- concrete\_effective\_lengths - optional; type *int*
- concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- concrete\_cover - optional; type *double*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_left - optional; type *double*
- concrete\_cover\_right - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- concrete\_cover\_min - optional; type *array\_of\_int*
- concrete\_cover\_min\_top - optional; type *array\_of\_int*
- concrete\_cover\_min\_left - optional; type *array\_of\_int*
- concrete\_cover\_min\_right - optional; type *array\_of\_int*
- concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- concrete\_durability - optional; type *int*



- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_left - optional; type *int*
- concrete\_durability\_right - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_concrete\_shear\_reinforcement\_spans*
  - member\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA' }
    - stirrup\_start\_offset - optional; type *double*
    - stirrup\_end\_offset - optional; type *double*
    - one\_stirrup\_length - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_length - optional; type *string*
    - length - optional; type *double*
    - one\_stirrup\_weight - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
    - weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_concrete\_longitudinal\_reinforcement\_items*
  - member\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction

- `bar_size_designation_corner_type` - optional; type `bar_size_designation_corner_type` - type `undefined` with restriction
- `corner_reinforcement_enabled` - optional; type `boolean`
- `reinforcement_area_symmetrical` - optional; type `double`
- `reinforcement_area_unsymmetrical_at_side` - optional; type `double`
- `reinforcement_area_unsymmetrical_top_side` - optional; type `double`
- `reinforcement_area_unsymmetrical_bottom_side` - optional; type `double`
- `reinforcement_area_uniformly_surrounding` - optional; type `double`
- `reinforcement_area_line` - optional; type `double`
- `reinforcement_area_single` - optional; type `double`
- `reinforcement_area_corner` - optional; type `double`
- `reinforcement_area_total` - optional; type `double`
- `span_position_reference_internal_node` - optional; type `int`
- `span_position_reference_x_location_relative` - optional; type `double`
- `span_position_reference_x_location_absolute` - optional; type `double`
- `span_start_relative` - optional; type `double`
- `span_start_absolute` - optional; type `double`
- `span_end_relative` - optional; type `double`
- `span_end_absolute` - optional; type `double`
- `span_length` - optional; type `double`
- `additional_offset_type` - optional; type `additional_offset_type` - type `undefined` with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_type_single_line_type` - optional; type `additional_offset_type_single_line_type` - type `undefined` with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_top_side` - optional; type `double`
- `additional_offset_bottom_side` - optional; type `double`
- `additional_offset_left_side` - optional; type `double`
- `additional_offset_right_side` - optional; type `double`
- `additional_offset_reference_type` - optional; type `additional_offset_reference_type` - type `undefined` with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_start_type` - optional; type `additional_offset_reference_type_at_start_type` - type `undefined` with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_end_type` - optional; type `additional_offset_reference_type_at_end_type` - type `undefined` with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_horizontal_offset` - optional; type `double`
- `additional_horizontal_offset_at_start` - optional; type `double`
- `additional_horizontal_offset_at_end` - optional; type `double`
- `additional_vertical_offset` - optional; type `double`
- `additional_vertical_offset_at_start` - optional; type `double`
- `additional_vertical_offset_at_end` - optional; type `double`
- `anchorage_start_anchor_type` - optional; type `anchorage_start_anchor_type` - type `undefined` with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_start_anchor_length` - optional; type `double`
- `anchorage_start_bending_diameter` - optional; type `double`
- `anchorage_end_anchor_type` - optional; type `anchorage_end_anchor_type` - type `undefined` with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_end_anchor_length` - optional; type `double`
- `anchorage_end_bending_diameter` - optional; type `double`
- `one_rebar_length` - optional; type `double`
- `one_rebar_minimal_and_maximal_length` - optional; type `string`
- `one_rebar_unsymmetrical_at_side_length` - optional; type `double`
- `one_rebar_unsymmetrical_at_side_minimal_and_maximal_length` - optional; type `string`
- `one_rebar_unsymmetrical_top_side_length` - optional; type `double`
- `one_rebar_unsymmetrical_top_side_minimal_and_maximal_length` - optional; type `string`
- `one_rebar_unsymmetrical_bottom_side_length` - optional; type `double`
- `one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_length` - optional; type `string`
- `one_rebar_corner_length` - optional; type `double`
- `one_rebar_corner_minimal_and_maximal_length` - optional; type `string`

- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- steel\_effective\_lengths - optional; type *int*
- steel\_boundary\_conditions - optional; type *int*
- steel\_member\_local\_section\_reduction - optional; type *int*
- steel\_member\_transverse\_weld - optional; type *int*
- steel\_member\_shear\_panel - optional; type *int*
- steel\_member\_rotational\_restraint - optional; type *int*
- member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_steel\_design\_fr\_configuration - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- timber\_member\_local\_section\_reduction - optional; type *int*
- timber\_member\_shear\_panel - optional; type *int*
- timber\_member\_rotational\_restraint - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- end\_modifications\_member\_start\_extension - optional; type *double*
- end\_modifications\_member\_start\_slope\_y - optional; type *double*
- end\_modifications\_member\_start\_slope\_z - optional; type *double*
- end\_modifications\_member\_end\_extension - optional; type *double*
- end\_modifications\_member\_end\_slope\_y - optional; type *double*
- end\_modifications\_member\_end\_slope\_z - optional; type *double*
- has\_any\_end\_modifications - optional; type *boolean*
- deflection\_check\_direction - optional; type *member\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_start - optional; type *int*
- design\_support\_on\_member\_end - optional; type *int*
- design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_design\_supports\_on\_internal\_nodes*
  - member\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_y\_axis - optional; type *array\_of\_member\_deflection\_segments\_y\_axis*
  - member\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type *array\_of\_member\_deflection\_segments\_z\_axis*
  - member\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 58. get\_member\_definable\_stiffness

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_definable\\_stiffness](http://localhost:8082/get_member_definable_stiffness)

**Input:** *get\_member\_definable\_stiffness\_request* (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_definable\_stiffness*

- no type *int*

**Output:** *get\_member\_definable\_stiffness\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_definable\_stiffnessResponse*

- value type *member\_definable\_stiffness*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to - optional; type *array\_of\_int*

- torsional\_stiffness - optional; type *double*
- bending\_stiffness\_y - optional; type *double*
- bending\_stiffness\_z - optional; type *double*
- axial\_stiffness - optional; type *double*
- shear\_stiffness\_y - optional; type *double*
- shear\_stiffness\_z - optional; type *double*
- specific\_weight - optional; type *double*
- section\_area - optional; type *double*
- rotation - optional; type *double*
- thermal\_expansion\_alpha - optional; type *double*
- thermal\_expansion\_width - optional; type *double*
- thermal\_expansion\_height - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 59. get\_member\_eccentricity

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_eccentricity](http://localhost:8082/get_member_eccentricity)

**Input:** [get\\_member\\_eccentricity\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_eccentricity](#)

- no type *int*

**Output:** [get\\_member\\_eccentricity\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_eccentricityResponse](#)

- value type [member\\_eccentricity](#)
  - no type *int*
  - axial\_offset\_active - optional; type *boolean*
  - comment - optional; type *string*
  - coordinate\_system - optional; type *string*
  - generating\_object\_info - optional; type *string*
  - hinge\_location\_at\_node - optional; type *boolean*
  - horizontal\_section\_alignment - optional; type [member\\_eccentricity\\_horizontal\\_section\\_alignment](#) - type *undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_RIGHT' }
  - is\_generated - optional; type *boolean*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - offset - optional; type [vector\\_3d](#)
    - x type *double*
    - y type *double*
    - z type *double*
  - offset\_x - optional; type *double*
  - offset\_y - optional; type *double*
  - offset\_z - optional; type *double*
  - specification\_type - optional; type [member\\_eccentricity\\_specification\\_type](#) - type *undefined* with restriction - enum { 'TYPE\_ABSOLUTE', 'TYPE\_RELATIVE', 'TYPE\_RELATIVE\_AND\_ABSOLUTE' }
  - transverse\_offset\_reference\_type - optional; type [member\\_eccentricity\\_transverse\\_offset\\_reference\\_type](#) - type *undefined* with restriction - enum { 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_MEMBER\_SECTION', 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_SURFACE\_THICKNESS', 'TRANSVERSE\_OFFSET\_TYPE\_NONE' }
  - transverse\_offset\_horizontal\_alignment - optional; type [member\\_eccentricity\\_transverse\\_offset\\_horizontal\\_alignment](#) - type *undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_RIGHT' }
  - transverse\_offset\_member\_reference\_node - optional; type *int*
  - transverse\_offset\_reference\_member - optional; type *int*
  - transverse\_offset\_reference\_surface - optional; type *int*
  - transverse\_offset\_surface\_reference\_node - optional; type *int*
  - transverse\_offset\_vertical\_alignment - optional; type [member\\_eccentricity\\_transverse\\_offset\\_vertical\\_alignment](#) - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
  - vertical\_section\_alignment - optional; type [member\\_eccentricity\\_vertical\\_section\\_alignment](#) - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 60. get\_member\_hinge

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_hinge](http://localhost:8082/get_member_hinge)

**Input:** [get\\_member\\_hinge\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_hinge](#)

- no type *int*

**Output:** [get\\_member\\_hinge\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_hingeResponse](#)

- value type [member\\_hinge](#)
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - coordinate\_system - optional; type *string*
  - axial\_release\_n - optional; type *double*
  - axial\_release\_n\_nonlinearity - optional; type [member\\_hinge\\_axial\\_release\\_n\\_nonlinearity](#) - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2' }

'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }

- axial\_release\_vy - optional; type *double*
- axial\_release\_vy\_nonlinearity - optional; type *member\_hinge\_axial\_release\_vy\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- axial\_release\_vz - optional; type *double*
- axial\_release\_vz\_nonlinearity - optional; type *member\_hinge\_axial\_release\_vz\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- comment - optional; type *string*
- diagram\_along\_x\_end - optional; type *member\_hinge\_diagram\_along\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_x\_is\_sorted - optional; type *boolean*
- diagram\_along\_x\_start - optional; type *member\_hinge\_diagram\_along\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_x\_symmetric - optional; type *boolean*
- diagram\_along\_x\_table - optional; type *array\_of\_member\_hinge\_diagram\_along\_x\_table*
  - member\_hinge\_diagram\_along\_x\_table - optional, unbounded; type *member\_hinge\_diagram\_along\_x\_table*
    - no - optional; type *int*
    - displacement - optional; type *double*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_along\_y\_end - optional; type *member\_hinge\_diagram\_along\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_y\_is\_sorted - optional; type *boolean*
- diagram\_along\_y\_start - optional; type *member\_hinge\_diagram\_along\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_y\_symmetric - optional; type *boolean*
- diagram\_along\_y\_table - optional; type *array\_of\_member\_hinge\_diagram\_along\_y\_table*
  - member\_hinge\_diagram\_along\_y\_table - optional, unbounded; type *member\_hinge\_diagram\_along\_y\_table*
    - no - optional; type *int*
    - displacement - optional; type *double*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_along\_z\_end - optional; type *member\_hinge\_diagram\_along\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_z\_is\_sorted - optional; type *boolean*
- diagram\_along\_z\_start - optional; type *member\_hinge\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_z\_symmetric - optional; type *boolean*
- diagram\_along\_z\_table - optional; type *array\_of\_member\_hinge\_diagram\_along\_z\_table*
  - member\_hinge\_diagram\_along\_z\_table - optional, unbounded; type *member\_hinge\_diagram\_along\_z\_table*
    - no - optional; type *int*
    - displacement - optional; type *double*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_x\_end - optional; type *member\_hinge\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_is\_sorted - optional; type *boolean*
- diagram\_around\_x\_start - optional; type *member\_hinge\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_symmetric - optional; type *boolean*
- diagram\_around\_x\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_x\_table*
  - member\_hinge\_diagram\_around\_x\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_x\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_y\_end - optional; type *member\_hinge\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_is\_sorted - optional; type *boolean*
- diagram\_around\_y\_start - optional; type *member\_hinge\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_symmetric - optional; type *boolean*
- diagram\_around\_y\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_y\_table*
  - member\_hinge\_diagram\_around\_y\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_y\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*

- moment - optional; type *double*
- spring - optional; type *double*
- note - optional; type *string*
- diagram\_around\_z\_end - optional; type *member\_hinge\_diagram\_around\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_is\_sorted - optional; type *boolean*
- diagram\_around\_z\_start - optional; type *member\_hinge\_diagram\_around\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_symmetric - optional; type *boolean*
- diagram\_around\_z\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_z\_table*
  - member\_hinge\_diagram\_around\_z\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_z\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- friction\_coefficient\_x - optional; type *double*
- friction\_coefficient\_xy - optional; type *double*
- friction\_coefficient\_xz - optional; type *double*
- friction\_coefficient\_y - optional; type *double*
- friction\_coefficient\_yx - optional; type *double*
- friction\_coefficient\_yz - optional; type *double*
- friction\_coefficient\_z - optional; type *double*
- friction\_coefficient\_zx - optional; type *double*
- friction\_coefficient\_zy - optional; type *double*
- friction\_direction\_independent\_x - optional; type *boolean*
- friction\_direction\_independent\_y - optional; type *boolean*
- friction\_direction\_independent\_z - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- members - optional; type *string*
- moment\_release\_mt - optional; type *double*
- moment\_release\_mt\_nonlinearity - optional; type *member\_hinge\_moment\_release\_mt\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- moment\_release\_my - optional; type *double*
- moment\_release\_my\_nonlinearity - optional; type *member\_hinge\_moment\_release\_my\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- moment\_release\_mz - optional; type *double*
- moment\_release\_mz\_nonlinearity - optional; type *member\_hinge\_moment\_release\_mz\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- special\_type\_of\_hinge\_direction\_along\_x - optional; type *boolean*
- special\_type\_of\_hinge\_direction\_along\_y - optional; type *boolean*
- special\_type\_of\_hinge\_direction\_along\_z - optional; type *boolean*
- special\_type\_of\_hinge\_direction\_around\_x - optional; type *boolean*
- special\_type\_of\_hinge\_direction\_around\_y - optional; type *boolean*
- special\_type\_of\_hinge\_direction\_around\_z - optional; type *boolean*
- special\_type\_of\_hinge\_enabled - optional; type *boolean*
- partial\_activity\_along\_x\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_x\_negative\_force - optional; type *double*
- partial\_activity\_along\_x\_negative\_slippage - optional; type *double*
- partial\_activity\_along\_x\_negative\_type - optional; type *member\_hinge\_partial\_activity\_along\_x\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_x\_positive\_displacement - optional; type *double*
- partial\_activity\_along\_x\_positive\_force - optional; type *double*
- partial\_activity\_along\_x\_positive\_slippage - optional; type *double*
- partial\_activity\_along\_x\_positive\_type - optional; type *member\_hinge\_partial\_activity\_along\_x\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_y\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_y\_negative\_force - optional; type *double*
- partial\_activity\_along\_y\_negative\_slippage - optional; type *double*
- partial\_activity\_along\_y\_negative\_type - optional; type *member\_hinge\_partial\_activity\_along\_y\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_y\_positive\_displacement - optional; type *double*
- partial\_activity\_along\_y\_positive\_force - optional; type *double*
- partial\_activity\_along\_y\_positive\_slippage - optional; type *double*
- partial\_activity\_along\_y\_positive\_type - optional; type *member\_hinge\_partial\_activity\_along\_y\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_z\_negative\_displacement - optional; type *double*

- `partial_activity_along_z_negative_force` - optional; type *double*
- `partial_activity_along_z_negative_slippage` - optional; type *double*
- `partial_activity_along_z_negative_type` - optional; type *member\_hinge\_partial\_activity\_along\_z\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_along_z_positive_displacement` - optional; type *double*
- `partial_activity_along_z_positive_force` - optional; type *double*
- `partial_activity_along_z_positive_slippage` - optional; type *double*
- `partial_activity_along_z_positive_type` - optional; type *member\_hinge\_partial\_activity\_along\_z\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_x_negative_moment` - optional; type *double*
- `partial_activity_around_x_negative_rotation` - optional; type *double*
- `partial_activity_around_x_negative_slippage` - optional; type *double*
- `partial_activity_around_x_negative_type` - optional; type *member\_hinge\_partial\_activity\_around\_x\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_x_positive_moment` - optional; type *double*
- `partial_activity_around_x_positive_rotation` - optional; type *double*
- `partial_activity_around_x_positive_slippage` - optional; type *double*
- `partial_activity_around_x_positive_type` - optional; type *member\_hinge\_partial\_activity\_around\_x\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_y_negative_moment` - optional; type *double*
- `partial_activity_around_y_negative_rotation` - optional; type *double*
- `partial_activity_around_y_negative_slippage` - optional; type *double*
- `partial_activity_around_y_negative_type` - optional; type *member\_hinge\_partial\_activity\_around\_y\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_y_positive_moment` - optional; type *double*
- `partial_activity_around_y_positive_rotation` - optional; type *double*
- `partial_activity_around_y_positive_slippage` - optional; type *double*
- `partial_activity_around_y_positive_type` - optional; type *member\_hinge\_partial\_activity\_around\_y\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_z_negative_moment` - optional; type *double*
- `partial_activity_around_z_negative_rotation` - optional; type *double*
- `partial_activity_around_z_negative_slippage` - optional; type *double*
- `partial_activity_around_z_negative_type` - optional; type *member\_hinge\_partial\_activity\_around\_z\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `partial_activity_around_z_positive_moment` - optional; type *double*
- `partial_activity_around_z_positive_rotation` - optional; type *double*
- `partial_activity_around_z_positive_slippage` - optional; type *double*
- `partial_activity_around_z_positive_type` - optional; type *member\_hinge\_partial\_activity\_around\_z\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- `id_for_export_import` - optional; type *string*
- `metadata_for_export_import` - optional; type *string*

## 61. `get_member_imperfection`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_member_imperfection`

**Input:** `get_member_imperfection_request` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_member\_imperfection*

- no type *int*
- `imperfection_case_no` type *int*

**Output:** `get_member_imperfection_response` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_member\_imperfectionResponse*

- value type *member\_imperfection*
  - no type *int*
  - `imperfection_type` - optional; type *member\_imperfection\_imperfection\_type* - type *undefined* with restriction - enum { 'IMPERFECTION\_TYPE\_INITIAL\_BOW', 'IMPERFECTION\_TYPE\_INITIAL\_BOW\_AND\_CRITERION', 'IMPERFECTION\_TYPE\_INITIAL\_SWAY' }
  - `members` - optional; type *array\_of\_int*
  - `imperfection_case` - optional; type *int*
  - `definition_type` - optional; type *member\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_ABSOLUTE', 'DEFINITION\_TYPE\_ANSI\_CURRENT', 'DEFINITION\_TYPE\_ANSI\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_CSA\_CURRENT', 'DEFINITION\_TYPE\_CSA\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_EN\_1992\_1\_1993\_1', 'DEFINITION\_TYPE\_EN\_1993\_1\_1', 'DEFINITION\_TYPE\_EN\_1995\_1\_1', 'DEFINITION\_TYPE\_EN\_1999\_1\_1', 'DEFINITION\_TYPE\_GB\_50017\_2017', 'DEFINITION\_TYPE\_GB\_50017\_2017\_CURRENT', 'DEFINITION\_TYPE\_GB\_50017\_2017\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_NOTIONAL\_LOAD', 'DEFINITION\_TYPE\_RELATIVE' }
  - `coordinate_system` - optional; type *string*
  - `imperfection_direction` - optional; type *member\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE' }

'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V\_NEGATIVE' }

- basic\_value\_absolute - optional; type *double*
- basic\_value\_coefficient - optional; type *double*
- basic\_value\_relative - optional; type *double*
- basic\_value\_force - optional; type *double*
- section\_design - optional; type *member\_imperfection\_section\_design* - type *undefined* with restriction - enum { 'SECTION\_DESIGN\_ELASTIC', 'SECTION\_DESIGN\_PLASTIC' }
- active\_criterion - optional; type *member\_imperfection\_active\_criterion* - type *undefined* with restriction - enum { 'ACTIVITY\_CRITERION\_ALWAYS', 'ACTIVITY\_CRITERION\_DEFINE', 'ACTIVITY\_CRITERION\_DIN\_18800', 'ACTIVITY\_CRITERION\_EN\_1993', 'ACTIVITY\_CRITERION\_EN\_1999' }
- active\_bow - optional; type *double*
- standard\_factor\_enumeration - optional; type *member\_imperfection\_standard\_factor\_enumeration* - type *undefined* with restriction - enum { 'STANDARD\_FACTOR\_ASD', 'STANDARD\_FACTOR\_LRFD' }
- standard\_factor\_number - optional; type *double*
- height - optional; type *double*
- column\_in\_row - optional; type *int*
- number\_of\_floors - optional; type *int*
- case\_object - optional; type *int*
- reduction\_factor\_h - optional; type *double*
- reduction\_factor\_m - optional; type *double*
- initial\_sway - optional; type *double*
- initial\_sway\_inverted - optional; type *double*
- delta - optional; type *double*
- parameters - optional; type *array\_of\_int*
- reference\_to\_list\_of\_members - optional; type *boolean*
- refer\_distance\_from\_objects\_to\_assign - optional; type *boolean*
- imperfection\_over\_total\_length\_of\_objects\_to\_assign - optional; type *boolean*
- distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_a\_relative - optional; type *double*
- distance\_b\_relative - optional; type *double*
- distance\_a\_absolute - optional; type *double*
- distance\_b\_absolute - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 62. get\_member\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_member\_load

**Input:** get\_member\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_load*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_member\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_loadResponse*

- value type *member\_load*
  - no type *int*
  - load\_type - optional; type *member\_load\_load\_type* - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_AXIAL\_DISPLACEMENT', 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_DISPLACEMENT', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_INITIAL\_PRESTRESS', 'LOAD\_TYPE\_MOMENT', 'LOAD\_TYPE\_PIPE\_CONTENT\_FULL', 'LOAD\_TYPE\_PIPE\_CONTENT\_PARTIAL', 'LOAD\_TYPE\_PIPE\_INTERNAL\_PRESSURE', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY MOTION', 'LOAD\_TYPE\_ROTATION', 'LOAD\_TYPE\_TEMPERATURE', 'LOAD\_TYPE\_TEMPERATURE\_CHANGE' }
  - members - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *string*
  - load\_distribution - optional; type *member\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }
  - load\_direction - optional; type *member\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_PRINCIPAL\_U', 'LOAD\_DIRECTION\_PRINCIPAL\_V' }
  - load\_direction\_orientation - optional; type *member\_load\_load\_direction\_orientation* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
  - form\_finding\_definition\_type - optional; type *member\_load\_form\_finding\_definition\_type* - type *undefined* with restriction - enum { 'FORM\_FINDING\_TYPE\_FORCE', 'FORM\_FINDING\_TYPE\_GEOMETRIC' }
  - magnitude - optional; type *double*
  - magnitude\_1 - optional; type *double*
  - magnitude\_2 - optional; type *double*
  - magnitude\_3 - optional; type *double*
  - magnitude\_t\_c - optional; type *double*
  - magnitude\_t\_c\_1 - optional; type *double*
  - magnitude\_t\_c\_2 - optional; type *double*
  - magnitude\_t\_c\_3 - optional; type *double*
  - magnitude\_delta\_t - optional; type *double*
  - magnitude\_delta\_t\_1 - optional; type *double*
  - magnitude\_delta\_t\_2 - optional; type *double*



- magnitude\_delta\_t\_3 - optional; type *double*
- magnitude\_t\_t - optional; type *double*
- magnitude\_t\_t\_1 - optional; type *double*
- magnitude\_t\_t\_2 - optional; type *double*
- magnitude\_t\_t\_3 - optional; type *double*
- magnitude\_t\_b - optional; type *double*
- magnitude\_t\_b\_1 - optional; type *double*
- magnitude\_t\_b\_2 - optional; type *double*
- magnitude\_t\_b\_3 - optional; type *double*
- individual\_mass\_components - optional; type *boolean*
- mass\_global - optional; type *double*
- mass\_x - optional; type *double*
- mass\_y - optional; type *double*
- mass\_z - optional; type *double*
- distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_a\_absolute - optional; type *double*
- distance\_a\_relative - optional; type *double*
- distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_b\_absolute - optional; type *double*
- distance\_b\_relative - optional; type *double*
- distance\_c\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_c\_absolute - optional; type *double*
- distance\_c\_relative - optional; type *double*
- count\_n - optional; type *int*
- varying\_load\_parameters\_are\_defined\_as\_relative - optional; type *boolean*
- varying\_load\_parameters - optional; type *array\_of\_member\_load\_varying\_load\_parameters*
  - member\_load\_varying\_load\_parameters - optional, unbounded; type *member\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
    - magnitude\_t\_c - optional; type *double*
    - magnitude\_delta\_t - optional; type *double*
    - magnitude\_t\_t - optional; type *double*
    - magnitude\_t\_b - optional; type *double*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- angular\_velocity - optional; type *double*
- angular\_acceleration - optional; type *double*
- axis\_definition\_type - optional; type *member\_load\_axis\_definition\_type* - type *undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *member\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *member\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- filling\_height - optional; type *double*
- reference\_to\_list\_of\_members - optional; type *boolean*
- distance\_from\_member\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- has\_force\_eccentricity - optional; type *boolean*
- eccentricity\_horizontal\_alignment - optional; type *member\_load\_eccentricity\_horizontal\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_RIGHT' }
- eccentricity\_vertical\_alignment - optional; type *member\_load\_eccentricity\_vertical\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_TOP' }
- eccentricity\_section\_middle - optional; type *member\_load\_eccentricity\_section\_middle* - type *undefined* with restriction - enum { 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_CENTER\_OF\_GRAVITY', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_NONE', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_SHEAR\_CENTER' }
- is\_eccentricity\_at\_end\_different\_from\_start - optional; type *boolean*
- eccentricity\_y\_at\_start - optional; type *double*
- eccentricity\_z\_at\_start - optional; type *double*
- eccentricity\_y\_at\_end - optional; type *double*
- eccentricity\_z\_at\_end - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- form\_finding\_internal\_force - optional; type *member\_load\_form\_finding\_internal\_force* - type *undefined* with restriction - enum { 'FORM\_FINDING\_INTERNAL\_FORCE\_COMPRESSION', 'FORM\_FINDING\_INTERNAL\_FORCE\_TENSION' }
- form\_finding\_geometry\_definition - optional; type *member\_load\_form\_finding\_geometry\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LENGTH', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LOW\_POINT\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_MAXIMUM\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_UNSTRESSED\_LENGTH' }
- form\_finding\_force\_definition - optional; type *member\_load\_form\_finding\_force\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_AVERAGE', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_DENSITY', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_HORIZONTAL\_TENSION\_COMPONENT', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MAXIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_J\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMUM\_FORCE\_IN\_MEMBER' }

- 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_I\_END';
- 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_J\_END' }
- form\_finding\_magnitude\_is\_defined\_as\_relative - optional; type *boolean*
- form\_finding\_magnitude\_absolute - optional; type *double*
- form\_finding\_magnitude\_relative - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 63. [get\\_member\\_nonlinearity](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_nonlinearity](http://localhost:8082/get_member_nonlinearity)

**Input:** [get\\_member\\_nonlinearity\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_nonlinearity](#)

- no type *int*

**Output:** [get\\_member\\_nonlinearity\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_nonlinearityResponse](#)

- value type [member\\_nonlinearity](#)
  - no type *int*
  - type - optional; type [member\\_nonlinearity\\_type](#) - type *undefined* with restriction - enum { 'TYPE\_FAILURE\_IF\_COMPRESSION', 'TYPE\_FAILURE\_IF\_COMPRESSION\_WITH\_SLIPPAGE', 'TYPE\_FAILURE\_IF\_TENSION', 'TYPE\_FAILURE\_IF\_TENSION\_WITH\_SLIPPAGE', 'TYPE\_SLIPPAGE', 'TYPE\_TEARING', 'TYPE\_TEARING\_IF\_COMPRESSION', 'TYPE\_TEARING\_IF\_TENSION', 'TYPE\_YIELDING', 'TYPE\_YIELDING\_IF\_COMPRESSION', 'TYPE\_YIELDING\_IF\_TENSION' }
  - assigned\_to - optional; type [array\\_of\\_int](#)
  - comment - optional; type *string*
  - compression\_force - optional; type *double*
  - generating\_object\_info - optional; type *string*
  - is\_generated - optional; type *boolean*
  - name - optional; type *string*
  - slippage - optional; type *double*
  - tension\_force - optional; type *double*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 64. [get\\_member\\_representative](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_representative](http://localhost:8082/get_member_representative)

**Input:** [get\\_member\\_representative\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_representative](#)

- no type *int*

**Output:** [get\\_member\\_representative\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_representativeResponse](#)

- value type [member\\_representative](#)
  - no type *int*
  - type - optional; type [member\\_representative\\_type](#) - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE', 'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - comment - optional; type *string*
  - number\_of\_members - optional; type *int*
  - total\_length - optional; type *double*
  - total\_volume - optional; type *double*
  - total\_mass - optional; type *double*
  - total\_surface\_of\_coating - optional; type *double*
  - nodes\_on\_member\_from\_start - optional; type [array\\_of\\_int](#)
  - is\_deactivated\_for\_calculation - optional; type *boolean*
  - line - optional; type *int*
  - section\_distribution\_type - optional; type [member\\_representative\\_section\\_distribution\\_type](#) - type *undefined* with restriction - enum { 'SECTION\_DISTRIBUTION\_TYPE\_LINEAR', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
  - reference\_type - optional; type [member\\_representative\\_reference\\_type](#) - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - nodes - optional; type [array\\_of\\_int](#)
  - node\_start - optional; type *int*
  - node\_end - optional; type *int*
  - analytical\_length - optional; type *double*
  - analytical\_volume - optional; type *double*
  - analytical\_mass - optional; type *double*
  - analytical\_surface\_of\_coating - optional; type *double*
  - analytical\_center\_of\_gravity - optional; type [vector\\_3d](#)
    - x type *double*
    - y type *double*
    - z type *double*
  - analytical\_center\_of\_gravity\_x - optional; type *double*
  - analytical\_center\_of\_gravity\_y - optional; type *double*

- analytical\_center\_of\_gravity\_z - optional; type *double*
- length - optional; type *double*
- volume - optional; type *double*
- mass - optional; type *double*
- surface\_of\_coating - optional; type *double*
- center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- center\_of\_gravity\_x - optional; type *double*
- center\_of\_gravity\_y - optional; type *double*
- center\_of\_gravity\_z - optional; type *double*
- member\_representative - optional; type *int*
- design\_properties\_via\_member - optional; type *boolean*
- design\_properties\_via\_parent\_member\_set - optional; type *boolean*
- design\_properties\_parent\_member\_set - optional; type *int*
- member\_type\_rib\_alignment - optional; type *member\_representative\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
- member\_rib\_first\_surface - optional; type *int*
- member\_rib\_second\_surface - optional; type *int*
- member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
- member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
- result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_representative\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- flange\_dimensions - optional; type *array\_of\_member\_representative\_flange\_dimensions*
  - member\_representative\_flange\_dimensions - optional, unbounded; type *member\_representative\_flange\_dimensions*
    - no - optional; type *int*
    - end\_ordinate - optional; type *double*
    - length - optional; type *double*
    - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED' }
    - reference\_length - optional; type *double*
    - reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
    - width\_minus\_y\_integrative - optional; type *double*
    - width\_minus\_y\_effictive - optional; type *double*
    - width\_minus\_y\_maximal - optional; type *double*
    - width\_plus\_y\_integrative - optional; type *double*
    - width\_plus\_y\_effictive - optional; type *double*
    - width\_plus\_y\_maximal - optional; type *double*
    - distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- synchronize\_width\_mode - optional; type *boolean*
- relative\_ordinates\_mode - optional; type *boolean*
- member\_type\_definable\_stiffness - optional; type *int*
- result\_beam\_y\_z - optional; type *double*
- result\_beam\_y\_plus - optional; type *double*
- result\_beam\_z\_minus - optional; type *double*
- result\_beam\_y\_minus - optional; type *double*
- result\_beam\_z\_minus - optional; type *double*
- result\_beam\_radius - optional; type *double*
- result\_beam\_include\_surfaces - optional; type *array\_of\_int*
- result\_beam\_include\_all\_surfaces - optional; type *boolean*
- result\_beam\_include\_solids - optional; type *array\_of\_int*
- result\_beam\_include\_all\_solids - optional; type *boolean*
- result\_beam\_include\_members - optional; type *array\_of\_int*
- result\_beam\_include\_all\_members - optional; type *boolean*
- result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- result\_beam\_exclude\_members - optional; type *array\_of\_int*
- projected\_length - optional; type *double*
- section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_start\_absolute - optional; type *double*
- section\_distance\_from\_start\_relative - optional; type *double*
- section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_end\_absolute - optional; type *double*
- section\_distance\_from\_end\_relative - optional; type *double*
- section\_alignment - optional; type *member\_representative\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- curved\_member\_is\_cantilevers - optional; type *boolean*
- curved\_member\_cantilevers\_type - optional; type *member\_representative\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- rotation\_specification\_type - optional; type *member\_representative\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- rotation\_angle - optional; type *double*
- rotation\_help\_node - optional; type *int*
- rotation\_plane\_type - optional; type *member\_representative\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- rotation\_surface - optional; type *int*
- rotation\_surface\_plane\_type - optional; type *member\_representative\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- is\_rotated - optional; type *boolean*
- section\_start - optional; type *int*
- section\_end - optional; type *int*
- section\_internal - optional; type *int*

- section\_material - optional; type *int*
- members - optional; type *array\_of\_int*
- member\_hinge\_start - optional; type *int*
- member\_hinge\_end - optional; type *int*
- member\_eccentricity\_start - optional; type *int*
- member\_eccentricity\_end - optional; type *int*
- support - optional; type *int*
- member\_transverse\_stiffener - optional; type *int*
- member\_nonlinearity - optional; type *int*
- member\_result\_intermediate\_point - optional; type *int*
- concrete\_effective\_lengths - optional; type *int*
- steel\_effective\_lengths - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- aluminum\_effective\_lengths - optional; type *int*
- concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_representative\_concrete\_shear\_reinforcement\_spans*
  - member\_representative\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_representative\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA' }
    - stirrup\_start\_offset - optional; type *double*
    - stirrup\_end\_offset - optional; type *double*
    - one\_stirrup\_length - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_length - optional; type *string*
    - length - optional; type *double*
    - one\_stirrup\_weight - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
    - weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_representative\_concrete\_longitudinal\_reinforcement\_items*
  - member\_representative\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_representative\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction

- `bar_size_designation_unsymmetrical_top_side_type` - optional; type `bar_size_designation_unsymmetrical_top_side_type` - type *undefined* with restriction
- `bar_size_designation_unsymmetrical_bottom_side_type` - optional; type `bar_size_designation_unsymmetrical_bottom_side_type` - type *undefined* with restriction
- `bar_size_designation_uniformly_surrounding_type` - optional; type `bar_size_designation_uniformly_surrounding_type` - type *undefined* with restriction
- `bar_size_designation_line_type` - optional; type `bar_size_designation_line_type` - type *undefined* with restriction
- `bar_size_designation_single_type` - optional; type `bar_size_designation_single_type` - type *undefined* with restriction
- `bar_size_designation_corner_type` - optional; type `bar_size_designation_corner_type` - type *undefined* with restriction
- `corner_reinforcement_enabled` - optional; type `boolean`
- `reinforcement_area_symmetrical` - optional; type `double`
- `reinforcement_area_unsymmetrical_at_side` - optional; type `double`
- `reinforcement_area_unsymmetrical_top_side` - optional; type `double`
- `reinforcement_area_unsymmetrical_bottom_side` - optional; type `double`
- `reinforcement_area_uniformly_surrounding` - optional; type `double`
- `reinforcement_area_line` - optional; type `double`
- `reinforcement_area_single` - optional; type `double`
- `reinforcement_area_corner` - optional; type `double`
- `reinforcement_area_total` - optional; type `double`
- `span_position_reference_internal_node` - optional; type `int`
- `span_position_reference_x_location_relative` - optional; type `double`
- `span_position_reference_x_location_absolute` - optional; type `double`
- `span_start_relative` - optional; type `double`
- `span_start_absolute` - optional; type `double`
- `span_end_relative` - optional; type `double`
- `span_end_absolute` - optional; type `double`
- `span_length` - optional; type `double`
- `additional_offset_type` - optional; type `additional_offset_type` - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_type_single_line_type` - optional; type `additional_offset_type_single_line_type` - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_top_side` - optional; type `double`
- `additional_offset_bottom_side` - optional; type `double`
- `additional_offset_left_side` - optional; type `double`
- `additional_offset_right_side` - optional; type `double`
- `additional_offset_reference_type` - optional; type `additional_offset_reference_type` - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_start_type` - optional; type `additional_offset_reference_type_at_start_type` - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_end_type` - optional; type `additional_offset_reference_type_at_end_type` - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_horizontal_offset` - optional; type `double`
- `additional_horizontal_offset_at_start` - optional; type `double`
- `additional_horizontal_offset_at_end` - optional; type `double`
- `additional_vertical_offset` - optional; type `double`
- `additional_vertical_offset_at_start` - optional; type `double`
- `additional_vertical_offset_at_end` - optional; type `double`
- `anchorage_start_anchor_type` - optional; type `anchorage_start_anchor_type` - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_start_anchor_length` - optional; type `double`
- `anchorage_start_bending_diameter` - optional; type `double`
- `anchorage_end_anchor_type` - optional; type `anchorage_end_anchor_type` - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_end_anchor_length` - optional; type `double`
- `anchorage_end_bending_diameter` - optional; type `double`

- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- concrete\_cover - optional; type *double*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- concrete\_cover\_left - optional; type *double*
- concrete\_cover\_right - optional; type *double*
- concrete\_cover\_min - optional; type *array\_of\_int*
- concrete\_cover\_min\_top - optional; type *array\_of\_int*
- concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- concrete\_cover\_min\_left - optional; type *array\_of\_int*
- concrete\_cover\_min\_right - optional; type *array\_of\_int*
- concrete\_durability - optional; type *int*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- concrete\_durability\_left - optional; type *int*
- concrete\_durability\_right - optional; type *int*
- steel\_boundary\_conditions - optional; type *int*
- steel\_member\_local\_section\_reduction - optional; type *int*
- steel\_member\_transverse\_weld - optional; type *int*
- steel\_member\_shear\_panel - optional; type *int*
- steel\_member\_rotational\_restraint - optional; type *int*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- timber\_member\_local\_section\_reduction - optional; type *int*
- timber\_member\_shear\_panel - optional; type *int*
- timber\_member\_rotational\_restraint - optional; type *int*
- aluminum\_boundary\_conditions - optional; type *int*
- aluminum\_member\_local\_section\_reduction - optional; type *int*
- aluminum\_member\_transverse\_weld - optional; type *int*
- aluminum\_member\_shear\_panel - optional; type *int*
- aluminum\_member\_rotational\_restraint - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- end\_modifications\_member\_start\_extension - optional; type *double*
- end\_modifications\_member\_start\_slope\_y - optional; type *double*
- end\_modifications\_member\_start\_slope\_z - optional; type *double*
- end\_modifications\_member\_end\_extension - optional; type *double*
- end\_modifications\_member\_end\_slope\_y - optional; type *double*
- end\_modifications\_member\_end\_slope\_z - optional; type *double*
- has\_any\_end\_modifications - optional; type *boolean*
- deflection\_check\_direction - optional; type *member\_representative\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_representative\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_start - optional; type *int*
- design\_support\_on\_member\_end - optional; type *int*
- design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_representative\_design\_supports\_on\_internal\_nodes*
  - member\_representative\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_representative\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*

- deflection\_segments\_y\_axis - optional; type *array\_of\_member\_representative\_deflection\_segments\_y\_axis*
  - member\_representative\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_representative\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type *array\_of\_member\_representative\_deflection\_segments\_z\_axis*
  - member\_representative\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_representative\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 65. get\_member\_result\_intermediate\_point

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_result\\_intermediate\\_point](http://localhost:8082/get_member_result_intermediate_point)

**Input:** [get\\_member\\_result\\_intermediate\\_point\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_result\\_intermediate\\_point](#)

- no type *int*

**Output:** [get\\_member\\_result\\_intermediate\\_point\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_result\\_intermediate\\_pointResponse](#)

- value type [member\\_result\\_intermediate\\_point](#)
  - no type *int*
  - comment - optional; type *string*
  - distances - optional; type *array\_of\_member\_result\_intermediate\_point\_distances*
    - member\_result\_intermediate\_point\_distances - optional, unbounded; type *member\_result\_intermediate\_point\_distances*
      - no - optional; type *int*
      - value - optional; type *double*
      - note - optional; type *string*
  - distances\_are\_defined\_as\_absolute - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - is\_generated - optional; type *boolean*
  - members - optional; type *array\_of\_int*
  - name - optional; type *string*
  - point\_count - optional; type *int*
  - uniform\_distribution - optional; type *boolean*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 66. get\_member\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_set](http://localhost:8082/get_member_set)

**Input:** [get\\_member\\_set\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_set](#)

- no type *int*

**Output:** [get\\_member\\_set\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_setResponse](#)

- value type [member\\_set](#)
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - set\_type - optional; type *member\_set\_set\_type* - type *undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
  - length - optional; type *double*
  - center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_of\_gravity\_x - optional; type *double*
  - center\_of\_gravity\_y - optional; type *double*
  - center\_of\_gravity\_z - optional; type *double*
  - position - optional; type *string*
  - position\_short - optional; type *string*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - volume - optional; type *double*
  - mass - optional; type *double*
  - member\_set\_representative - optional; type *int*
  - discontinuous\_torsional\_warping - optional; type *boolean*
  - design\_properties\_activated - optional; type *boolean*
  - steel\_effective\_lengths - optional; type *int*
  - steel\_boundary\_conditions - optional; type *int*
  - steel\_member\_local\_section\_reductions - optional; type *int*
  - steel\_member\_shear\_panels - optional; type *int*
  - steel\_member\_rotational\_restraints - optional; type *int*
  - member\_steel\_design\_uls\_configuration - optional; type *int*
  - member\_steel\_design\_sls\_configuration - optional; type *int*
  - member\_steel\_design\_fr\_configuration - optional; type *int*

- member\_steel\_design\_seismic\_configuration - optional; type *int*
- aluminum\_effective\_lengths - optional; type *int*
- aluminum\_boundary\_conditions - optional; type *int*
- aluminum\_member\_local\_section\_reductions - optional; type *int*
- aluminum\_member\_shear\_panels - optional; type *int*
- aluminum\_member\_rotational\_restraints - optional; type *int*
- aluminum\_member\_transverse\_weld - optional; type *int*
- member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_set\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_set\_concrete\_shear\_reinforcement\_spans*
  - member\_set\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_set\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA' }
    - stirrup\_start\_offset - optional; type *double*
    - stirrup\_end\_offset - optional; type *double*
    - one\_stirrup\_length - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_length - optional; type *string*
    - length - optional; type *double*
    - one\_stirrup\_weight - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
    - weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction



- `bar_size_designation_uniformly_surrounding_type` - optional; type `bar_size_designation_uniformly_surrounding_type` - type *undefined* with restriction
- `bar_size_designation_line_type` - optional; type `bar_size_designation_line_type` - type *undefined* with restriction
- `bar_size_designation_single_type` - optional; type `bar_size_designation_single_type` - type *undefined* with restriction
- `bar_size_designation_corner_type` - optional; type `bar_size_designation_corner_type` - type *undefined* with restriction
- `corner_reinforcement_enabled` - optional; type *boolean*
- `reinforcement_area_symmetrical` - optional; type *double*
- `reinforcement_area_unsymmetrical_at_side` - optional; type *double*
- `reinforcement_area_unsymmetrical_top_side` - optional; type *double*
- `reinforcement_area_unsymmetrical_bottom_side` - optional; type *double*
- `reinforcement_area_uniformly_surrounding` - optional; type *double*
- `reinforcement_area_line` - optional; type *double*
- `reinforcement_area_single` - optional; type *double*
- `reinforcement_area_corner` - optional; type *double*
- `reinforcement_area_total` - optional; type *double*
- `span_position_reference_internal_node` - optional; type *int*
- `span_position_reference_x_location_relative` - optional; type *double*
- `span_position_reference_x_location_absolute` - optional; type *double*
- `span_start_relative` - optional; type *double*
- `span_start_absolute` - optional; type *double*
- `span_end_relative` - optional; type *double*
- `span_end_absolute` - optional; type *double*
- `span_length` - optional; type *double*
- `additional_offset_type` - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_type_single_line_type` - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- `additional_offset_top_side` - optional; type *double*
- `additional_offset_bottom_side` - optional; type *double*
- `additional_offset_left_side` - optional; type *double*
- `additional_offset_right_side` - optional; type *double*
- `additional_offset_reference_type` - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_start_type` - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_offset_reference_type_at_end_type` - optional; type *additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- `additional_horizontal_offset` - optional; type *double*
- `additional_horizontal_offset_at_start` - optional; type *double*
- `additional_horizontal_offset_at_end` - optional; type *double*
- `additional_vertical_offset` - optional; type *double*
- `additional_vertical_offset_at_start` - optional; type *double*
- `additional_vertical_offset_at_end` - optional; type *double*
- `anchorage_start_anchor_type` - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK',  
'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE',  
'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR',  
'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_start_anchor_length` - optional; type *double*
- `anchorage_start_bending_diameter` - optional; type *double*
- `anchorage_end_anchor_type` - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK',  
'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE',  
'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR',  
'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- `anchorage_end_anchor_length` - optional; type *double*
- `anchorage_end_bending_diameter` - optional; type *double*
- `one_rebar_length` - optional; type *double*
- `one_rebar_minimal_and_maximal_length` - optional; type *string*
- `one_rebar_unsymmetrical_at_side_length` - optional; type *double*
- `one_rebar_unsymmetrical_at_side_minimal_and_maximal_length` - optional; type *string*

- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- concrete\_cover - optional; type *double*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- concrete\_cover\_left - optional; type *double*
- concrete\_cover\_right - optional; type *double*
- concrete\_cover\_min - optional; type *array\_of\_int*
- concrete\_cover\_min\_top - optional; type *array\_of\_int*
- concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- concrete\_cover\_min\_left - optional; type *array\_of\_int*
- concrete\_cover\_min\_right - optional; type *array\_of\_int*
- concrete\_durability - optional; type *int*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- concrete\_durability\_left - optional; type *int*
- concrete\_durability\_right - optional; type *int*
- concrete\_effective\_lengths - optional; type *int*
- member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- service\_class\_timber\_design - optional; type *int*
- moisture\_class\_timber\_design - optional; type *int*
- service\_conditions\_timber\_design - optional; type *int*
- timber\_member\_shear\_panels - optional; type *int*
- timber\_member\_rotational\_restraints - optional; type *int*
- timber\_local\_section\_reductions - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_transverse\_stiffener - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- deflection\_check\_direction - optional; type *member\_set\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_set\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_set\_start - optional; type *int*
- design\_support\_on\_member\_set\_end - optional; type *int*
- design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_set\_design\_supports\_on\_internal\_nodes*
  - member\_set\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_set\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- members - optional; type *array\_of\_int*
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_y\_axis - optional; type *array\_of\_member\_set\_deflection\_segments\_y\_axis*
  - member\_set\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_set\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type *array\_of\_member\_set\_deflection\_segments\_z\_axis*
  - member\_set\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_set\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 67. get\_member\_set\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_set\\_imperfection](http://localhost:8082/get_member_set_imperfection)

**Input:** `get_member_set_imperfection_request` (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_set\_imperfection*

- no type *int*
- imperfection\_case\_no type *int*

Output: *get\_member\_set\_imperfection\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_set\_imperfectionResponse*

- value type *member\_set\_imperfection*
  - no type *int*
  - imperfection\_type - optional; type *member\_set\_imperfection\_imperfection\_type* - type *undefined* with restriction - enum { 'IMPERFECTION\_TYPE\_INITIAL\_BOW', 'IMPERFECTION\_TYPE\_INITIAL\_BOW\_AND\_CRITERION', 'IMPERFECTION\_TYPE\_INITIAL\_SWAY' }
  - member\_sets - optional; type *array\_of\_int*
  - imperfection\_case - optional; type *int*
  - definition\_type - optional; type *member\_set\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_ABSOLUTE', 'DEFINITION\_TYPE\_ANSI\_CURRENT', 'DEFINITION\_TYPE\_ANSI\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_CSA\_CURRENT', 'DEFINITION\_TYPE\_CSA\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_EN\_1992\_1\_1993\_1', 'DEFINITION\_TYPE\_EN\_1993\_1\_1', 'DEFINITION\_TYPE\_EN\_1995\_1\_1', 'DEFINITION\_TYPE\_EN\_1999\_1\_1', 'DEFINITION\_TYPE\_GB\_50017\_2017', 'DEFINITION\_TYPE\_GB\_50017\_2017\_CURRENT', 'DEFINITION\_TYPE\_GB\_50017\_2017\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_NOTIONAL\_LOAD', 'DEFINITION\_TYPE\_RELATIVE' }
  - coordinate\_system - optional; type *string*
  - imperfection\_direction - optional; type *member\_set\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V\_NEGATIVE' }
  - basic\_value\_absolute - optional; type *double*
  - basic\_value\_relative - optional; type *double*
  - basic\_value\_coefficient - optional; type *double*
  - basic\_value\_force - optional; type *double*
  - section\_design - optional; type *member\_set\_imperfection\_section\_design* - type *undefined* with restriction - enum { 'SECTION\_DESIGN\_ELASTIC', 'SECTION\_DESIGN\_PLASTIC' }
  - active\_criterion - optional; type *member\_set\_imperfection\_active\_criterion* - type *undefined* with restriction - enum { 'ACTIVITY\_CRITERION\_ALWAYS', 'ACTIVITY\_CRITERION\_DEFINE', 'ACTIVITY\_CRITERION\_DIN\_18800', 'ACTIVITY\_CRITERION\_EN\_1993', 'ACTIVITY\_CRITERION\_EN\_1999' }
  - active\_bow - optional; type *double*
  - column\_in\_row - optional; type *int*
  - number\_of\_floors - optional; type *int*
  - standard\_factor\_enumeration - optional; type *member\_set\_imperfection\_standard\_factor\_enumeration* - type *undefined* with restriction - enum { 'STANDARD\_FACTOR\_ASD', 'STANDARD\_FACTOR\_LRFD' }
  - standard\_factor\_number - optional; type *double*
  - height - optional; type *double*
  - case\_object - optional; type *int*
  - reduction\_factor\_h - optional; type *double*
  - reduction\_factor\_m - optional; type *double*
  - initial\_sway - optional; type *double*
  - initial\_sway\_inverted - optional; type *double*
  - delta - optional; type *double*
  - parameters - optional; type *array\_of\_int*
  - refer\_distance\_from\_objects\_to\_assign - optional; type *boolean*
  - imperfection\_over\_total\_length\_of\_objects\_to\_assign - optional; type *boolean*
  - distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_a\_relative - optional; type *double*
  - distance\_b\_relative - optional; type *double*
  - distance\_a\_absolute - optional; type *double*
  - distance\_b\_absolute - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 68. *get\_member\_set\_load*

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: [http://localhost:8082/get\\_member\\_set\\_load](http://localhost:8082/get_member_set_load)

Input: *get\_member\_set\_load\_request* (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_set\_load*

- no type *int*
- load\_case\_no type *int*

Output: *get\_member\_set\_load\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_set\_loadResponse*

- value type *member\_set\_load*
  - no type *int*
  - load\_type - optional; type *member\_set\_load\_load\_type* - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_AXIAL\_DISPLACEMENT', 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_DISPLACEMENT', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_INITIAL\_PRESTRESS', 'LOAD\_TYPE\_MOMENT', 'LOAD\_TYPE\_PIPE\_CONTENT\_FULL', 'LOAD\_TYPE\_PIPE\_CONTENT\_PARTIAL', 'LOAD\_TYPE\_PIPE\_INTERNAL\_PRESSURE', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY MOTION', 'LOAD\_TYPE\_ROTATION', 'LOAD\_TYPE\_TEMPERATURE', 'LOAD\_TYPE\_TEMPERATURE\_CHANGE' }
  - member\_sets - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *string*
  - load\_distribution - optional; type *member\_set\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N' }

'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC',  
'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM',  
'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING',  
'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }

- load\_direction - optional; type *member\_set\_load\_load\_direction* - type *undefined* with restriction - enum {  
'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED',  
'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE',  
'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED',  
'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE',  
'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED',  
'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X',  
'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_PRINCIPAL\_U',  
'LOAD\_DIRECTION\_PRINCIPAL\_V' }
- load\_direction\_orientation - optional; type *member\_set\_load\_load\_direction\_orientation* - type *undefined* with  
restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
- form\_finding\_definition\_type - optional; type *member\_set\_load\_form\_finding\_definition\_type* - type *undefined*  
with restriction - enum { 'FORM\_FINDING\_TYPE\_FORCE', 'FORM\_FINDING\_TYPE\_GEOMETRIC' }
- magnitude - optional; type *double*
- magnitude\_1 - optional; type *double*
- magnitude\_2 - optional; type *double*
- magnitude\_3 - optional; type *double*
- magnitude\_t\_c - optional; type *double*
- magnitude\_t\_c\_1 - optional; type *double*
- magnitude\_t\_c\_2 - optional; type *double*
- magnitude\_t\_c\_3 - optional; type *double*
- magnitude\_delta\_t - optional; type *double*
- magnitude\_delta\_t\_1 - optional; type *double*
- magnitude\_delta\_t\_2 - optional; type *double*
- magnitude\_delta\_t\_3 - optional; type *double*
- magnitude\_t\_t - optional; type *double*
- magnitude\_t\_t\_1 - optional; type *double*
- magnitude\_t\_t\_2 - optional; type *double*
- magnitude\_t\_t\_3 - optional; type *double*
- magnitude\_t\_b - optional; type *double*
- magnitude\_t\_b\_1 - optional; type *double*
- magnitude\_t\_b\_2 - optional; type *double*
- magnitude\_t\_b\_3 - optional; type *double*
- mass\_global - optional; type *double*
- mass\_x - optional; type *double*
- mass\_y - optional; type *double*
- mass\_z - optional; type *double*
- distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_a\_absolute - optional; type *double*
- distance\_a\_relative - optional; type *double*
- distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_b\_absolute - optional; type *double*
- distance\_b\_relative - optional; type *double*
- distance\_c\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_c\_absolute - optional; type *double*
- distance\_c\_relative - optional; type *double*
- count\_n - optional; type *int*
- varying\_load\_parameters\_are\_defined\_as\_relative - optional; type *boolean*
- varying\_load\_parameters - optional; type *array\_of\_member\_set\_load\_varying\_load\_parameters*
  - member\_set\_load\_varying\_load\_parameters - optional, unbounded; type  
*member\_set\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
    - magnitude\_t\_c - optional; type *double*
    - magnitude\_delta\_t - optional; type *double*
    - magnitude\_t\_t - optional; type *double*
    - magnitude\_t\_b - optional; type *double*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- angular\_velocity - optional; type *double*
- angular\_acceleration - optional; type *double*
- axis\_definition\_type - optional; type *member\_set\_load\_axis\_definition\_type* - type *undefined* with restriction -  
enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *member\_set\_load\_axis\_definition\_axis* - type *undefined* with restriction -  
enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *member\_set\_load\_axis\_definition\_axis\_orientation* - type  
*undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- filling\_height - optional; type *double*
- distance\_from\_member\_set\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- has\_force\_eccentricity - optional; type *boolean*
- eccentricity\_horizontal\_alignment - optional; type *member\_set\_load\_eccentricity\_horizontal\_alignment* - type  
*undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_RIGHT' }
- eccentricity\_vertical\_alignment - optional; type *member\_set\_load\_eccentricity\_vertical\_alignment* - type  
*undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_TOP' }
- eccentricity\_section\_middle - optional; type *member\_set\_load\_eccentricity\_section\_middle* - type *undefined*  
with restriction - enum { 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_CENTER\_OF\_GRAVITY',

- 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_NONE', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_SHEAR\_CENTER' }
- is\_eccentricity\_at\_end\_different\_from\_start - optional; type *boolean*
- eccentricity\_y\_at\_start - optional; type *double*
- eccentricity\_z\_at\_start - optional; type *double*
- eccentricity\_y\_at\_end - optional; type *double*
- eccentricity\_z\_at\_end - optional; type *double*
- form\_finding\_internal\_force - optional; type *member\_set\_load\_form\_finding\_internal\_force* - type *undefined* with restriction - enum { 'FORM\_FINDING\_INTERNAL\_FORCE\_COMPRESSION', 'FORM\_FINDING\_INTERNAL\_FORCE\_TENSION' }
- form\_finding\_geometry\_definition - optional; type *member\_set\_load\_form\_finding\_geometry\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LENGTH', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LOW\_POINT\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_MAXIMUM\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_UNSTRESSED\_LENGTH' }
- form\_finding\_force\_definition - optional; type *member\_set\_load\_form\_finding\_force\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_AVERAGE', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_DENSITY', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_HORIZONTAL\_TENSION\_COMPONENT', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MAXIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_J\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_J\_END' }
- form\_finding\_magnitude\_is\_defined\_as\_relative - optional; type *boolean*
- form\_finding\_magnitude\_absolute - optional; type *double*
- form\_finding\_magnitude\_relative - optional; type *double*
- individual\_mass\_components - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 69. get\_member\_set\_representative

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_set\\_representative](http://localhost:8082/get_member_set_representative)

**Input:** [get\\_member\\_set\\_representative\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_set\\_representative](#)

- no type *int*

**Output:** [get\\_member\\_set\\_representative\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type [get\\_member\\_set\\_representativeResponse](#)

- value type [member\\_set\\_representative](#)
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - member\_sets - optional; type *array\_of\_int*
  - comment - optional; type *string*
  - number\_of\_member\_sets - optional; type *int*
  - number\_of\_members - optional; type *int*
  - section\_ids - optional; type *string*
  - total\_length - optional; type *double*
  - total\_volume - optional; type *double*
  - total\_mass - optional; type *double*
  - total\_surface\_of\_coating - optional; type *double*
  - total\_center\_of\_gravity\_x - optional; type *double*
  - total\_center\_of\_gravity\_y - optional; type *double*
  - total\_center\_of\_gravity\_z - optional; type *double*
  - member\_model\_no - optional; type *int*
  - member\_model\_type - optional; type *member\_set\_representative\_member\_model\_type* - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE', 'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }
  - member\_model\_is\_deactivated\_for\_calculation - optional; type *boolean*
  - member\_model\_line - optional; type *int*
  - member\_model\_section\_distribution\_type - optional; type *member\_set\_representative\_member\_model\_section\_distribution\_type* - type *undefined* with restriction - enum { 'SECTION\_DISTRIBUTION\_TYPE\_LINEAR', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
  - member\_model\_reference\_type - optional; type *member\_set\_representative\_member\_model\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - member\_model\_nodes - optional; type *array\_of\_int*
  - member\_model\_node\_start - optional; type *int*
  - member\_model\_node\_end - optional; type *int*
  - member\_model\_analytical\_length - optional; type *double*
  - member\_model\_analytical\_volume - optional; type *double*
  - member\_model\_analytical\_mass - optional; type *double*
  - member\_model\_analytical\_surface\_of\_coating - optional; type *double*
  - member\_model\_analytical\_center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - member\_model\_analytical\_center\_of\_gravity\_x - optional; type *double*

- member\_model\_analytical\_center\_of\_gravity\_y - optional; type *double*
- member\_model\_analytical\_center\_of\_gravity\_z - optional; type *double*
- member\_model\_length - optional; type *double*
- member\_model\_volume - optional; type *double*
- member\_model\_mass - optional; type *double*
- member\_model\_surface\_of\_coating - optional; type *double*
- member\_model\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- member\_model\_center\_of\_gravity\_x - optional; type *double*
- member\_model\_center\_of\_gravity\_y - optional; type *double*
- member\_model\_center\_of\_gravity\_z - optional; type *double*
- member\_set\_model\_position - optional; type *string*
- member\_set\_model\_position\_short - optional; type *string*
- member\_model\_member\_representative - optional; type *int*
- member\_model\_design\_properties\_via\_member - optional; type *boolean*
- member\_model\_design\_properties\_via\_parent\_member\_set - optional; type *boolean*
- member\_model\_design\_properties\_parent\_member\_set - optional; type *int*
- member\_model\_comment - optional; type *string*
- member\_model\_member\_type\_rib\_alignment - optional; type *member\_set\_representative\_member\_model\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
- member\_model\_member\_rib\_first\_surface - optional; type *int*
- member\_model\_member\_rib\_second\_surface - optional; type *int*
- member\_model\_member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
- member\_model\_member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_model\_align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
- member\_model\_flange\_dimensions - optional; type *array\_of\_member\_set\_representative\_member\_model\_flange\_dimensions*
  - member\_set\_representative\_member\_model\_flange\_dimensions - optional, unbounded; type *member\_set\_representative\_member\_model\_flange\_dimensions*
    - no - optional; type *int*
    - end\_ordinate - optional; type *double*
    - length - optional; type *double*
    - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED' }
    - reference\_length - optional; type *double*
    - reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
    - width\_minus\_y\_integrative - optional; type *double*
    - width\_minus\_y\_effictive - optional; type *double*
    - width\_minus\_y\_maximal - optional; type *double*
    - width\_plus\_y\_integrative - optional; type *double*
    - width\_plus\_y\_effictive - optional; type *double*
    - width\_plus\_y\_maximal - optional; type *double*
    - distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- member\_model\_synchronize\_width\_mode - optional; type *boolean*
- member\_model\_relative\_ordinates\_mode - optional; type *boolean*
- member\_model\_member\_type\_definable\_stiffness - optional; type *int*
- member\_model\_result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_set\_representative\_member\_model\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- member\_model\_result\_beam\_y\_z - optional; type *double*
- member\_model\_result\_beam\_y\_plus - optional; type *double*
- member\_model\_result\_beam\_z\_plus - optional; type *double*
- member\_model\_result\_beam\_y\_minus - optional; type *double*
- member\_model\_result\_beam\_z\_minus - optional; type *double*
- member\_model\_result\_beam\_radius - optional; type *double*
- member\_model\_result\_beam\_include\_surfaces - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_surfaces - optional; type *boolean*
- member\_model\_result\_beam\_include\_solids - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_solids - optional; type *boolean*
- member\_model\_result\_beam\_include\_members - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_members - optional; type *boolean*
- member\_model\_result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- member\_model\_result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- member\_model\_result\_beam\_exclude\_members - optional; type *array\_of\_int*
- member\_model\_projected\_length - optional; type *double*
- member\_model\_section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- member\_model\_section\_distance\_from\_start\_absolute - optional; type *double*
- member\_model\_section\_distance\_from\_end\_relative - optional; type *double*
- member\_model\_section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- member\_model\_section\_distance\_from\_end\_absolute - optional; type *double*
- member\_model\_section\_distance\_from\_start\_relative - optional; type *double*
- member\_model\_section\_alignment - optional; type *member\_set\_representative\_member\_model\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- member\_model\_curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- member\_model\_curved\_member\_is\_cantilevers - optional; type *boolean*
- member\_model\_curved\_member\_cantilevers\_type - optional; type *member\_set\_representative\_member\_model\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- member\_model\_rotation\_specification\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- member\_model\_rotation\_angle - optional; type *double*
- member\_model\_rotation\_help\_node - optional; type *int*

- member\_model\_rotation\_plane\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- member\_model\_rotation\_surface - optional; type *int*
- member\_model\_rotation\_surface\_plane\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- member\_model\_is\_rotated - optional; type *boolean*
- member\_model\_section\_start - optional; type *int*
- member\_model\_section\_end - optional; type *int*
- member\_model\_section\_internal - optional; type *int*
- member\_model\_section\_material - optional; type *int*
- member\_model\_member\_hinge\_start - optional; type *int*
- member\_model\_member\_hinge\_end - optional; type *int*
- member\_model\_member\_eccentricity\_start - optional; type *int*
- member\_model\_member\_eccentricity\_end - optional; type *int*
- member\_model\_support - optional; type *int*
- member\_model\_member\_transverse\_stiffener - optional; type *int*
- member\_model\_member\_nonlinearity - optional; type *int*
- member\_model\_member\_result\_intermediate\_point - optional; type *int*
- member\_model\_concrete\_effective\_lengths - optional; type *int*
- member\_model\_steel\_effective\_lengths - optional; type *int*
- member\_model\_timber\_effective\_lengths - optional; type *int*
- member\_model\_aluminum\_effective\_lengths - optional; type *int*
- member\_model\_concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans*
  - member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTANCES', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTANCES' }
    - stirrup\_start\_offset - optional; type *double*
    - stirrup\_end\_offset - optional; type *double*
    - one\_stirrup\_length - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_length - optional; type *string*
    - length - optional; type *double*
    - one\_stirrup\_weight - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
    - weight - optional; type *double*
- member\_model\_concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*

- bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
- bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
- bar\_diameter\_uniformly\_surrounding - optional; type *double*
- bar\_diameter\_line - optional; type *double*
- bar\_diameter\_single - optional; type *double*
- bar\_diameter\_corner - optional; type *double*
- bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
- bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
- bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
- bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
- bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
- bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
- bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
- bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
- corner\_reinforcement\_enabled - optional; type *boolean*
- reinforcement\_area\_symmetrical - optional; type *double*
- reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
- reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
- reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
- reinforcement\_area\_uniformly\_surrounding - optional; type *double*
- reinforcement\_area\_line - optional; type *double*
- reinforcement\_area\_single - optional; type *double*
- reinforcement\_area\_corner - optional; type *double*
- reinforcement\_area\_total - optional; type *double*
- span\_position\_reference\_internal\_node - optional; type *int*
- span\_position\_reference\_x\_location\_relative - optional; type *double*
- span\_position\_reference\_x\_location\_absolute - optional; type *double*
- span\_start\_relative - optional; type *double*
- span\_start\_absolute - optional; type *double*
- span\_end\_relative - optional; type *double*
- span\_end\_absolute - optional; type *double*
- span\_length - optional; type *double*
- additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- additional\_offset\_top\_side - optional; type *double*
- additional\_offset\_bottom\_side - optional; type *double*
- additional\_offset\_left\_side - optional; type *double*
- additional\_offset\_right\_side - optional; type *double*
- additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_end\_type - optional; type *additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE' }



'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }

- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*

- member\_model\_concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- member\_model\_concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- member\_model\_concrete\_cover - optional; type *double*
- member\_model\_concrete\_cover\_top - optional; type *double*
- member\_model\_concrete\_cover\_bottom - optional; type *double*
- member\_model\_concrete\_cover\_left - optional; type *double*
- member\_model\_concrete\_cover\_right - optional; type *double*
- member\_model\_concrete\_cover\_min - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_top - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_left - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_right - optional; type *array\_of\_int*
- member\_model\_concrete\_durability - optional; type *int*
- member\_model\_concrete\_durability\_top - optional; type *int*
- member\_model\_concrete\_durability\_bottom - optional; type *int*
- member\_model\_concrete\_durability\_left - optional; type *int*
- member\_model\_concrete\_durability\_right - optional; type *int*
- member\_model\_steel\_boundary\_conditions - optional; type *int*
- member\_model\_steel\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_steel\_member\_transverse\_weld - optional; type *int*
- member\_model\_steel\_member\_shear\_panel - optional; type *int*
- member\_model\_steel\_member\_rotational\_restraint - optional; type *int*
- member\_model\_timber\_service\_class - optional; type *int*
- member\_model\_timber\_moisture\_class - optional; type *int*
- member\_model\_timber\_service\_conditions - optional; type *int*
- member\_model\_timber\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_timber\_member\_shear\_panel - optional; type *int*
- member\_model\_timber\_member\_rotational\_restraint - optional; type *int*
- member\_model\_aluminum\_boundary\_conditions - optional; type *int*
- member\_model\_aluminum\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_aluminum\_member\_transverse\_weld - optional; type *int*
- member\_model\_aluminum\_member\_shear\_panel - optional; type *int*
- member\_model\_aluminum\_member\_rotational\_restraint - optional; type *int*
- member\_model\_stress\_analysis\_configuration - optional; type *int*
- member\_model\_member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_model\_end\_modifications\_member\_start\_extension - optional; type *double*
- member\_model\_end\_modifications\_member\_start\_slope\_y - optional; type *double*
- member\_model\_end\_modifications\_member\_start\_slope\_z - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_extension - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_slope\_y - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_slope\_z - optional; type *double*
- member\_model\_has\_any\_end\_modifications - optional; type *boolean*
- member\_model\_deflection\_check\_direction - optional; type *member\_set\_representative\_member\_model\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- member\_model\_deflection\_check\_displacement\_reference - optional; type *member\_set\_representative\_member\_model\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }

- member\_model\_design\_support\_on\_member\_start - optional; type *int*
- member\_model\_design\_support\_on\_member\_end - optional; type *int*
- member\_model\_design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes*
  - member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- member\_model\_is\_generated - optional; type *boolean*
- member\_model\_generating\_object\_info - optional; type *string*
- member\_set\_model\_no - optional; type *int*
- member\_set\_model\_user\_defined\_name\_enabled - optional; type *boolean*
- member\_set\_model\_name - optional; type *string*
- member\_set\_model\_set\_type - optional; type *member\_set\_representative\_member\_set\_model\_set\_type* - type *undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
- member\_set\_model\_members - optional; type *array\_of\_int*
- member\_set\_model\_length - optional; type *double*
- member\_set\_model\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- member\_set\_model\_center\_of\_gravity\_x - optional; type *double*
- member\_set\_model\_center\_of\_gravity\_y - optional; type *double*
- member\_set\_model\_center\_of\_gravity\_z - optional; type *double*
- member\_set\_model\_volume - optional; type *double*
- member\_set\_model\_mass - optional; type *double*
- member\_set\_model\_member\_set\_representative - optional; type *int*
- member\_set\_model\_discontinuous\_torsional\_warping - optional; type *boolean*
- member\_set\_model\_design\_properties\_activated - optional; type *boolean*
- member\_set\_model\_steel\_effective\_lengths - optional; type *int*
- member\_set\_model\_steel\_boundary\_conditions - optional; type *int*
- member\_set\_model\_steel\_member\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_steel\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_steel\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_seismic\_configuration - optional; type *int*
- member\_set\_model\_aluminum\_effective\_lengths - optional; type *int*
- member\_set\_model\_aluminum\_boundary\_conditions - optional; type *int*
- member\_set\_model\_aluminum\_member\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_aluminum\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_aluminum\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_aluminum\_member\_transverse\_weld - optional; type *int*
- member\_set\_model\_member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans*
  - member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DIS' }
    - stirrup\_start\_offset - optional; type *double*
    - stirrup\_end\_offset - optional; type *double*
    - one\_stirrup\_length - optional; type *double*
    - minimal\_and\_maximal\_stirrup\_length - optional; type *string*
    - length - optional; type *double*

- one\_stirrup\_weight - optional; type *double*
- minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
- weight - optional; type *double*
- member\_set\_model\_concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_top\_side - optional; type *double*
    - additional\_offset\_bottom\_side - optional; type *double*
    - additional\_offset\_left\_side - optional; type *double*
    - additional\_offset\_right\_side - optional; type *double*
    - additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
    - additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }

'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }

- additional\_offset\_reference\_type\_at\_end\_type - optional; type *undefined* with restriction - enum {  
*additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- member\_set\_model\_concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- member\_set\_model\_concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- member\_set\_model\_concrete\_cover - optional; type *double*
- member\_set\_model\_concrete\_cover\_top - optional; type *double*
- member\_set\_model\_concrete\_cover\_bottom - optional; type *double*
- member\_set\_model\_concrete\_cover\_left - optional; type *double*
- member\_set\_model\_concrete\_cover\_right - optional; type *double*
- member\_set\_model\_concrete\_cover\_min - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_top - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_left - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_right - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_durability - optional; type *int*
- member\_set\_model\_concrete\_durability\_top - optional; type *int*
- member\_set\_model\_concrete\_durability\_bottom - optional; type *int*
- member\_set\_model\_concrete\_durability\_left - optional; type *int*
- member\_set\_model\_concrete\_durability\_right - optional; type *int*
- member\_set\_model\_concrete\_effective\_lengths - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_set\_model\_service\_class\_timber\_design - optional; type *int*
- member\_set\_model\_moisture\_class\_timber\_design - optional; type *int*
- member\_set\_model\_timber\_effective\_lengths - optional; type *int*
- member\_set\_model\_service\_conditions\_timber\_design - optional; type *int*
- member\_set\_model\_timber\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_timber\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_timber\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_transverse\_stiffener - optional; type *int*
- member\_set\_model\_stress\_analysis\_configuration - optional; type *int*
- member\_set\_model\_deflection\_check\_direction - optional; type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- member\_set\_model\_deflection\_check\_displacement\_reference - optional; type *member\_set\_representative\_member\_set\_model\_deflection\_check\_displacement\_reference* - type

*undefined* with restriction - enum {  
 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS',  
 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }

- member\_set\_model\_design\_support\_on\_member\_set\_start - optional; type *int*
- member\_set\_model\_design\_support\_on\_member\_set\_end - optional; type *int*
- member\_set\_model\_design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes*
  - member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- member\_set\_model\_comment - optional; type *string*
- member\_set\_model\_is\_generated - optional; type *boolean*
- member\_set\_model\_generating\_object\_info - optional; type *string*
- member\_set\_model\_member\_set\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_y\_axis - optional; type *array\_of\_member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis*
  - member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_model\_deflection\_segments\_z\_axis - optional; type *array\_of\_member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis*
  - member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_set\_model\_deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- member\_set\_model\_deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- member\_set\_model\_deflection\_segments\_y\_axis - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis*
  - member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_set\_model\_deflection\_segments\_z\_axis - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis*
  - member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 70. get\_member\_stiffness\_modification

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_member\\_stiffness\\_modification](http://localhost:8082/get_member_stiffness_modification)

**Input:** [get\\_member\\_stiffness\\_modification\\_request](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_member\\_stiffness\\_modification](#)

- no type *int*

**Output:** [get\\_member\\_stiffness\\_modification\\_response](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_member\\_stiffness\\_modificationResponse](#)

- value type [member\\_stiffness\\_modification](#)
  - no type *int*
  - type - optional; type [member\\_stiffness\\_modification\\_type](#) - type *undefined* with restriction - enum {  
 'TYPE\_CONCRETE\_STRUCTURES\_ACI', 'TYPE\_CONCRETE\_STRUCTURES\_CSA',  
 'TYPE\_PARTIAL\_STIFFNESSES\_FACTORS', 'TYPE\_STEEL\_STRUCTURES', 'TYPE\_STEEL\_STRUCTURES\_CSA',  
 'TYPE\_TOTAL\_STIFFNESSES\_FACTORS' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - factor\_of\_axial\_stiffness - optional; type *double*
  - factor\_of\_bending\_y\_stiffness - optional; type *double*
  - factor\_of\_bending\_z\_stiffness - optional; type *double*
  - partial\_stiffness\_factor\_of\_shear\_y\_stiffness - optional; type *double*
  - partial\_stiffness\_factor\_of\_shear\_z\_stiffness - optional; type *double*
  - partial\_stiffness\_factor\_of\_torsion\_stiffness - optional; type *double*
  - partial\_stiffness\_factor\_of\_weight - optional; type *double*
  - total\_stiffness\_factor\_of\_total\_stiffness - optional; type *double*
  - steel\_structure\_csa\_stiffness\_factor\_of\_shear\_y\_stiffness - optional; type *double*
  - steel\_structure\_csa\_stiffness\_factor\_of\_shear\_z\_stiffness - optional; type *double*
  - steel\_structure\_csa\_stiffness\_factor\_of\_torsion\_stiffness - optional; type *double*
  - steel\_structure\_csa\_factor\_of\_axial\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_factor\_of\_bending\_y\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_factor\_of\_bending\_z\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_factor\_of\_shear\_y\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_factor\_of\_shear\_z\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_stiffness\_factor\_of\_torsion\_stiffness\_enable - optional; type *boolean*
  - steel\_structure\_csa\_determine\_tau\_b - optional; type *member\_stiffness\_modification\_steel\_structure\_csa\_determine\_tau\_b* - type *undefined* with restriction -

- enum { 'ITERATIVE', 'SET\_TO\_1' }
- steel\_structure\_gb\_direct\_method\_enabled - optional; type *boolean*
- steel\_structure\_determine\_tau\_b - optional; type *member\_stiffness\_modification\_steel\_structure\_determine\_tau\_b* - type *undefined* with restriction - enum { 'ITERATIVE', 'SET\_TO\_1' }
- steel\_structure\_design\_method - optional; type *member\_stiffness\_modification\_steel\_structure\_design\_method* - type *undefined* with restriction - enum { 'ASD', 'LRFD' }
- concrete\_structure\_component\_type - optional; type *member\_stiffness\_modification\_concrete\_structure\_component\_type* - type *undefined* with restriction - enum { 'COMPONENT\_TYPE\_BEAMS', 'COMPONENT\_TYPE\_COLUMNS', 'COMPONENT\_TYPE\_FLAT\_PLATES\_AND\_FLAT\_SLABS', 'COMPONENT\_TYPE\_WALLS\_CRACKED', 'COMPONENT\_TYPE\_WALLS\_UNCRACKED' }
- assigned\_to\_structure\_modification - optional; type *array\_of\_int*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 71. get\_member\_support

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_member\_support

**Input:** get\_member\_support\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_support*

- no type *int*

**Output:** get\_member\_support\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_supportResponse*

- value type *member\_support*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - members - optional; type *array\_of\_int*
  - spring\_translation - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - spring\_rotation - optional; type *double*
  - spring\_translation\_x - optional; type *double*
  - spring\_translation\_y - optional; type *double*
  - spring\_translation\_z - optional; type *double*
  - spring\_shear - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - spring\_shear\_x - optional; type *double*
  - spring\_shear\_y - optional; type *double*
  - spring\_shear\_z - optional; type *double*
  - nonlinearity - optional; type *member\_support\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_FAILURE\_IF\_NEGATIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_FAILURE\_IF\_POSITIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_NONE' }
  - support\_dimensions\_enabled - optional; type *boolean*
  - support\_width\_y - optional; type *double*
  - support\_width\_z - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 72. get\_member\_transverse\_stiffener

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_member\_transverse\_stiffener

**Input:** get\_member\_transverse\_stiffener\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_transverse\_stiffener*

- no type *int*

**Output:** get\_member\_transverse\_stiffener\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_member\_transverse\_stiffenerResponse*

- value type *member\_transverse\_stiffener*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - members - optional; type *array\_of\_int*
  - member\_sets - optional; type *array\_of\_int*
  - components - optional; type *array\_of\_member\_transverse\_stiffener\_components*
    - member\_transverse\_stiffener\_components - optional, unbounded; type *member\_transverse\_stiffener\_components*
      - no - optional; type *int*
      - stiffener\_type - optional; type *stiffener\_type* - type *undefined* with restriction - enum { 'STIFFENER\_COMPONENT\_TYPE\_ANGLE', 'STIFFENER\_COMPONENT\_TYPE\_CHANNEL\_SECTION', 'STIFFENER\_COMPONENT\_TYPE\_CONNECTING\_COLUMN\_END', 'STIFFENER\_COMPONENT\_TYPE\_CONNECTING\_COLUMN\_START', 'STIFFENER\_COMPONENT\_TYPE\_END\_PLATE\_END', 'STIFFENER\_COMPONENT\_TYPE\_END\_PLATE\_START', 'STIFFENER\_COMPONENT\_TYPE\_FLAT', 'STIFFENER\_COMPONENT\_TYPE\_WARPING\_RESTRAINT' }

- position - optional; type *double*
- position\_type - optional; type *position\_type* - type *undefined* with restriction - enum { 'STIFFENER\_COMPONENT\_POSITION\_DOUBLE\_SIDED', 'STIFFENER\_COMPONENT\_POSITION\_SINGLE\_SIDED\_LEFT', 'STIFFENER\_COMPONENT\_POSITION\_SINGLE\_SIDED\_RIGHT' }
- multiple - optional; type *boolean*
- note - optional; type *string*
- multiple\_number - optional; type *int*
- multiple\_offset\_definition\_type - optional; type *multiple\_offset\_definition\_type* - type *undefined* with restriction - enum { 'OFFSET\_DEFINITION\_TYPE\_ABSOLUTE', 'OFFSET\_DEFINITION\_TYPE\_RELATIVE' }
- multiple\_offset - optional; type *double*
- material - optional; type *int*
- consider\_stiffener - optional; type *boolean*
- thickness - optional; type *double*
- width - optional; type *double*
- height - optional; type *double*
- non\_rigid - optional; type *boolean*
- rigid - optional; type *boolean*
- width\_b\_u - optional; type *double*
- height\_h\_u - optional; type *double*
- thickness\_t\_u - optional; type *double*
- thickness\_s\_u - optional; type *double*
- width\_b - optional; type *double*
- thickness\_t - optional; type *double*
- column\_section - optional; type *int*
- height\_h\_m - optional; type *double*
- section - optional; type *int*
- cantilever\_l\_k - optional; type *double*
- full\_warping\_restraint - optional; type *boolean*
- user\_defined\_restraint - optional; type *boolean*
- user\_defined\_restraint\_value - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 73. get\_mesh\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_mesh\\_settings](http://localhost:8082/get_mesh_settings)

**Input:** [get\\_mesh\\_settings\\_request](#) (soap:body, use = literal) [Source code](#)

parameters [type get\\_mesh\\_settings](#)

**Output:** [get\\_mesh\\_settings\\_response](#) (soap:body, use = literal) [Source code](#)

parameters [type get\\_mesh\\_settingsResponse](#)

- value [type meshConfig\\_type](#)
  - general\_target\_length\_of\_fe [type double](#)
  - general\_maximum\_distance\_between\_node\_and\_line [type double](#)
  - general\_maximum\_number\_of\_mesh\_nodes [type int](#)
  - members\_number\_of\_divisions\_for\_special\_types [type int](#)
  - members\_activate\_division\_for\_large\_deformation\_and\_post\_critical\_analysis [type boolean](#)
  - members\_number\_of\_divisions\_for\_result\_diagram [type int](#)
  - members\_number\_of\_divisions\_for\_min\_max\_values [type int](#)
  - members\_use\_division\_for\_concrete\_members [type boolean](#)
  - members\_number\_of\_divisions\_for\_concrete\_members [type int](#)
  - members\_use\_division\_for\_straight\_members [type boolean](#)
  - members\_division\_for\_straight\_members\_type [type boolean](#)
  - members\_length\_of\_fe [type double](#)
  - members\_minimum\_number\_of\_divisions [type int](#)
  - members\_use\_division\_for\_members\_with\_nodes [type boolean](#)
  - surfaces\_maximum\_ratio\_of\_fe [type double](#)
  - surfaces\_maximum\_out\_of\_plane\_inclination [type double](#)
  - surfaces\_mesh\_refinement [type boolean](#)
  - surfaces\_relationship [type double](#)
  - surfaces\_integrate\_also\_unutilized\_objects [type boolean](#)
  - surfaces\_shape\_of\_finite\_elements [type meshConfig\\_surfaces\\_shape\\_of\\_finite\\_elements\\_type](#) - type *undefined* with restriction - enum { 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_QUADRANGLES\_ONLY', 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_TRIANGLES\_AND\_QUADRANGLES', 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_TRIANGLES\_ONLY' }
  - surfaces\_same\_squares [type boolean](#)
  - surfaces\_triangles\_for\_membranes [type boolean](#)
  - surfaces\_mapped\_mesh\_preferred [type boolean](#)
  - solids\_use\_refinement\_if\_containing\_close\_nodes [type boolean](#)
  - solids\_maximum\_number\_of\_elements [type int](#)
  - solids\_use\_target\_length\_of\_fe\_for\_type\_soil [type boolean](#)
  - solids\_target\_length\_of\_fe\_for\_type\_soil [type double](#)
  - SurfacesMeshQualityConfig [type meshConfig\\_SurfacesMeshQualityConfig\\_type](#)
    - mesh\_quality\_color\_indicator\_ok\_color [type color](#) - type *undefined* with restriction
    - mesh\_quality\_color\_indicator\_warning\_color [type color](#) - type *undefined* with restriction
    - mesh\_quality\_color\_indicator\_failure\_color [type color](#) - type *undefined* with restriction
    - QualityCriteriaConfig [type meshConfig\\_SurfacesMeshQualityConfig\\_QualityCriteriaConfig\\_type](#)
      - quality\_criterion\_check\_aspect\_ratio [type boolean](#)
      - quality\_criterion\_check\_aspect\_ratio\_warning [type double](#)
      - quality\_criterion\_check\_aspect\_ratio\_failure [type double](#)
      - quality\_criterion\_parallel\_deviations [type boolean](#)
      - quality\_criterion\_parallel\_deviations\_warning [type double](#)
      - quality\_criterion\_parallel\_deviations\_failure [type double](#)
      - quality\_criterion\_corner\_angles\_of\_triangle\_elements [type boolean](#)
      - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_warning [type double](#)
      - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_failure [type double](#)

- quality\_criterion\_corner\_angles\_of\_quadrangle\_elements type *boolean*
- quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_warning type *double*
- quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_failure type *double*
- quality\_criterion\_warping\_of\_membrane\_elements type *boolean*
- quality\_criterion\_warping\_of\_membrane\_elements\_warning type *double*
- quality\_criterion\_warping\_of\_membrane\_elements\_failure type *double*
- quality\_criterion\_warping\_of\_non\_membrane\_elements type *boolean*
- quality\_criterion\_warping\_of\_non\_membrane\_elements\_warning type *double*
- quality\_criterion\_warping\_of\_non\_membrane\_elements\_failure type *double*
- quality\_criterion\_jacobian\_ratio type *boolean*
- quality\_criterion\_jacobian\_ratio\_warning type *double*
- quality\_criterion\_jacobian\_ratio\_failure type *double*
- SolidsMeshQualityConfig type *meshConfig\_SolidsMeshQualityConfig\_type*
  - mesh\_quality\_color\_indicator\_ok\_color type *color* - type *undefined* with restriction
  - mesh\_quality\_color\_indicator\_warning\_color type *color* - type *undefined* with restriction
  - mesh\_quality\_color\_indicator\_failure\_color type *color* - type *undefined* with restriction
  - QualityCriteriaConfig type *meshConfig\_QualityCriteriaConfig\_type*
    - quality\_criterion\_check\_aspect\_ratio type *boolean*
    - quality\_criterion\_check\_aspect\_ratio\_warning type *double*
    - quality\_criterion\_check\_aspect\_ratio\_failure type *double*
    - quality\_criterion\_parallel\_deviations type *boolean*
    - quality\_criterion\_parallel\_deviations\_warning type *double*
    - quality\_criterion\_parallel\_deviations\_failure type *double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements type *boolean*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_warning type *double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_failure type *double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements type *boolean*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_warning type *double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_failure type *double*
    - quality\_criterion\_warping type *boolean*
    - quality\_criterion\_warping\_warning type *double*
    - quality\_criterion\_warping\_failure type *double*
    - quality\_criterion\_jacobian\_ratio type *boolean*
    - quality\_criterion\_jacobian\_ratio\_warning type *double*
    - quality\_criterion\_jacobian\_ratio\_failure type *double*
- windSimulationMeshConfig type *meshConfig\_windSimulationMeshConfig\_type*
  - windsimulation\_mesh\_config\_value\_simplify\_model type *boolean*
  - windsimulation\_mesh\_config\_value\_determine\_details\_by type *meshConfig\_windSimulationMeshConfig\_windsimulation\_mesh\_config\_value\_determine\_details\_by\_type* - type *undefined* with restriction - enum { 'E\_WINDSIMULATION\_DETERMINE\_DETAILS\_BY\_\_DETAIL\_SIZE', 'E\_WINDSIMULATION\_DETERMINE\_DETAILS\_BY\_\_LEVEL\_OF\_DETAILS' }
  - windsimulation\_mesh\_config\_value\_level\_of\_details type *int*
  - windsimulation\_mesh\_config\_value\_detail\_size type *double*
  - windsimulation\_mesh\_config\_value\_small\_openings\_closure\_type type *meshConfig\_windSimulationMeshConfig\_windsimulation\_mesh\_config\_value\_small\_openings\_closure\_type* - type *undefined* with restriction - enum { 'E\_WINDSIMULATION\_OPENINGS\_CLOSURE\_TYPE\_PERCENT\_OF\_MODEL\_DIAMETER', 'E\_WINDSIMULATION\_OPENINGS\_CLOSURE\_TYPE\_REAL\_SIZE' }
  - windsimulation\_mesh\_config\_value\_small\_openings\_closure\_value type *double*
  - windsimulation\_mesh\_config\_value\_optimized\_member\_topology type *boolean*
  - windsimulation\_mesh\_config\_value\_optimized\_member\_topo\_value type *int*
  - windsimulation\_mesh\_config\_value\_active\_objects\_only type *boolean*
  - windsimulation\_mesh\_config\_value\_terrain type *boolean*
  - windsimulation\_mesh\_config\_value\_terrain\_from\_model type *boolean*
  - windsimulation\_mesh\_config\_value\_terrain\_objects\_id type *string*
  - windsimulation\_mesh\_config\_value\_terrain\_objects\_all type *boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model type *boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_ifc\_objects\_id type *string*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_ifc\_objects\_all type *boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_visual\_objects\_id type *string*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_visual\_objects\_all type *boolean*
  - windsimulation\_mesh\_config\_value\_keep\_results\_if\_mesh\_deleted type *boolean*
  - windsimulation\_mesh\_config\_value\_consider\_surface\_thickness type *boolean*
  - windsimulation\_mesh\_config\_value\_run\_rwind\_silent type *boolean*

#### 74. get\_model\_info

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_model\_info

**Input:** get\_model\_info\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_info*

**Output:** get\_model\_info\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_infoResponse*

- value type *modelInfo\_type*
  - property\_node\_count type *int*
  - property\_line\_count type *int*
  - property\_surface\_count type *int*
  - property\_member\_count type *int*
  - property\_solid\_count type *int*
  - property\_lc\_count type *int*
  - property\_co\_count type *int*
  - property\_cs\_count type *int*
  - property\_rc\_count type *int*
  - property\_has\_results type *boolean*
  - property\_has\_printout\_report type *boolean*
  - property\_weight type *double*
  - property\_dimensions type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - property\_rsection\_area type *double*



- property\_rsection\_point\_count type *int*
- property\_rsection\_line\_count type *int*
- property\_rsection\_part\_count type *int*
- property\_rsection\_element\_count type *int*
- property\_rsection\_stress\_point\_count type *int*
- property\_rsection\_is\_calculable type *boolean*

#### 75. get\_model\_main\_parameters

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_model\_main\_parameters

**Input:** get\_model\_main\_parameters\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_main\_parameters*

**Output:** get\_model\_main\_parameters\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_main\_parametersResponse*

- value type *model\_main\_parameters*
  - model\_id type *string*
  - model\_name type *string*
  - model\_description type *string*
  - model\_comment type *string*
  - model\_path type *string*
  - project\_id type *string*
  - project\_name type *string*
  - project\_description type *string*
  - project\_folder type *string*

#### 76. get\_model\_type

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_model\_type

**Input:** get\_model\_type\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_type*

**Output:** get\_model\_type\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_model\_typeResponse*

- value type *model\_type* - type *undefined* with restriction - enum { 'E\_MODEL\_TYPE\_1D\_X\_3D', 'E\_MODEL\_TYPE\_1D\_X\_AXIAL', 'E\_MODEL\_TYPE\_2D\_XY\_3D', 'E\_MODEL\_TYPE\_2D\_XY\_PLATE', 'E\_MODEL\_TYPE\_2D\_XZ\_3D', 'E\_MODEL\_TYPE\_2D\_XZ\_PLANE\_STRAIN', 'E\_MODEL\_TYPE\_2D\_XZ\_PLANE\_STRESS', 'E\_MODEL\_TYPE\_3D' }

#### 77. get\_nodal\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_nodal\_load

**Input:** get\_nodal\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_nodal\_load*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_nodal\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_nodal\_loadResponse*

- value type *nodal\_load*
  - no type *int*
  - load\_type - optional; type *nodal\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_COMPONENTS', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MASS', 'LOAD\_TYPE\_MOMENT' }
  - nodes - optional; type *array\_of\_int*
  - force\_eccentricity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - force\_eccentricity\_x - optional; type *double*
  - force\_eccentricity\_y - optional; type *double*
  - force\_eccentricity\_z - optional; type *double*
  - has\_force\_eccentricity - optional; type *boolean*
  - coordinate\_system - optional; type *int*
  - has\_specific\_direction - optional; type *boolean*
  - specific\_direction\_type - optional; type *nodal\_load\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
  - axes\_sequence - optional; type *nodal\_load\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - rotated\_about\_angle\_x - optional; type *double*
  - rotated\_about\_angle\_y - optional; type *double*
  - rotated\_about\_angle\_z - optional; type *double*
  - rotated\_about\_angle\_1 - optional; type *double*
  - rotated\_about\_angle\_2 - optional; type *double*
  - rotated\_about\_angle\_3 - optional; type *double*
  - directed\_to\_node\_direction\_node - optional; type *int*
  - parallel\_to\_two\_nodes\_first\_node - optional; type *int*
  - parallel\_to\_two\_nodes\_second\_node - optional; type *int*
  - parallel\_to\_line - optional; type *int*
  - parallel\_to\_member - optional; type *int*

- components\_force - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- components\_force\_x - optional; type *double*
- components\_force\_y - optional; type *double*
- components\_force\_z - optional; type *double*
- components\_moment - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- components\_moment\_x - optional; type *double*
- components\_moment\_y - optional; type *double*
- components\_moment\_z - optional; type *double*
- force\_magnitude - optional; type *double*
- load\_direction - optional; type *nodal\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
- moment\_magnitude - optional; type *double*
- mass\_global - optional; type *double*
- mass - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- mass\_x - optional; type *double*
- mass\_y - optional; type *double*
- mass\_z - optional; type *double*
- individual\_mass\_components - optional; type *boolean*
- mass\_moment\_of\_inertia - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- mass\_moment\_of\_inertia\_x - optional; type *double*
- mass\_moment\_of\_inertia\_y - optional; type *double*
- mass\_moment\_of\_inertia\_z - optional; type *double*
- mass\_has\_rotational\_mass - optional; type *boolean*
- mass\_rotational\_mass - optional; type *double*
- mass\_angular\_velocity - optional; type *double*
- mass\_angular\_acceleration - optional; type *double*
- mass\_radius - optional; type *double*
- mass\_axis\_of\_rotation - optional; type *nodal\_load\_mass\_axis\_of\_rotation* - type *undefined* with restriction - enum { 'AXIS\_OF\_ROTATION\_X\_NEGATIVE', 'AXIS\_OF\_ROTATION\_X\_POSITIVE', 'AXIS\_OF\_ROTATION\_Y\_NEGATIVE', 'AXIS\_OF\_ROTATION\_Y\_POSITIVE', 'AXIS\_OF\_ROTATION\_Z\_NEGATIVE', 'AXIS\_OF\_ROTATION\_Z\_POSITIVE' }
- mass\_angle - optional; type *double*
- has\_shifted\_display - optional; type *boolean*
- offset - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- offset\_x - optional; type *double*
- offset\_y - optional; type *double*
- offset\_z - optional; type *double*
- size\_or\_distance - optional; type *double*
- comment - optional; type *string*
- load\_case - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 78. get\_nodal\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_nodal\_mesh\_refinement

**Input:** get\_nodal\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_nodal\_mesh\_refinement*

- no type *int*

**Output:** get\_nodal\_mesh\_refinement\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_nodal\_mesh\_refinementResponse*

- value type *nodal\_mesh\_refinement*
  - no type *int*
  - type - optional; type *nodal\_mesh\_refinement\_type* - type *undefined* with restriction - enum { 'TYPE\_CIRCULAR', 'TYPE\_RECTANGULAR' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - comment - optional; type *string*
  - nodes - optional; type *array\_of\_int*
  - circular\_radius - optional; type *double*
  - circular\_target\_inner\_length - optional; type *double*
  - circular\_target\_outer\_length - optional; type *double*
  - circular\_length\_arrangement - optional; type *nodal\_mesh\_refinement\_circular\_length\_arrangement* - type *undefined* with restriction - enum { 'LENGTH\_ARRANGEMENT\_COMBINED', 'LENGTH\_ARRANGEMENT\_GRADUALLY', 'LENGTH\_ARRANGEMENT\_RADIAL' }
  - rectangular\_side - optional; type *double*
  - rectangular\_target\_inner\_length - optional; type *double*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - apply\_only\_on\_selected\_surfaces - optional; type *boolean*
  - selected\_surfaces - optional; type *array\_of\_int*
  - id\_for\_export\_import - optional; type *string*

- `metadata_for_export_import` - optional; type *string*

## 79. `get_nodal_support`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_nodal_support`

**Input:** `get_nodal_support_request` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_nodal\_support*

- no type *int*

**Output:** `get_nodal_support_response` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_nodal\_supportResponse*

- value type *nodal\_support*
  - no type *int*
  - `user_defined_name_enabled` - optional; type *boolean*
  - `name` - optional; type *string*
  - `nodes` - optional; type *array\_of\_int*
  - `spring` - optional; type *vector\_3d*
    - `x` type *double*
    - `y` type *double*
    - `z` type *double*
  - `spring_x` - optional; type *double*
  - `spring_y` - optional; type *double*
  - `spring_z` - optional; type *double*
  - `rotational_restraint` - optional; type *vector\_3d*
    - `x` type *double*
    - `y` type *double*
    - `z` type *double*
  - `rotational_restraint_x` - optional; type *double*
  - `rotational_restraint_y` - optional; type *double*
  - `rotational_restraint_z` - optional; type *double*
  - `spring_x_nonlinearity` - optional; type *nodal\_support\_spring\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `spring_y_nonlinearity` - optional; type *nodal\_support\_spring\_y\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `spring_z_nonlinearity` - optional; type *nodal\_support\_spring\_z\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `rotational_restraint_x_nonlinearity` - optional; type *nodal\_support\_rotational\_restraint\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `rotational_restraint_y_nonlinearity` - optional; type *nodal\_support\_rotational\_restraint\_y\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `rotational_restraint_z_nonlinearity` - optional; type *nodal\_support\_rotational\_restraint\_z\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - `axes_sequence` - optional; type *nodal\_support\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - `column_base_semi_rigid` - optional; type *double*
  - `column_base_support_type` - optional; type *nodal\_support\_column\_base\_support\_type* - type *undefined* with restriction - enum { 'BASE\_SUPPORT\_TYPE\_HINGED', 'BASE\_SUPPORT\_TYPE\_RIGID', 'BASE\_SUPPORT\_TYPE\_SEMI\_RIGID' }
  - `column_head_support_type` - optional; type *nodal\_support\_column\_head\_support\_type* - type *undefined* with restriction - enum { 'HEAD\_SUPPORT\_TYPE\_HINGED', 'HEAD\_TYPE\_SUPPORT\_SEMI\_RIGID' }
  - `column_head_type` - optional; type *nodal\_support\_column\_head\_type* - type *undefined* with restriction - enum { 'HEAD\_TYPE\_CIRCULAR', 'HEAD\_TYPE\_RECTANGULAR' }
  - `column_height` - optional; type *double*
  - `column_material` - optional; type *int*
  - `column_rotation` - optional; type *double*
  - `column_rotational_restraint_x` - optional; type *double*

- `column_rotational_restraint_y` - optional; type *double*
- `column_section` - optional; type *int*
- `column_section_same_as_head` - optional; type *boolean*
- `column_shear_stiffness` - optional; type *boolean*
- `column_spring_x` - optional; type *double*
- `column_spring_y` - optional; type *double*
- `column_spring_z` - optional; type *double*
- `column_support_type` - optional; type *nodal\_support\_column\_support\_type* - type *undefined* with restriction - enum { 'SUPPORT\_TYPE\_ELASTIC\_NODAL\_SUPPORT', 'SUPPORT\_TYPE\_ELASTIC\_SURFACE\_FOUNDATIONS', 'SUPPORT\_TYPE\_WITH\_ADAPTED\_FE\_MESH' }
- `column_width_x` - optional; type *double*
- `column_width_y` - optional; type *double*
- `comment` - optional; type *string*
- `coordinate_system` - optional; type *int*
- `diagram_along_x_end` - optional; type *nodal\_support\_diagram\_along\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_x_is_sorted` - optional; type *boolean*
- `diagram_along_x_start` - optional; type *nodal\_support\_diagram\_along\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_x_symmetric` - optional; type *boolean*
- `diagram_along_x_table` - optional; type *array\_of\_nodal\_support\_diagram\_along\_x\_table*
  - `nodal_support_diagram_along_x_table` - optional, unbounded; type *nodal\_support\_diagram\_along\_x\_table*
    - `no` - optional; type *int*
    - `displacement` - optional; type *double*
    - `force` - optional; type *double*
    - `spring` - optional; type *double*
    - `note` - optional; type *string*
- `diagram_along_y_end` - optional; type *nodal\_support\_diagram\_along\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_y_is_sorted` - optional; type *boolean*
- `diagram_along_y_start` - optional; type *nodal\_support\_diagram\_along\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_y_symmetric` - optional; type *boolean*
- `diagram_along_y_table` - optional; type *array\_of\_nodal\_support\_diagram\_along\_y\_table*
  - `nodal_support_diagram_along_y_table` - optional, unbounded; type *nodal\_support\_diagram\_along\_y\_table*
    - `no` - optional; type *int*
    - `displacement` - optional; type *double*
    - `force` - optional; type *double*
    - `spring` - optional; type *double*
    - `note` - optional; type *string*
- `diagram_along_z_end` - optional; type *nodal\_support\_diagram\_along\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_z_is_sorted` - optional; type *boolean*
- `diagram_along_z_start` - optional; type *nodal\_support\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_z_symmetric` - optional; type *boolean*
- `diagram_along_z_table` - optional; type *array\_of\_nodal\_support\_diagram\_along\_z\_table*
  - `nodal_support_diagram_along_z_table` - optional, unbounded; type *nodal\_support\_diagram\_along\_z\_table*
    - `no` - optional; type *int*
    - `displacement` - optional; type *double*
    - `force` - optional; type *double*
    - `spring` - optional; type *double*
    - `note` - optional; type *string*
- `diagram_around_x_end` - optional; type *nodal\_support\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_around_x_is_sorted` - optional; type *boolean*
- `diagram_around_x_start` - optional; type *nodal\_support\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_around_x_symmetric` - optional; type *boolean*
- `diagram_around_x_table` - optional; type *array\_of\_nodal\_support\_diagram\_around\_x\_table*
  - `nodal_support_diagram_around_x_table` - optional, unbounded; type *nodal\_support\_diagram\_around\_x\_table*
    - `no` - optional; type *int*
    - `rotation` - optional; type *double*
    - `moment` - optional; type *double*
    - `spring` - optional; type *double*
    - `note` - optional; type *string*
- `diagram_around_y_end` - optional; type *nodal\_support\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_around_y_is_sorted` - optional; type *boolean*
- `diagram_around_y_start` - optional; type *nodal\_support\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_around_y_symmetric` - optional; type *boolean*
- `diagram_around_y_table` - optional; type *array\_of\_nodal\_support\_diagram\_around\_y\_table*
  - `nodal_support_diagram_around_y_table` - optional, unbounded; type *nodal\_support\_diagram\_around\_y\_table*
    - `no` - optional; type *int*
    - `rotation` - optional; type *double*
    - `moment` - optional; type *double*
    - `spring` - optional; type *double*
    - `note` - optional; type *string*
- `diagram_around_z_end` - optional; type *nodal\_support\_diagram\_around\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_around_z_is_sorted` - optional; type *boolean*
- `diagram_around_z_start` - optional; type *nodal\_support\_diagram\_around\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }

'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }

- diagram\_around\_z\_symmetric - optional; type *boolean*
- diagram\_around\_z\_table - optional; type *array\_of\_nodal\_support\_diagram\_around\_z\_table*
  - nodal\_support\_diagram\_around\_z\_table - optional, unbounded; type *nodal\_support\_diagram\_around\_z\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- directed\_to\_node\_direction\_node - optional; type *int*
- directed\_to\_node\_first\_axis - optional; type *nodal\_support\_directed\_to\_node\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- directed\_to\_node\_plane\_node - optional; type *int*
- directed\_to\_node\_second\_axis - optional; type *nodal\_support\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- fictitious\_column\_enabled - optional; type *boolean*
- friction\_coefficient\_x - optional; type *double*
- friction\_coefficient\_xy - optional; type *double*
- friction\_coefficient\_xz - optional; type *double*
- friction\_coefficient\_y - optional; type *double*
- friction\_coefficient\_yx - optional; type *double*
- friction\_coefficient\_yz - optional; type *double*
- friction\_coefficient\_z - optional; type *double*
- friction\_coefficient\_zx - optional; type *double*
- friction\_coefficient\_zy - optional; type *double*
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *nodal\_support\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_axis - optional; type *nodal\_support\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- partial\_activity\_along\_x\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_x\_negative\_force - optional; type *double*
- partial\_activity\_along\_x\_negative\_slippage - optional; type *double*
- partial\_activity\_along\_x\_negative\_type - optional; type *nodal\_support\_partial\_activity\_along\_x\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_x\_positive\_displacement - optional; type *double*
- partial\_activity\_along\_x\_positive\_force - optional; type *double*
- partial\_activity\_along\_x\_positive\_slippage - optional; type *double*
- partial\_activity\_along\_x\_positive\_type - optional; type *nodal\_support\_partial\_activity\_along\_x\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_y\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_y\_negative\_force - optional; type *double*
- partial\_activity\_along\_y\_negative\_slippage - optional; type *double*
- partial\_activity\_along\_y\_negative\_type - optional; type *nodal\_support\_partial\_activity\_along\_y\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_y\_positive\_displacement - optional; type *double*
- partial\_activity\_along\_y\_positive\_force - optional; type *double*
- partial\_activity\_along\_y\_positive\_slippage - optional; type *double*
- partial\_activity\_along\_y\_positive\_type - optional; type *nodal\_support\_partial\_activity\_along\_y\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_z\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_z\_negative\_force - optional; type *double*
- partial\_activity\_along\_z\_negative\_slippage - optional; type *double*
- partial\_activity\_along\_z\_negative\_type - optional; type *nodal\_support\_partial\_activity\_along\_z\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_along\_z\_positive\_displacement - optional; type *double*
- partial\_activity\_along\_z\_positive\_force - optional; type *double*
- partial\_activity\_along\_z\_positive\_slippage - optional; type *double*
- partial\_activity\_along\_z\_positive\_type - optional; type *nodal\_support\_partial\_activity\_along\_z\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_around\_x\_negative\_moment - optional; type *double*
- partial\_activity\_around\_x\_negative\_rotation - optional; type *double*
- partial\_activity\_around\_x\_negative\_slippage - optional; type *double*
- partial\_activity\_around\_x\_negative\_type - optional; type *nodal\_support\_partial\_activity\_around\_x\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_around\_x\_positive\_moment - optional; type *double*
- partial\_activity\_around\_x\_positive\_rotation - optional; type *double*
- partial\_activity\_around\_x\_positive\_slippage - optional; type *double*
- partial\_activity\_around\_x\_positive\_type - optional; type *nodal\_support\_partial\_activity\_around\_x\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_around\_y\_negative\_moment - optional; type *double*
- partial\_activity\_around\_y\_negative\_rotation - optional; type *double*
- partial\_activity\_around\_y\_negative\_slippage - optional; type *double*
- partial\_activity\_around\_y\_negative\_type - optional; type *nodal\_support\_partial\_activity\_around\_y\_negative\_type* - type *undefined* with restriction - enum {

'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }

- partial\_activity\_around\_y\_positive\_moment - optional; type *double*
- partial\_activity\_around\_y\_positive\_rotation - optional; type *double*
- partial\_activity\_around\_y\_positive\_slippage - optional; type *double*
- partial\_activity\_around\_y\_positive\_type - optional; type *nodal\_support\_partial\_activity\_around\_y\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_around\_z\_negative\_moment - optional; type *double*
- partial\_activity\_around\_z\_negative\_rotation - optional; type *double*
- partial\_activity\_around\_z\_negative\_slippage - optional; type *double*
- partial\_activity\_around\_z\_negative\_type - optional; type *nodal\_support\_partial\_activity\_around\_z\_negative\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- partial\_activity\_around\_z\_positive\_moment - optional; type *double*
- partial\_activity\_around\_z\_positive\_rotation - optional; type *double*
- partial\_activity\_around\_z\_positive\_slippage - optional; type *double*
- partial\_activity\_around\_z\_positive\_type - optional; type *nodal\_support\_partial\_activity\_around\_z\_positive\_type* - type *undefined* with restriction - enum { 'PARTIAL\_ACTIVITY\_TYPE\_COMPLETE', 'PARTIAL\_ACTIVITY\_TYPE\_FAILURE', 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- rotated\_about\_angle\_1 - optional; type *double*
- rotated\_about\_angle\_2 - optional; type *double*
- rotated\_about\_angle\_3 - optional; type *double*
- rotated\_about\_angle\_x - optional; type *double*
- rotated\_about\_angle\_y - optional; type *double*
- rotated\_about\_angle\_z - optional; type *double*
- specific\_direction\_enabled - optional; type *boolean*
- specific\_direction\_type - optional; type *nodal\_support\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
- stiffness\_diagram\_around\_x\_depends\_on - optional; type *nodal\_support\_stiffness\_diagram\_around\_x\_depends\_on* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_P', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PX', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PY', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PZ' }
- stiffness\_diagram\_around\_x\_end - optional; type *nodal\_support\_stiffness\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_x\_is\_sorted - optional; type *boolean*
- stiffness\_diagram\_around\_x\_start - optional; type *nodal\_support\_stiffness\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_x\_symmetric - optional; type *boolean*
- stiffness\_diagram\_around\_x\_table - optional; type *array\_of\_nodal\_support\_stiffness\_diagram\_around\_x\_table*
  - nodal\_support\_stiffness\_diagram\_around\_x\_table - optional, unbounded; type *nodal\_support\_stiffness\_diagram\_around\_x\_table*
    - no - optional; type *int*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- stiffness\_diagram\_around\_y\_depends\_on - optional; type *nodal\_support\_stiffness\_diagram\_around\_y\_depends\_on* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_P', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PX', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PY', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PZ' }
- stiffness\_diagram\_around\_y\_end - optional; type *nodal\_support\_stiffness\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_y\_is\_sorted - optional; type *boolean*
- stiffness\_diagram\_around\_y\_start - optional; type *nodal\_support\_stiffness\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_y\_symmetric - optional; type *boolean*
- stiffness\_diagram\_around\_y\_table - optional; type *array\_of\_nodal\_support\_stiffness\_diagram\_around\_y\_table*
  - nodal\_support\_stiffness\_diagram\_around\_y\_table - optional, unbounded; type *nodal\_support\_stiffness\_diagram\_around\_y\_table*
    - no - optional; type *int*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- stiffness\_diagram\_around\_z\_depends\_on - optional; type *nodal\_support\_stiffness\_diagram\_around\_z\_depends\_on* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_P', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PX', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PY', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PZ' }
- stiffness\_diagram\_around\_z\_end - optional; type *nodal\_support\_stiffness\_diagram\_around\_z\_end* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_z\_is\_sorted - optional; type *boolean*
- stiffness\_diagram\_around\_z\_start - optional; type *nodal\_support\_stiffness\_diagram\_around\_z\_start* - type *undefined* with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- stiffness\_diagram\_around\_z\_symmetric - optional; type *boolean*
- stiffness\_diagram\_around\_z\_table - optional; type *array\_of\_nodal\_support\_stiffness\_diagram\_around\_z\_table*
  - nodal\_support\_stiffness\_diagram\_around\_z\_table - optional, unbounded; type *nodal\_support\_stiffness\_diagram\_around\_z\_table*
    - no - optional; type *int*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- support\_dimension\_diameter\_x - optional; type *double*
- support\_dimension\_diameter\_y - optional; type *double*
- support\_dimension\_diameter\_z - optional; type *double*
- support\_dimension\_height\_x - optional; type *double*
- support\_dimension\_height\_y - optional; type *double*
- support\_dimension\_height\_z - optional; type *double*

- support\_dimension\_type\_on\_x - optional; type *nodal\_support\_support\_dimension\_type\_on\_x* - type *undefined* with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- support\_dimension\_type\_on\_y - optional; type *nodal\_support\_support\_dimension\_type\_on\_y* - type *undefined* with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- support\_dimension\_type\_on\_z - optional; type *nodal\_support\_support\_dimension\_type\_on\_z* - type *undefined* with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- support\_dimension\_width\_x - optional; type *double*
- support\_dimension\_width\_y - optional; type *double*
- support\_dimension\_width\_z - optional; type *double*
- support\_dimensions\_enabled - optional; type *boolean*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 80. get\_node

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_node

**Input:** get\_node\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_node*

- no type *int*

**Output:** get\_node\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_nodeResponse*

- value type *node*
  - no type *int*
  - type - optional; type *node\_type* - type *undefined* with restriction - enum { 'TYPE\_BETWEEN\_TWO\_NODES', 'TYPE\_BETWEEN\_TWO\_POINTS', 'TYPE\_ON\_LINE', 'TYPE\_ON\_MEMBER', 'TYPE\_STANDARD' }
  - reference\_node - optional; type *int*
  - coordinate\_system - optional; type *int*
  - coordinate\_system\_type - optional; type *node\_coordinate\_system\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_CARTESIAN', 'COORDINATE\_SYSTEM\_POLAR', 'COORDINATE\_SYSTEM\_X\_CYLINDRICAL', 'COORDINATE\_SYSTEM\_Y\_CYLINDRICAL', 'COORDINATE\_SYSTEM\_Z\_CYLINDRICAL' }
  - coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - coordinate\_1 - optional; type *double*
  - coordinate\_2 - optional; type *double*
  - coordinate\_3 - optional; type *double*
  - global\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - global\_coordinate\_1 - optional; type *double*
  - global\_coordinate\_2 - optional; type *double*
  - global\_coordinate\_3 - optional; type *double*
  - between\_two\_nodes\_start\_node - optional; type *int*
  - between\_two\_points\_start\_point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - between\_two\_points\_start\_point\_coordinate\_1 - optional; type *double*
  - between\_two\_points\_start\_point\_coordinate\_2 - optional; type *double*
  - between\_two\_points\_start\_point\_coordinate\_3 - optional; type *double*
  - between\_two\_nodes\_end\_node - optional; type *int*
  - between\_two\_points\_end\_point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - between\_two\_points\_end\_point\_coordinate\_1 - optional; type *double*
  - between\_two\_points\_end\_point\_coordinate\_2 - optional; type *double*
  - between\_two\_points\_end\_point\_coordinate\_3 - optional; type *double*
  - on\_line\_reference\_line - optional; type *int*
  - on\_member\_reference\_member - optional; type *int*
  - reference\_type - optional; type *node\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - reference\_object\_projected\_length - optional; type *double*
  - distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_from\_start\_relative - optional; type *double*
  - distance\_from\_start\_absolute - optional; type *double*
  - offset\_in\_local\_direction\_y - optional; type *double*
  - offset\_in\_local\_direction\_z - optional; type *double*
  - offset\_in\_global\_direction\_x - optional; type *double*
  - offset\_in\_global\_direction\_y - optional; type *double*
  - offset\_in\_global\_direction\_z - optional; type *double*
  - support - optional; type *int*
  - mesh\_refinement - optional; type *int*
  - concrete\_design\_ultimate\_configuration - optional; type *int*
  - concrete\_design\_fire\_configuration - optional; type *int*
  - punching\_design - optional; type *boolean*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 81. get\_note

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_note

**Input:** get\_note\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_note*

- no type *int*

**Output:** get\_note\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_noteResponse*

- value type *note*
  - no type *int*
  - type - optional; type *note\_type* - type *undefined* with restriction - enum { 'NOTE\_TYPE\_LINE', 'NOTE\_TYPE\_MEMBER', 'NOTE\_TYPE\_NODE', 'NOTE\_TYPE\_POINT', 'NOTE\_TYPE\_SURFACE' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - text - optional; type *string*
  - point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - point\_coordinate\_x - optional; type *double*
  - point\_coordinate\_y - optional; type *double*
  - point\_coordinate\_z - optional; type *double*
  - node - optional; type *int*
  - member - optional; type *int*
  - member\_reference\_type - optional; type *note\_member\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - member\_length - optional; type *double*
  - member\_distance\_is\_defined\_as\_relative - optional; type *boolean*
  - member\_distance\_relative - optional; type *double*
  - member\_distance\_absolute - optional; type *double*
  - line - optional; type *int*
  - line\_length - optional; type *double*
  - surface - optional; type *int*
  - surface\_reference\_type - optional; type *note\_surface\_reference\_type* - type *undefined* with restriction - enum { 'OFFSET\_TYPE\_XY', 'OFFSET\_TYPE\_XZ', 'OFFSET\_TYPE\_YZ' }
  - surface\_first\_coordinate - optional; type *double*
  - surface\_second\_coordinate - optional; type *double*
  - offset - optional; type *boolean*
  - offset\_type - optional; type *note\_offset\_type* - type *undefined* with restriction - enum { 'OFFSET\_TYPE\_XY', 'OFFSET\_TYPE\_XYZ', 'OFFSET\_TYPE\_XZ', 'OFFSET\_TYPE\_YZ' }
  - offset\_coordinate - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - offset\_coordinate\_x - optional; type *double*
  - offset\_coordinate\_y - optional; type *double*
  - offset\_coordinate\_z - optional; type *double*
  - rotation - optional; type *double*
  - show\_comment - optional; type *boolean*
  - display\_properties\_index - optional; type *int*
  - comment - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 82. get\_nth\_object\_number

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_nth\_object\_number

**Input:** get\_nth\_object\_number\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_nth\_object\_number*

- type type *object\_types* - type *undefined* with restriction - enum { 'E\_OBJECT\_TYPE\_ACTION', 'E\_OBJECT\_TYPE\_ACTION\_COMBINATION', 'E\_OBJECT\_TYPE\_BUILDING\_STORY', 'E\_OBJECT\_TYPE\_CLIPPING\_BOX', 'E\_OBJECT\_TYPE\_CLIPPING\_PLANE', 'E\_OBJECT\_TYPE\_COMBINATION\_WIZARD', 'E\_OBJECT\_TYPE\_COORDINATE\_SYSTEM', 'E\_OBJECT\_TYPE\_CUTTING\_LINE\_SETTING', 'E\_OBJECT\_TYPE\_CUTTING\_PATTERN', 'E\_OBJECT\_TYPE\_DESIGN\_SITUATION', 'E\_OBJECT\_TYPE\_DESIGN\_SUPPORT', 'E\_OBJECT\_TYPE\_DIMENSION', 'E\_OBJECT\_TYPE\_FREE\_CIRCULAR\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_CONCENTRATED\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_POLYGON\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_RECTANGULAR\_LOAD', 'E\_OBJECT\_TYPE\_GLOBAL\_PARAMETER', 'E\_OBJECT\_TYPE\_IMPERFECTION\_CASE', 'E\_OBJECT\_TYPE\_IMPOSED\_LINE\_DEFORMATION', 'E\_OBJECT\_TYPE\_IMPOSED\_NODAL\_DEFORMATION', 'E\_OBJECT\_TYPE\_INTERSECTION', 'E\_OBJECT\_TYPE\_LINE', 'E\_OBJECT\_TYPE\_LINE\_GRID', 'E\_OBJECT\_TYPE\_LINE\_HINGE', 'E\_OBJECT\_TYPE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_LINE\_SET', 'E\_OBJECT\_TYPE\_LINE\_SET\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_SUPPORT', 'E\_OBJECT\_TYPE\_LINE\_WELDED\_JOINT', 'E\_OBJECT\_TYPE\_LOAD\_CASE', 'E\_OBJECT\_TYPE\_LOAD\_COMBINATION', 'E\_OBJECT\_TYPE\_MATERIAL', 'E\_OBJECT\_TYPE\_MEMBER', 'E\_OBJECT\_TYPE\_MEMBER\_DEFINABLE\_STIFFNESS', 'E\_OBJECT\_TYPE\_MEMBER\_ECCENTRICITY', 'E\_OBJECT\_TYPE\_MEMBER\_HINGE', 'E\_OBJECT\_TYPE\_MEMBER\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_NONLINEARITY', 'E\_OBJECT\_TYPE\_MEMBER\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_RESULT\_INTERMEDIATE\_POINT', 'E\_OBJECT\_TYPE\_MEMBER\_SET', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_STIFFNESS\_MODIFICATION', 'E\_OBJECT\_TYPE\_MEMBER\_SUPPORT', 'E\_OBJECT\_TYPE\_MEMBER\_TRANSVERSE\_STIFFENER', 'E\_OBJECT\_TYPE\_NODAL\_LOAD', 'E\_OBJECT\_TYPE\_NODAL\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_NODAL\_SUPPORT', 'E\_OBJECT\_TYPE\_NODE', 'E\_OBJECT\_TYPE\_NOTE', 'E\_OBJECT\_TYPE\_OBJECT\_SNAP', 'E\_OBJECT\_TYPE\_OPENING', 'E\_OBJECT\_TYPE\_OPENING\_LOAD', 'E\_OBJECT\_TYPE\_RESULT\_COMBINATION', 'E\_OBJECT\_TYPE\_RESULT\_SECTION', 'E\_OBJECT\_TYPE\_RIGID\_LINK', 'E\_OBJECT\_TYPE\_SECTION', 'E\_OBJECT\_TYPE\_SOIL\_MASSIF', 'E\_OBJECT\_TYPE\_SOIL\_SAMPLE', 'E\_OBJECT\_TYPE\_SOLID', 'E\_OBJECT\_TYPE\_SOLID\_CONTACTS', 'E\_OBJECT\_TYPE\_SOLID\_GAS', 'E\_OBJECT\_TYPE\_SOLID\_LOAD', 'E\_OBJECT\_TYPE\_SOLID\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_SOLID\_SET', 'E\_OBJECT\_TYPE\_SOLID\_SET\_LOAD', 'E\_OBJECT\_TYPE\_SPECTRAL\_ANALYSIS\_SETTINGS' }



```
'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION',
'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT',
'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY',
'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD',
'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT',
'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION',
'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS',
'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION',
'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
▪ order type int
▪ parent_no - optional; type int
```

**Output:** get\_nth\_object\_number\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_nth_object_numberResponse
▪ value type int
```

### 83. get\_object\_count

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_object\_count

**Input:** get\_object\_count\_request (soap:body, use = literal) [Source code](#)

```
parameters type get_object_count
▪ type type object_types - type undefined with restriction - enum { 'E_OBJECT_TYPE_ACTION',
'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX',
'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD',
'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING',
'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION',
'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD',
'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD',
'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD',
'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE',
'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION',
'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID',
'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT',
'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT',
'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION',
'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS',
'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE',
'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD',
'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE',
'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET',
'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD',
'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER',
'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT',
'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE',
'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD',
'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK',
'E_OBJECT_TYPE_SECTION', 'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE',
'E_OBJECT_TYPE_SOLID', 'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS',
'E_OBJECT_TYPE_SOLID_LOAD', 'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET',
'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS',
'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION',
'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT',
'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY',
'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD',
'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT',
'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION',
'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS',
'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION',
'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
▪ parent_no - optional; type int
```

**Output:** get\_object\_count\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_object_countResponse
▪ value type int
```

### 84. get\_object\_information

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_object\_information

**Input:** get\_object\_information\_request (soap:body, use = literal) [Source code](#)

```
parameters type get_object_information
▪ type type object_types - type undefined with restriction - enum { 'E_OBJECT_TYPE_ACTION',
'E_OBJECT_TYPE_ACTION_COMBINATION', 'E_OBJECT_TYPE_BUILDING_STORY', 'E_OBJECT_TYPE_CLIPPING_BOX',
'E_OBJECT_TYPE_CLIPPING_PLANE', 'E_OBJECT_TYPE_COMBINATION_WIZARD',
'E_OBJECT_TYPE_COORDINATE_SYSTEM', 'E_OBJECT_TYPE_CUTTING_LINE_SETTING',
'E_OBJECT_TYPE_CUTTING_PATTERN', 'E_OBJECT_TYPE_DESIGN_SITUATION',
'E_OBJECT_TYPE_DESIGN_SUPPORT', 'E_OBJECT_TYPE_DIMENSION', 'E_OBJECT_TYPE_FREE_CIRCULAR_LOAD',
'E_OBJECT_TYPE_FREE_CONCENTRATED_LOAD', 'E_OBJECT_TYPE_FREE_LINE_LOAD',
'E_OBJECT_TYPE_FREE_POLYGON_LOAD', 'E_OBJECT_TYPE_FREE_RECTANGULAR_LOAD',
'E_OBJECT_TYPE_GLOBAL_PARAMETER', 'E_OBJECT_TYPE_IMPERFECTION_CASE',
'E_OBJECT_TYPE_IMPOSED_LINE_DEFORMATION', 'E_OBJECT_TYPE_IMPOSED_NODAL_DEFORMATION',
'E_OBJECT_TYPE_INTERSECTION', 'E_OBJECT_TYPE_LINE', 'E_OBJECT_TYPE_LINE_GRID',
'E_OBJECT_TYPE_LINE_HINGE', 'E_OBJECT_TYPE_LINE_LOAD', 'E_OBJECT_TYPE_LINE_MESH_REFINEMENT',
'E_OBJECT_TYPE_LINE_SET', 'E_OBJECT_TYPE_LINE_SET_LOAD', 'E_OBJECT_TYPE_LINE_SUPPORT',
'E_OBJECT_TYPE_LINE_WELDED_JOINT', 'E_OBJECT_TYPE_LOAD_CASE', 'E_OBJECT_TYPE_LOAD_COMBINATION',
'E_OBJECT_TYPE_MATERIAL', 'E_OBJECT_TYPE_MEMBER', 'E_OBJECT_TYPE_MEMBER_DEFINABLE_STIFFNESS',
'E_OBJECT_TYPE_MEMBER_ECCENTRICITY', 'E_OBJECT_TYPE_MEMBER_HINGE',
'E_OBJECT_TYPE_MEMBER_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_LOAD',
```

```
'E_OBJECT_TYPE_MEMBER_NONLINEARITY', 'E_OBJECT_TYPE_MEMBER_REPRESENTATIVE',
'E_OBJECT_TYPE_MEMBER_RESULT_INTERMEDIATE_POINT', 'E_OBJECT_TYPE_MEMBER_SET',
'E_OBJECT_TYPE_MEMBER_SET_IMPERFECTION', 'E_OBJECT_TYPE_MEMBER_SET_LOAD',
'E_OBJECT_TYPE_MEMBER_SET_REPRESENTATIVE', 'E_OBJECT_TYPE_MEMBER_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_MEMBER_SUPPORT', 'E_OBJECT_TYPE_MEMBER_TRANSVERSE_STIFFENER',
'E_OBJECT_TYPE_NODAL_LOAD', 'E_OBJECT_TYPE_NODAL_MESH_REFINEMENT',
'E_OBJECT_TYPE_NODAL_SUPPORT', 'E_OBJECT_TYPE_NODE', 'E_OBJECT_TYPE_NOTE',
'E_OBJECT_TYPE_OBJECT_SNAP', 'E_OBJECT_TYPE_OPENING', 'E_OBJECT_TYPE_OPENING_LOAD',
'E_OBJECT_TYPE_RESULT_COMBINATION', 'E_OBJECT_TYPE_RESULT_SECTION', 'E_OBJECT_TYPE_RIGID_LINK',
'E_OBJECT_TYPE_SECTION', 'E_OBJECT_TYPE_SOIL_MASSIF', 'E_OBJECT_TYPE_SOIL_SAMPLE',
'E_OBJECT_TYPE_SOLID', 'E_OBJECT_TYPE_SOLID_CONTACTS', 'E_OBJECT_TYPE_SOLID_GAS',
'E_OBJECT_TYPE_SOLID_LOAD', 'E_OBJECT_TYPE_SOLID_MESH_REFINEMENT', 'E_OBJECT_TYPE_SOLID_SET',
'E_OBJECT_TYPE_SOLID_SET_LOAD', 'E_OBJECT_TYPE_SPECTRAL_ANALYSIS_SETTINGS',
'E_OBJECT_TYPE_STATIC_ANALYSIS_SETTINGS', 'E_OBJECT_TYPE_STRUCTURE_MODIFICATION',
'E_OBJECT_TYPE_SURFACE', 'E_OBJECT_TYPE_SURFACES_CONTACT',
'E_OBJECT_TYPE_SURFACES_CONTACT_TYPE', 'E_OBJECT_TYPE_SURFACE_ECCENTRICITY',
'E_OBJECT_TYPE_SURFACE_IMPERFECTION', 'E_OBJECT_TYPE_SURFACE_LOAD',
'E_OBJECT_TYPE_SURFACE_MESH_REFINEMENT', 'E_OBJECT_TYPE_SURFACE_RESULTS_ADJUSTMENT',
'E_OBJECT_TYPE_SURFACE_SET', 'E_OBJECT_TYPE_SURFACE_SET_IMPERFECTION',
'E_OBJECT_TYPE_SURFACE_SET_LOAD', 'E_OBJECT_TYPE_SURFACE_STIFFNESS_MODIFICATION',
'E_OBJECT_TYPE_SURFACE_SUPPORT', 'E_OBJECT_TYPE_TERRAIN', 'E_OBJECT_TYPE_THICKNESS',
'E_OBJECT_TYPE_VISUAL_OBJECT', 'E_OBJECT_TYPE_WIND_PROFILE', 'E_OBJECT_TYPE_WIND_SIMULATION',
'E_OBJECT_TYPE_WIND_SIMULATION_ANALYSIS_SETTINGS' }
```

**Output:** `get_object_information_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_object_informationResponse
 ■ value type object_information
 ■ singular_name type string
 ■ plural_name type string
 ■ attributes type attributes
 ■ attribute_information - optional, unbounded; type attribute
 ■ id - optional; type string
 ■ name - optional; type string
 ■ symbol - optional; type string
 ■ plane_symbol - optional; type string
 ■ table_column - optional; type table_column
 ■ index type int
 ■ header_text type string
 ■ header_top_text type string
 ■ header_merged_cells type int
 ■ width type int
 ■ horizontal_alignment type horizontal_alignment - type undefined with restriction - enum {
 'ALIGN_ABSOLUTE', 'ALIGN_CENTER', 'ALIGN_JUSTIFY', 'ALIGN_LEFT', 'ALIGN_RIGHT',
 'ALIGN_UNKNOWN' }
 ■ vertical_alignment type vertical_alignment - type undefined with restriction - enum {
 'ALIGN_BASELINE', 'ALIGN_BOTTOM', 'ALIGN_CENTER', 'ALIGN_TOP', 'ALIGN_UNKNOWN' }
```

## 85. `get_object_snap`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_object_snap`

**Input:** `get_object_snap_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_object_snap
 ■ no type int
```

**Output:** `get_object_snap_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_object_snapResponse
 ■ value type object_snap
 ■ no type int
 ■ type - optional; type object_snap_type - type undefined with restriction - enum { 'TYPE_STANDARD' }
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ snap_nodes - optional; type boolean
 ■ snap_centers_and_focuses - optional; type boolean
 ■ snap_intersections - optional; type boolean
 ■ snap_perpendicular - optional; type boolean
 ■ snap_extend - optional; type boolean
 ■ snap_parallel - optional; type boolean
 ■ snap_tangent - optional; type boolean
 ■ snap_circle_quadrants - optional; type boolean
 ■ snap_shapes - optional; type boolean
 ■ snap_parts - optional; type boolean
 ■ snap_absolute_distance - optional; type boolean
 ■ snap_relative_distance - optional; type boolean
 ■ snap_snappable_points_only - optional; type boolean
 ■ snap_guidelines - optional; type boolean
 ■ snap_background_layers - optional; type boolean
 ■ snap_line_grids - optional; type boolean
 ■ comment - optional; type string
 ■ parts_count - optional; type int
 ■ absolute_distance - optional; type double
 ■ relative_distance - optional; type double
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

## 86. `get_opening`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_opening`

**Input:** `get_opening_request` (soap:body, use = literal) [Source code](#)

parameters type *get\_opening*

- no type *int*

Output: *get\_opening\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_openingResponse*

- value type *opening*
  - no type *int*
  - area - optional; type *double*
  - boundary\_lines - optional; type *array\_of\_int*
  - center\_of\_opening - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_of\_opening\_x - optional; type *double*
  - center\_of\_opening\_y - optional; type *double*
  - center\_of\_opening\_z - optional; type *double*
  - position\_full\_description - optional; type *string*
  - position\_short\_description - optional; type *string*
  - surfaces - optional; type *array\_of\_int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 87. *get\_opening\_load*

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: [http://localhost:8082/get\\_opening\\_load](http://localhost:8082/get_opening_load)

Input: *get\_opening\_load\_request* (soap:body, use = literal) [Source code](#)

parameters type *get\_opening\_load*

- no type *int*
- load\_case\_no type *int*

Output: *get\_opening\_load\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_opening\_loadResponse*

- value type *opening\_load*
  - no type *int*
  - load\_type - optional; type *opening\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_FORCE' }
  - openings - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *string*
  - load\_direction - optional; type *opening\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_Z' }
  - load\_distribution - optional; type *opening\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM\_TRAPEZOIDAL' }
  - magnitude - optional; type *double*
  - magnitude\_1 - optional; type *double*
  - magnitude\_2 - optional; type *double*
  - magnitude\_3 - optional; type *double*
  - node\_1 - optional; type *int*
  - node\_2 - optional; type *int*
  - node\_3 - optional; type *int*
  - smooth\_punctual\_load\_enabled - optional; type *boolean*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 88. *get\_optimization\_settings*

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: [http://localhost:8082/get\\_optimization\\_settings](http://localhost:8082/get_optimization_settings)

Input: *get\_optimization\_settings\_request* (soap:body, use = literal) [Source code](#)

parameters type *get\_optimization\_settings*

Output: *get\_optimization\_settings\_response* (soap:body, use = literal) [Source code](#)

parameters type *get\_optimization\_settingsResponse*

- value type *optimizationSettingsConfig\_type*
  - general\_optimization\_active type *boolean*
  - general\_keep\_best\_number\_model\_mutations type *int*
  - general\_optimize\_on type *optimizationSettingsConfig\_general\_optimize\_on\_type* - type *undefined* with restriction - enum { 'E\_OPTIMIZE\_ON\_TYPE\_MAX\_GLOBAL\_PARAMETER', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_CO2\_EMISSIONS', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_COST', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_GLOBAL\_PARAMETER', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_MEMBER\_DEFORMATION', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_NODAL\_DEFORMATION', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_SURFACE\_DEFORMATION', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_VECTORIAL\_DISPLACEMENT', 'E\_OPTIMIZE\_ON\_TYPE\_MIN\_WHOLE\_WEIGHT', 'E\_OPTIMIZE\_ON\_TYPE\_BEGIN', 'E\_OPTIMIZE\_ON\_TYPE\_END' }
  - general\_optimizer type *optimizationSettingsConfig\_general\_optimizer\_type* - type *undefined* with restriction - enum { 'E\_OPTIMIZER\_TYPE\_ALL\_MUTATIONS', 'E\_OPTIMIZER\_TYPE\_PARTICLE\_SWARM',

```
'E_OPTIMIZER_TYPE_PERCENTS_OF_RANDOM_MUTATIONS', 'E_OPTIMIZER_TYPE__BEGIN',
'E_OPTIMIZER_TYPE__END' }
▪ general_number_random_mutations type double
```

#### 89. get\_parts\_list\_all\_by\_material

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_parts\_list\_all\_by\_material

**Input:** get\_parts\_list\_all\_by\_material\_request (soap:body, use = literal) [Source code](#)

parameters type [get\\_parts\\_list\\_all\\_by\\_material](#)

**Output:** get\_parts\_list\_all\_by\_material\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_all_by_materialResponse
▪ value type array_of_parts_list_all_by_material
▪ parts_list_all_by_material - optional, unbounded; type parts_list_all_by_material
 ▪ no - optional; type int
 ▪ material_name - optional; type string
 ▪ object_type - optional; type string
 ▪ total_coating - optional; type double
 ▪ volume - optional; type double
 ▪ weight - optional; type double
```

#### 90. get\_parts\_list\_member\_representatives\_by\_material

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_parts\_list\_member\_representatives\_by\_material

**Input:** get\_parts\_list\_member\_representatives\_by\_material\_request (soap:body, use = literal) [Source code](#)

parameters type [get\\_parts\\_list\\_member\\_representatives\\_by\\_material](#)

**Output:** get\_parts\_list\_member\_representatives\_by\_material\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_member_representatives_by_materialResponse
▪ value type array_of_parts_list_member_representatives_by_material
▪ parts_list_member_representatives_by_material - optional, unbounded; type parts_list_member_representatives_by_material
 ▪ no - optional; type int
 ▪ material_name - optional; type string
 ▪ section_name - optional; type string
 ▪ member_representative_no - optional; type string
 ▪ quantity - optional; type double
 ▪ length - optional; type double
 ▪ unit_surface_area - optional; type double
 ▪ volume - optional; type double
 ▪ unit_weight - optional; type double
 ▪ member_weight - optional; type double
 ▪ total_length - optional; type double
 ▪ total_surface_area - optional; type double
 ▪ total_volume - optional; type double
 ▪ total_weight - optional; type double
```

#### 91. get\_parts\_list\_member\_set\_representatives\_by\_material

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_parts\_list\_member\_set\_representatives\_by\_material

**Input:** get\_parts\_list\_member\_set\_representatives\_by\_material\_request (soap:body, use = literal) [Source code](#)

parameters type [get\\_parts\\_list\\_member\\_set\\_representatives\\_by\\_material](#)

**Output:** get\_parts\_list\_member\_set\_representatives\_by\_material\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_member_set_representatives_by_materialResponse
▪ value type array_of_parts_list_member_set_representatives_by_material
▪ parts_list_member_set_representatives_by_material - optional, unbounded; type parts_list_member_set_representatives_by_material
 ▪ no - optional; type int
 ▪ material_name - optional; type string
 ▪ section_name - optional; type string
 ▪ member_set_representative_no - optional; type string
 ▪ quantity - optional; type double
 ▪ length - optional; type double
 ▪ unit_surface_area - optional; type double
 ▪ volume - optional; type double
 ▪ unit_weight - optional; type double
 ▪ member_weight - optional; type double
 ▪ total_length - optional; type double
 ▪ total_surface_area - optional; type double
 ▪ total_volume - optional; type double
 ▪ total_weight - optional; type double
```

#### 92. get\_parts\_list\_member\_sets\_by\_material

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_parts\_list\_member\_sets\_by\_material

**Input:** get\_parts\_list\_member\_sets\_by\_material\_request (soap:body, use = literal) [Source code](#)

parameters type [get\\_parts\\_list\\_member\\_sets\\_by\\_material](#)

**Output:** `get_parts_list_member_sets_by_material_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_member_sets_by_materialResponse
 value type array_of_parts_list_member_sets_by_material
 parts_list_member_sets_by_material - optional, unbounded; type parts_list_member_sets_by_material
 no - optional; type int
 material_name - optional; type string
 section_name - optional; type string
 member_sets_no - optional; type string
 quantity - optional; type double
 length - optional; type double
 unit_surface_area - optional; type double
 volume - optional; type double
 unit_weight - optional; type double
 member_weight - optional; type double
 total_length - optional; type double
 total_surface_area - optional; type double
 total_volume - optional; type double
 total_weight - optional; type double
```

### 93. `get_parts_list_members_by_material`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_parts_list_members_by_material`

**Input:** `get_parts_list_members_by_material_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_members_by_material
```

**Output:** `get_parts_list_members_by_material_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_members_by_materialResponse
 value type array_of_parts_list_members_by_material
 parts_list_members_by_material - optional, unbounded; type parts_list_members_by_material
 no - optional; type int
 material_name - optional; type string
 section_name - optional; type string
 members_no - optional; type string
 quantity - optional; type double
 length - optional; type double
 unit_surface_area - optional; type double
 volume - optional; type double
 unit_weight - optional; type double
 member_weight - optional; type double
 total_length - optional; type double
 total_surface_area - optional; type double
 total_volume - optional; type double
 total_weight - optional; type double
```

### 94. `get_parts_list_solids_by_material`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_parts_list_solids_by_material`

**Input:** `get_parts_list_solids_by_material_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_solids_by_material
```

**Output:** `get_parts_list_solids_by_material_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_solids_by_materialResponse
 value type array_of_parts_list_solids_by_material
 parts_list_solids_by_material - optional, unbounded; type parts_list_solids_by_material
 no - optional; type int
 material_name - optional; type string
 solids_no - optional; type string
 quantity - optional; type double
 coating - optional; type double
 volume - optional; type double
 unit_weight - optional; type double
 solid_weight - optional; type double
 total_coating - optional; type double
 total_volume - optional; type double
 total_weight - optional; type double
```

### 95. `get_parts_list_surfaces_by_material`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_parts_list_surfaces_by_material`

**Input:** `get_parts_list_surfaces_by_material_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_surfaces_by_material
```

**Output:** `get_parts_list_surfaces_by_material_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_parts_list_surfaces_by_materialResponse
 value type array_of_parts_list_surfaces_by_material
 parts_list_surfaces_by_material - optional, unbounded; type parts_list_surfaces_by_material
 no - optional; type int
 material_name - optional; type string
 thickness_name - optional; type string
 surfaces_no - optional; type string
```

- quantity - optional; type *double*
- surface\_area - optional; type *double*
- coating - optional; type *double*
- volume - optional; type *double*
- unit\_weight - optional; type *double*
- surface\_weight - optional; type *double*
- total\_coating - optional; type *double*
- total\_surface\_area - optional; type *double*
- total\_volume - optional; type *double*
- total\_weight - optional; type *double*

## 96. get\_result\_combination

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_result\_combination

**Input:** get\_result\_combination\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_result\_combination*

- no type *int*

**Output:** get\_result\_combination\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_result\_combinationResponse*

- value type *result\_combination*
  - no type *int*
  - design\_situation - optional; type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - to\_solve - optional; type *boolean*
  - comment - optional; type *string*
  - combination\_type - optional; type *result\_combination\_combination\_type* - type *undefined* with restriction - enum { 'COMBINATION\_TYPE\_ENVELOPE\_PERMANENT', 'COMBINATION\_TYPE\_ENVELOPE\_TRANSIENT', 'COMBINATION\_TYPE\_GENERAL', 'COMBINATION\_TYPE\_SUPERPOSITION' }
  - srss\_combination - optional; type *boolean*
  - srss\_extreme\_value\_sign - optional; type *result\_combination\_srss\_extreme\_value\_sign* - type *undefined* with restriction - enum { 'EXTREME\_VALUE\_SIGN\_ACCORDING\_TO\_LC\_CO', 'EXTREME\_VALUE\_SIGN\_NEGATIVE', 'EXTREME\_VALUE\_SIGN\_POSITIVE', 'EXTREME\_VALUE\_SIGN\_POSITIVE\_OR\_NEGATIVE' }
  - srss\_use\_equivalent\_linear\_combination - optional; type *boolean*
  - srss\_according\_load\_case\_or\_combination - optional; type *int*
  - items - optional; type *array\_of\_result\_combination\_items*
    - result\_combination\_items - optional, unbounded; type *result\_combination\_items*
      - no - optional; type *int*
      - case\_object\_item - optional; type *int*
      - operator\_type - optional; type *operator\_type* - type *undefined* with restriction - enum { 'OPERATOR\_AND', 'OPERATOR\_NONE', 'OPERATOR\_OR' }
      - left\_parenthesis - optional; type *boolean*
      - right\_parenthesis - optional; type *boolean*
      - group\_factor - optional; type *double*
      - case\_object\_factor - optional; type *double*
      - case\_object\_sub\_result\_type - optional; type *case\_object\_sub\_result\_type* - type *undefined* with restriction - enum { 'SUB\_RESULT\_INCREMENTAL\_ALL', 'SUB\_RESULT\_INCREMENTAL\_FINAL\_STATE', 'SUB\_RESULT\_INCREMENTAL\_SUB\_RESULT\_ID', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_ABSOLUTE\_SUM', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_X', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_X\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Y', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Y\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Z', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Z\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUMS\_ENVELOPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_X', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_Y', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_Z', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SRSS' }
      - case\_object\_sub\_result\_id - optional; type *int*
      - case\_object\_load\_type - optional; type *case\_object\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
      - group\_load\_type - optional; type *group\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
      - action - optional; type *int*
      - is\_leading - optional; type *boolean*
      - gamma - optional; type *double*
      - psi - optional; type *double*
      - xi - optional; type *double*
      - k\_fi - optional; type *double*
      - c\_esl - optional; type *double*
      - k\_def - optional; type *double*
      - psi\_0 - optional; type *double*
      - psi\_1 - optional; type *double*
      - psi\_2 - optional; type *double*
      - fi - optional; type *double*
      - gamma\_0 - optional; type *double*
      - alfa - optional; type *double*
      - k\_f - optional; type *double*
      - phi - optional; type *double*
      - rho - optional; type *double*
      - omega\_0 - optional; type *double*
      - gamma\_l\_1 - optional; type *double*
      - k\_creep - optional; type *double*
  - generate\_subcombinations - optional; type *boolean*
  - load\_duration - optional; type *int*
  - is\_generated - optional; type *boolean*
  - consider\_construction\_stage - optional; type *int*
  - consider\_construction\_stage\_active - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*

- metadata\_for\_export\_import - optional; type *string*

## 97. get\_result\_section

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_result\_section

**Input:** get\_result\_section\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_result\_section*

- no type *int*

**Output:** get\_result\_section\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_result\_sectionResponse*

- value type *result\_section*
  - no type *int*
  - type - optional; type *result\_section\_type* - type *undefined* with restriction - enum { 'TYPE\_2\_POINTS\_AND\_VECTOR', 'TYPE\_LINE' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to\_surfaces - optional; type *array\_of\_int*
  - assigned\_to\_solids - optional; type *array\_of\_int*
  - show\_results\_in\_direction - optional; type *result\_section\_show\_results\_in\_direction* - type *undefined* with restriction - enum { 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_X', 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_Y', 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_Z', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_X', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_Y', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_Z', 'SHOW\_RESULTS\_IN\_LOCAL\_MINUS\_Z', 'SHOW\_RESULTS\_IN\_LOCAL\_PLUS\_Y', 'SHOW\_RESULTS\_IN\_LOCAL\_PLUS\_Z' }
  - coordinate\_system - optional; type *int*
  - show\_values\_on\_isolines\_enabled - optional; type *boolean*
  - lines - optional; type *array\_of\_int*
  - first\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - first\_point\_coordinate\_1 - optional; type *double*
  - first\_point\_coordinate\_2 - optional; type *double*
  - first\_point\_coordinate\_3 - optional; type *double*
  - second\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - second\_point\_coordinate\_1 - optional; type *double*
  - second\_point\_coordinate\_2 - optional; type *double*
  - second\_point\_coordinate\_3 - optional; type *double*
  - projection\_in\_direction - optional; type *result\_section\_projection\_in\_direction* - type *undefined* with restriction - enum { 'PROJECTION\_IN\_GLOBAL\_X', 'PROJECTION\_IN\_GLOBAL\_Y', 'PROJECTION\_IN\_GLOBAL\_Z', 'PROJECTION\_IN\_USER\_DEFINED\_U', 'PROJECTION\_IN\_USER\_DEFINED\_V', 'PROJECTION\_IN\_USER\_DEFINED\_W', 'PROJECTION\_IN\_VECTOR' }
  - vector - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - vector\_coordinate\_1 - optional; type *double*
  - vector\_coordinate\_2 - optional; type *double*
  - vector\_coordinate\_3 - optional; type *double*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 98. get\_rigid\_link

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_rigid\_link

**Input:** get\_rigid\_link\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_rigid\_link*

- no type *int*

**Output:** get\_rigid\_link\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_rigid\_linkResponse*

- value type *rigid\_link*
  - no type *int*
  - type - optional; type *rigid\_link\_type* - type *undefined* with restriction - enum { 'TYPE\_DIAPHRAGM', 'TYPE\_LINE\_TO\_LINE', 'TYPE\_LINE\_TO\_SURFACE' }
  - line1 - optional; type *int*
  - line2 - optional; type *int*
  - surface - optional; type *int*
  - rigid\_link\_type - optional; type *rigid\_link\_rigid\_link\_type* - type *undefined* with restriction - enum { 'RESILIENT', 'RIGID' }
  - comment - optional; type *string*
  - ignore\_relative\_position - optional; type *boolean*
  - user\_defined\_distribution - optional; type *boolean*
  - line1\_start\_is\_relative - optional; type *boolean*
  - line1\_start\_relative - optional; type *double*
  - line1\_start\_absolute - optional; type *double*
  - line1\_end\_is\_relative - optional; type *boolean*
  - line1\_end\_relative - optional; type *double*
  - line1\_end\_absolute - optional; type *double*
  - line2\_start\_is\_relative - optional; type *boolean*
  - line2\_start\_relative - optional; type *double*

- line2\_start\_absolute - optional; type *double*
- line2\_end\_is\_relative - optional; type *boolean*
- line2\_end\_relative - optional; type *double*
- line2\_end\_absolute - optional; type *double*
- nodes - optional; type *array\_of\_int*
- lines - optional; type *array\_of\_int*
- center\_node\_no - optional; type *int*
- center\_user\_defined - optional; type *boolean*
- center\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- center\_point\_x - optional; type *double*
- center\_point\_y - optional; type *double*
- center\_point\_z - optional; type *double*
- link\_plane\_user\_defined - optional; type *boolean*
- link\_plane\_node1\_no - optional; type *int*
- link\_plane\_node2\_no - optional; type *int*
- link\_plane\_node3\_no - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 99. get\_section

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_section](http://localhost:8082/get_section)

**Input:** [get\\_section\\_request](#) (soap:body, use = literal) [Source code](#)

parameters type *get\_section*

- no type *int*

**Output:** [get\\_section\\_response](#) (soap:body, use = literal) [Source code](#)

parameters type *get\_sectionResponse*

- value type *section*
  - no type *int*
  - type - optional; type *section\_type* - type *undefined* with restriction - enum { 'TYPE\_BASIC', 'TYPE\_BUILT\_UP\_STEEL', 'TYPE\_BUILT\_UP\_TIMBER', 'TYPE\_GENERAL\_BY\_RSECTION', 'TYPE\_PARAMETRIC\_BARS', 'TYPE\_PARAMETRIC\_BRIDGES', 'TYPE\_PARAMETRIC\_MASSIVE\_I', 'TYPE\_PARAMETRIC\_MASSIVE\_II', 'TYPE\_PARAMETRIC\_THIN\_WALLED', 'TYPE\_PHASE', 'TYPE\_STANDARDIZED\_STEEL', 'TYPE\_STANDARDIZED\_TIMBER' }
  - manufacturing\_type - optional; type *section\_manufacturing\_type* - type *undefined* with restriction - enum { 'MANUFACTURING\_TYPE\_COLD\_FORMED', 'MANUFACTURING\_TYPE\_GLULAM', 'MANUFACTURING\_TYPE\_HOT\_ROLLED', 'MANUFACTURING\_TYPE\_NONE', 'MANUFACTURING\_TYPE\_SAWN', 'MANUFACTURING\_TYPE\_WELDED' }
  - name - optional; type *string*
  - shear\_stiffness\_deactivated - optional; type *boolean*
  - warping\_stiffness\_deactivated - optional; type *boolean*
  - thin\_walled\_model - optional; type *boolean*
  - us\_spelling\_of\_properites - optional; type *boolean*
  - has\_cost\_estimation - optional; type *boolean*
  - has\_emissions\_estimation - optional; type *boolean*
  - optimization - optional; type *boolean*
  - stress\_smoothing\_to\_avoid\_singularities - optional; type *boolean*
  - area\_axial - optional; type *double*
  - area\_shear\_y - optional; type *double*
  - area\_shear\_z - optional; type *double*
  - inclination\_principal\_axes - optional; type *double*
  - rotation\_angle - optional; type *double*
  - mirrored\_axis\_y - optional; type *boolean*
  - mirrored\_axis\_z - optional; type *boolean*
  - moment\_of\_inertia\_bending\_y - optional; type *double*
  - moment\_of\_inertia\_bending\_z - optional; type *double*
  - moment\_of\_inertia\_torsion - optional; type *double*
  - warping - optional; type *double*
  - depth\_temperature\_load - optional; type *double*
  - width\_temperature\_load - optional; type *double*
  - material - optional; type *int*
  - reference\_material - optional; type *int*
  - material\_part\_1 - optional; type *int*
  - material\_part\_2 - optional; type *int*
  - material\_part\_3 - optional; type *int*
  - material\_part\_4 - optional; type *int*
  - material\_part\_5 - optional; type *int*
  - material\_part\_6 - optional; type *int*
  - material\_part\_7 - optional; type *int*
  - material\_part\_8 - optional; type *int*
  - material\_part\_9 - optional; type *int*
  - material\_part\_10 - optional; type *int*
  - hybrid\_active - optional; type *boolean*
  - parametrization\_type - optional; type *section\_parametrization\_type* - type *undefined* with restriction - enum { 'BUILT\_UP\_STEEL\_CHANNELS\_U\_BU', 'BUILT\_UP\_STEEL\_CUT\_I\_SECTIONS\_WITH\_FLAT\_BAR\_CIF\_BU', 'BUILT\_UP\_STEEL\_CUT\_I\_SECTIONS\_WITH\_MODIFIED\_DEPTH\_CIMD\_BU', 'BUILT\_UP\_STEEL\_EQUAL\_LEG\_ANGLES\_LE\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_OR\_CHANNELS\_WITH\_SIDE\_FLAT\_BARS\_SF\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_WITH\_CHANNELS\_IU\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_WITH\_OTHER\_SECTION\_ON\_FLANGE\_IJOF\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_I\_BU', 'BUILT\_UP\_STEEL\_UNEQUAL\_LEG\_ANGLES\_LU\_BU', 'BUILT\_UP\_TIMBER\_RECTANGLES\_R\_BU', 'PARAMETRIC\_BARS\_FLAT\_BAR\_FLAT', 'PARAMETRIC\_BARS\_HALF\_OVAL\_BAR\_HALFOVAL', 'PARAMETRIC\_BARS\_HALF\_ROUND\_BAR\_HALFRROUND', 'PARAMETRIC\_BARS\_HEXAGON\_BAR\_HEXAGON', 'PARAMETRIC\_BARS\_ISOSCELES\_RIGHT\_TRIANGLE\_BAR\_TRIANGLEIR', 'PARAMETRIC\_BARS\_OCTAGON\_BAR\_OCTAGON', 'PARAMETRIC\_BARS\_ROUND\_BAR\_ROUND' }



'PARAMETRIC\_BARS\_ROUND\_CORNER\_SQUARE\_BAR\_SQUARER',  
'PARAMETRIC\_BARS\_SHARP\_CORNER\_SQUARE\_BAR\_SQUARES',  
'PARAMETRIC\_BRIDGES\_MULTI\_PIECE\_TAPERED\_FLANGE\_T\_SECTION\_TTF\_B',  
'PARAMETRIC\_MASSIVE\_II\_2X2\_MASSIVE\_RECTANGLES\_WITH\_FULLY\_RIGID\_CONNECTION\_2X2R\_M2',  
'PARAMETRIC\_MASSIVE\_II\_2X3\_MASSIVE\_RECTANGLES\_WITH\_FULLY\_RIGID\_CONNECTION\_2X3R\_M2',  
'PARAMETRIC\_MASSIVE\_II\_2\_MASSIVE\_RECTANGLES\_WITH\_A\_CONTINUOUS\_PACK\_AND\_2\_SPLICES\_2RP2S\_M2',  
'PARAMETRIC\_MASSIVE\_II\_2\_MASSIVE\_RECTANGLES\_2R\_M2',  
'PARAMETRIC\_MASSIVE\_II\_3\_MASSIVE\_RECTANGLES\_3R\_M2',  
'PARAMETRIC\_MASSIVE\_II\_4\_MASSIVE\_RECTANGLES\_4R\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_BOXH\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_OVERHANGING\_FLANGES\_AND\_HORIZONTAL\_CONNECTION\_LINES',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_OVERHANGING\_FLANGES\_AND\_VERTICAL\_CONNECTION\_LINES',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_VERTICAL\_CONNECTION\_LINES\_BOXV\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_FIVE\_LAYER\_RECTANGLE\_WITH\_3\_DIFFERENT\_MATERIALS\_5LR\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_OVAL\_WITH\_VERTICAL\_CONNECTION\_LINES\_OVAL\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_RECTANGLE\_WITH\_2\_CONTINUOUS\_SPLICES\_R2S\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_I',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_THIN\_FLANGES\_AND\_4\_REINFORCING',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_ISV',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_THREE\_LAYER\_RECTANGLE\_WITH\_2\_DIFFERENT\_MATERIALS\_3LR\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_BOX\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_2BOXH\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_BOX\_WITH\_VERTICAL\_CONNECTION\_LINES\_2BOXV\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_LAYER\_RECTANGLE\_WITH\_2\_DIFFERENT\_MATERIALS\_2LR\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_T\_SECTION\_WITH\_A\_HORIZONTAL\_CONNECTION\_LINE\_TH\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_T\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_TV\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_U\_SECTION\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_UH\_M2',  
'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_U\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_UV\_M2',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_CIRCLE\_CIRCLE\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLE\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_DTC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLE\_T\_SECTION\_DT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLY\_SYMMETRIC\_I\_SECTION\_ID\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_OVAL\_HALFOVAL\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_ROUND\_HALFRound\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HEXAGON\_HEXAGON\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HOLLOW\_CIRCLE\_HCIRCLE\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_I\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_L\_SECTION\_L\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OCTAGON\_OCTAGON\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OVAL\_OVAL\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_CHAMFERED\_RECTANGULAR\_OPENING\_RROC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_RECTANGULAR\_OPENING\_RRO\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_R\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_RECTANGLE\_RR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_SQUARE\_SQR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_SQUARE\_SQ\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_DOUBLE\_T\_SECTION\_DTT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_I\_SECTION\_ITF\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_T\_SECTION\_ITT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_U\_SECTION\_UT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TTWC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_TTW\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TRAPEZOID\_TR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_T\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IUC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_IU\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_T\_SECTION\_TU\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_UC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_U\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_Z\_SECTION\_Z\_M1',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_AX\_BOX\_AX',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_A\_BOX\_A',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_BX\_BOX\_BX',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_B\_BOX\_B',  
'PARAMETRIC\_THIN\_WALLED\_CHANNEL\_U',  
'PARAMETRIC\_THIN\_WALLED\_CIRCULAR\_HOLLOW\_SECTION\_CHS',  
'PARAMETRIC\_THIN\_WALLED\_CROSS\_SHAPED\_SECTION\_CROSS',  
'PARAMETRIC\_THIN\_WALLED\_ELLIPTICAL\_HOLLOW\_SECTION\_EHS',  
'PARAMETRIC\_THIN\_WALLED\_EQUAL\_LEG\_ANGLE\_LE',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_2\_SIDE\_FLAT\_BARS\_I2SFB',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_T\_SECTION\_IT',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_I',  
'PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_A\_PI\_A',  
'PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_B\_PI\_B',  
'PARAMETRIC\_THIN\_WALLED\_POLYGONAL\_HOLLOW\_SECTION\_PHS',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAKS\_OUTWARD\_AND\_INWARD\_RHSPD',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAK\_OUTWARD\_RHSPD',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_RHS',  
'PARAMETRIC\_THIN\_WALLED\_RIB\_HOLLOW\_SECTION\_RIBHS',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_FLAT\_BARS\_IS2FB',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_SLOPING\_STIFFENERS\_IS2SS',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_WELDED\_FLAT\_BARS\_IS2FBW',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_DOUBLE\_WEB\_THICKNESS\_IS2WT',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_EDGE\_STIFFENERS\_ON\_1\_FLANGE\_IS1FES',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_IS',  
'PARAMETRIC\_THIN\_WALLED\_SQUARE\_HOLLOW\_SECTION\_SHS',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_CHANNEL\_UTF',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_I\_SECTION\_ITF',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_WITH\_TAPERED\_WEB\_TTFW',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_TTF',  
'PARAMETRIC\_THIN\_WALLED\_TRAPEZOIDAL\_HOLLOW\_SECTION\_THS',  
'PARAMETRIC\_THIN\_WALLED\_T\_SECTION\_T',  
'PARAMETRIC\_THIN\_WALLED\_UNEQUAL\_LEG\_ANGLE\_LU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_INWARD\_UUESI',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_AND\_INWARD\_UUE',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_UUESO',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_UU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_I\_SECTION\_IU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_RECTANGULAR\_HOLLOW\_SECTION\_RHSU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_T\_SECTION\_TU',  
'PARAMETRIC\_THIN\_WALLED\_Z\_SECTION\_Z',  
'STANDARDIZED\_STEEL\_60\_DEGREE\_ANGLES\_L60\_S',  
'STANDARDIZED\_STEEL\_BAR\_SECTIONS\_BAR\_S',  
'STANDARDIZED\_STEEL\_CIRCULAR\_HOLLOW\_SECTIONS\_PIPES\_CHS\_S',  
'STANDARDIZED\_STEEL\_ELLIPTICAL\_AND\_SEMI\_ELLIPTICAL\_HOLLOW\_SECTIONS\_EHS\_S',

'STANDARDIZED\_STEEL\_I\_BEAMS\_H\_BEAMS\_I\_S',  
'STANDARDIZED\_STEEL\_L\_SECTIONS\_STEEL\_ANGLES\_L\_S',  
'STANDARDIZED\_STEEL\_RAIL\_SECTIONS\_RAIL\_S', 'STANDARDIZED\_STEEL\_SHEETS\_SHEET\_S',  
'STANDARDIZED\_STEEL\_SQUARE\_AND\_RECTANGULAR\_HOLLOW\_SECTIONS\_RHS\_S',  
'STANDARDIZED\_STEEL\_T\_SECTIONS\_STEEL\_TEES\_T\_S',  
'STANDARDIZED\_STEEL\_U\_SECTIONS\_STEEL\_CHANNELS\_U\_S',  
'STANDARDIZED\_STEEL\_Z\_SECTIONS\_STEEL\_ZEES\_Z\_S',  
'STANDARDIZED\_TIMBER\_TIMBER\_COMPOSED\_SECTIONS\_COMP\_S',  
'STANDARDIZED\_TIMBER\_TIMBER\_RECTANGLES\_RECT\_S'}

- combination\_type - optional; type *section\_combination\_type* - type *undefined* with restriction - enum {  
'BUILT\_UP\_NG\_2I', 'BUILT\_UP\_NG\_2I\_A', 'BUILT\_UP\_NG\_2LCLI', 'BUILT\_UP\_NG\_2LCLIR',  
'BUILT\_UP\_NG\_2LCLIR\_A', 'BUILT\_UP\_NG\_2LCLI\_A', 'BUILT\_UP\_NG\_2LCLO', 'BUILT\_UP\_NG\_2LCLOC',  
'BUILT\_UP\_NG\_2LCLOC\_A', 'BUILT\_UP\_NG\_2LCLO\_A', 'BUILT\_UP\_NG\_2LCLO\_A', 'BUILT\_UP\_NG\_2LCLO\_A',  
'BUILT\_UP\_NG\_2LHLI', 'BUILT\_UP\_NG\_2LHLI\_A', 'BUILT\_UP\_NG\_2LHLO', 'BUILT\_UP\_NG\_2LHLO\_A',  
'BUILT\_UP\_NG\_2LLHLI', 'BUILT\_UP\_NG\_2LLHLI\_A', 'BUILT\_UP\_NG\_2LLHLO', 'BUILT\_UP\_NG\_2LLHLO\_A',  
'BUILT\_UP\_NG\_2LSHLI', 'BUILT\_UP\_NG\_2LSHLI\_A', 'BUILT\_UP\_NG\_2LSHLO', 'BUILT\_UP\_NG\_2LSHLO\_A',  
'BUILT\_UP\_NG\_2UF', 'BUILT\_UP\_NG\_2UI', 'BUILT\_UP\_NG\_2UI\_A', 'BUILT\_UP\_NG\_2UO', 'BUILT\_UP\_NG\_2UO\_A',  
'BUILT\_UP\_NG\_3I', 'BUILT\_UP\_NG\_4LHLO', 'BUILT\_UP\_NG\_4LHLO\_A', 'BUILT\_UP\_NG\_4LLI',  
'BUILT\_UP\_NG\_4LLI\_A', 'BUILT\_UP\_NG\_4LLO', 'BUILT\_UP\_NG\_4LLO\_A', 'BUILT\_UP\_NG\_C1HF',  
'BUILT\_UP\_NG\_C1MD', 'BUILT\_UP\_NG\_HNBH', 'BUILT\_UP\_NG\_I1F', 'BUILT\_UP\_NG\_I1HIC', 'BUILT\_UP\_NG\_I1IC',  
'BUILT\_UP\_NG\_I2F', 'BUILT\_UP\_NG\_I2HIC', 'BUILT\_UP\_NG\_I2I', 'BUILT\_UP\_NG\_ICI-MAX', 'BUILT\_UP\_NG\_ICI-MIN',  
'BUILT\_UP\_NG\_IHF', 'BUILT\_UP\_NG\_IUD', 'BUILT\_UP\_NG\_IUU', 'BUILT\_UP\_NG\_IVF', 'BUILT\_UP\_NG\_IWT',  
'BUILT\_UP\_NG\_UF' }
- corrugated\_sheet\_width - optional; type *double*
- rail\_worn\_out - optional; type *double*
- phase\_parent - optional; type *int*
- phase\_parts - optional; type *string*
- comment - optional; type *string*
- advanced\_time\_dependent\_properties\_of\_concrete\_enabled - optional; type *boolean*
- creep\_enabled - optional; type *boolean*
- shrinkage\_enabled - optional; type *boolean*
- relative\_humidity - optional; type *double*
- function\_data\_function\_type - optional; type *section\_function\_data\_function\_type* - type *undefined* with restriction - enum { 'FUNCTION\_TYPE\_CREEP\_COEFFICIENT', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CA', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CD', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CS' }
- function\_data\_part\_id - optional; type *int*
- function\_data\_age\_of\_concrete\_at\_the\_considered\_moment - optional; type *double*
- function\_data\_number\_of\_steps - optional; type *int*
- function\_data\_coefficients - optional; type *array\_of\_section\_function\_data\_coefficients*
  - section\_function\_data\_coefficients - optional, unbounded; type *section\_function\_data\_coefficients*
    - no - optional; type *int*
    - time - optional; type *double*
    - coefficient - optional; type *double*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- cost\_estimation\_apply\_from\_material - optional; type *boolean*
- members\_weight\_active - optional; type *boolean*
- members\_weight\_unit\_cost - optional; type *double*
- members\_weight\_unit - optional; type *section\_members\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- members\_weight\_quantity - optional; type *double*
- members\_weight\_cost - optional; type *double*
- members\_volume\_active - optional; type *boolean*
- members\_volume\_unit\_cost - optional; type *double*
- members\_volume\_unit - optional; type *section\_members\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- members\_volume\_quantity - optional; type *double*
- members\_volume\_cost - optional; type *double*
- members\_surface\_active - optional; type *boolean*
- members\_surface\_unit\_cost - optional; type *double*
- members\_surface\_unit - optional; type *section\_members\_surface\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- members\_surface\_quantity - optional; type *double*
- members\_surface\_cost - optional; type *double*
- members\_length\_active - optional; type *boolean*
- members\_length\_unit\_cost - optional; type *double*
- members\_length\_unit - optional; type *section\_members\_length\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM', 'COST\_ESTIMATION\_FT', 'COST\_ESTIMATION\_IN', 'COST\_ESTIMATION\_KM', 'COST\_ESTIMATION\_M', 'COST\_ESTIMATION\_MM', 'COST\_ESTIMATION\_YD' }
- members\_length\_quantity - optional; type *double*
- members\_length\_cost - optional; type *double*
- sum\_cost - optional; type *double*
- cost\_percentage - optional; type *double*
- total\_cost - optional; type *double*
- emissions\_estimation\_apply\_from\_material - optional; type *boolean*
- emissions\_members\_weight\_active - optional; type *boolean*
- emissions\_members\_weight\_unit\_cost - optional; type *double*
- emissions\_members\_weight\_unit - optional; type *section\_emissions\_members\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CWT', 'EMISSIONS\_ESTIMATION\_G', 'EMISSIONS\_ESTIMATION\_KG', 'EMISSIONS\_ESTIMATION\_LB', 'EMISSIONS\_ESTIMATION\_OZ', 'EMISSIONS\_ESTIMATION\_SLUG', 'EMISSIONS\_ESTIMATION\_T', 'EMISSIONS\_ESTIMATION\_TON' }
- emissions\_members\_weight\_quantity - optional; type *double*
- emissions\_members\_weight\_cost - optional; type *double*
- emissions\_members\_volume\_active - optional; type *boolean*
- emissions\_members\_volume\_unit\_cost - optional; type *double*
- emissions\_members\_volume\_unit - optional; type *section\_emissions\_members\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM3', 'EMISSIONS\_ESTIMATION\_FLOZ', 'EMISSIONS\_ESTIMATION\_FT3', 'EMISSIONS\_ESTIMATION\_IMP GAL', 'EMISSIONS\_ESTIMATION\_IN3', 'EMISSIONS\_ESTIMATION\_L', 'EMISSIONS\_ESTIMATION\_M3', 'EMISSIONS\_ESTIMATION\_MM3', 'EMISSIONS\_ESTIMATION\_PT', 'EMISSIONS\_ESTIMATION\_QT', 'EMISSIONS\_ESTIMATION\_USGAL', 'EMISSIONS\_ESTIMATION\_YD3' }
- emissions\_members\_volume\_quantity - optional; type *double*
- emissions\_members\_volume\_cost - optional; type *double*
- emissions\_members\_surface\_active - optional; type *boolean*

- emissions\_members\_surface\_unit\_cost - optional; type *double*
- emissions\_members\_surface\_unit - optional; type *section\_emissions\_members\_surface\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM2', 'EMISSIONS\_ESTIMATION\_FT2', 'EMISSIONS\_ESTIMATION\_IN2', 'EMISSIONS\_ESTIMATION\_M2', 'EMISSIONS\_ESTIMATION\_MM2', 'EMISSIONS\_ESTIMATION\_YD2' }
- emissions\_members\_surface\_quantity - optional; type *double*
- emissions\_members\_surface\_cost - optional; type *double*
- emissions\_members\_length\_active - optional; type *boolean*
- emissions\_members\_length\_unit\_cost - optional; type *double*
- emissions\_members\_length\_unit - optional; type *section\_emissions\_members\_length\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM', 'EMISSIONS\_ESTIMATION\_FT', 'EMISSIONS\_ESTIMATION\_IN', 'EMISSIONS\_ESTIMATION\_KM', 'EMISSIONS\_ESTIMATION\_M', 'EMISSIONS\_ESTIMATION\_MM', 'EMISSIONS\_ESTIMATION\_YD' }
- emissions\_members\_length\_quantity - optional; type *double*
- emissions\_members\_length\_cost - optional; type *double*
- emissions\_sum\_cost - optional; type *double*
- emissions\_cost\_percentage - optional; type *double*
- emissions\_total\_cost - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 100. get\_soil\_massif

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_soil\_massif

**Input:** get\_soil\_massif\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_soil\_massif*

- no type *int*

**Output:** get\_soil\_massif\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_soil\_massifResponse*

- value type *soil\_massif*
  - no type *int*
  - type - optional; type *soil\_massif\_type* - type *undefined* with restriction - enum { 'TYPE\_PHANTOM', 'TYPE\_STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to\_type - optional; type *soil\_massif\_assigned\_to\_type* - type *undefined* with restriction - enum { 'ASSIGNED\_TO\_TYPE\_SOIL\_SAMPLES', 'ASSIGNED\_TO\_TYPE\_SOIL\_SOLIDS' }
  - assigned\_to\_soil\_samples - optional; type *array\_of\_int*
  - assigned\_to\_solids - optional; type *array\_of\_int*
  - assigned\_to\_solid\_sets - optional; type *array\_of\_int*
  - assigned\_to\_solids\_and\_solid\_sets - optional; type *string*
  - topology\_type - optional; type *soil\_massif\_topology\_type* - type *undefined* with restriction - enum { 'TOPOLOGY\_TYPE\_RECTANGLE' }
  - depth\_according\_to\_soil\_samples - optional; type *boolean*
  - diameter\_for\_circle\_topology - optional; type *double*
  - polyline\_for\_polygon\_topology - optional; type *int*
  - center\_x - optional; type *double*
  - center\_y - optional; type *double*
  - size - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - size\_x - optional; type *double*
  - size\_y - optional; type *double*
  - size\_z - optional; type *double*
  - rotation\_about\_z - optional; type *double*
  - groundwater - optional; type *boolean*
  - groundwater\_surface - optional; type *int*
  - model\_soil\_block\_via - optional; type *soil\_massif\_model\_soil\_block\_via* - type *undefined* with restriction - enum { 'MODEL\_SOIL\_BLOCK\_VIA\_FE\_SOLIDS', 'MODEL\_SOIL\_BLOCK\_VIA\_SUPPORT\_SPRINGS' }
  - relevant\_permanent\_loading - optional; type *int*
  - mesh\_nodes\_in\_position\_of\_precise\_layer\_surfaces - optional; type *boolean*
  - rock\_beneath\_last\_layer - optional; type *boolean*
  - change\_mesh\_refinement - optional; type *boolean*
  - size\_increasing\_fe\_during\_growing\_depth - optional; type *double*
  - comment - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 101. get\_soil\_sample

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_soil\_sample

**Input:** get\_soil\_sample\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_soil\_sample*

- no type *int*

**Output:** get\_soil\_sample\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_soil\_sampleResponse*

- value type *soil\_sample*
  - no type *int*
  - type - optional; type *soil\_sample\_type* - type *undefined* with restriction - enum { 'TYPE\_PHANTOM', 'TYPE\_STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - coordinates - optional; type *vector\_3d*
    - x type *double*

- y type *double*
- z type *double*
- coordinate\_0 - optional; type *double*
- coordinate\_1 - optional; type *double*
- coordinate\_2 - optional; type *double*
- import\_coordinate\_z\_from\_terrain - optional; type *boolean*
- groundwater - optional; type *boolean*
- groundwater\_ordinate - optional; type *double*
- layers\_table - optional; type *array\_of\_soil\_sample\_layers\_table*
  - soil\_sample\_layers\_table - optional, unbounded; type *soil\_sample\_layers\_table*
    - no - optional; type *int*
    - layer\_no - optional; type *int*
    - soil\_material - optional; type *int*
    - thickness - optional; type *double*
    - bottom\_ordinate - optional; type *double*
- layers\_thickness\_sum - optional; type *double*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 102. get\_solid

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid

**Input:** get\_solid\_request (soap:body, use = literal) [Source code](#)

- parameters type *get\_solid*
- no type *int*

**Output:** get\_solid\_response (soap:body, use = literal) [Source code](#)

- parameters type *get\_solidResponse*
- value type *solid*
    - no type *int*
    - type - optional; type *solid\_type* - type *undefined* with restriction - enum { 'TYPE\_CONTACT', 'TYPE\_GAS', 'TYPE\_HOLE', 'TYPE\_INTERSECTION', 'TYPE\_SOIL', 'TYPE\_STANDARD' }
    - analytical\_center\_of\_gravity - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - analytical\_center\_of\_gravity\_x - optional; type *double*
    - analytical\_center\_of\_gravity\_y - optional; type *double*
    - analytical\_center\_of\_gravity\_z - optional; type *double*
    - analytical\_mass - optional; type *double*
    - analytical\_surface\_area - optional; type *double*
    - analytical\_volume - optional; type *double*
    - boundary\_surfaces - optional; type *array\_of\_int*
    - center\_of\_gravity - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - center\_of\_gravity\_x - optional; type *double*
    - center\_of\_gravity\_y - optional; type *double*
    - center\_of\_gravity\_z - optional; type *double*
    - gas - optional; type *int*
    - is\_deactivated\_for\_calculation - optional; type *boolean*
    - mass - optional; type *double*
    - material - optional; type *int*
    - mesh\_refinement - optional; type *int*
    - solid\_contact - optional; type *int*
    - solid\_contact\_first\_surface - optional; type *int*
    - solid\_contact\_second\_surface - optional; type *int*
    - stress\_analysis\_configuration - optional; type *int*
    - surface\_area - optional; type *double*
    - volume - optional; type *double*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - grid\_enabled - optional; type *boolean*
    - grid\_inner\_points - optional; type *boolean*
    - grid\_boundary\_points - optional; type *boolean*
    - specific\_direction\_type - optional; type *solid\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
    - grid\_distance\_x - optional; type *double*
    - grid\_distance\_y - optional; type *double*
    - grid\_distance\_z - optional; type *double*
    - use\_center\_of\_gravity\_as\_grid\_origin - optional; type *boolean*
    - grid\_origin - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - grid\_origin\_x - optional; type *double*
    - grid\_origin\_y - optional; type *double*
    - grid\_origin\_z - optional; type *double*
    - grid\_adapt\_automatically - optional; type *boolean*
    - grid\_point\_count\_negative\_u - optional; type *int*
    - grid\_point\_count\_positive\_u - optional; type *int*
    - grid\_point\_count\_negative\_v - optional; type *int*
    - grid\_point\_count\_positive\_v - optional; type *int*
    - grid\_point\_count\_negative\_w - optional; type *int*
    - grid\_point\_count\_positive\_w - optional; type *int*
    - specific\_direction\_enabled - optional; type *boolean*

- coordinate\_system - optional; type *int*
- axes\_sequence - optional; type *solid\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
- rotated\_about\_angle\_x - optional; type *double*
- rotated\_about\_angle\_y - optional; type *double*
- rotated\_about\_angle\_z - optional; type *double*
- rotated\_about\_angle\_1 - optional; type *double*
- rotated\_about\_angle\_2 - optional; type *double*
- rotated\_about\_angle\_3 - optional; type *double*
- directed\_to\_node\_direction\_node - optional; type *int*
- directed\_to\_node\_plane\_node - optional; type *int*
- directed\_to\_node\_first\_axis - optional; type *solid\_directed\_to\_node\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- directed\_to\_node\_second\_axis - optional; type *solid\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *solid\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_axis - optional; type *solid\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 103. get\_solid\_contacts

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid\_contacts

**Input:** get\_solid\_contacts\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_solid\_contacts*

- no type *int*

**Output:** get\_solid\_contacts\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_solid\_contactsResponse*

- value type *solid\_contacts*
  - no type *int*
  - comment - optional; type *string*
  - friction\_coefficient - optional; type *double*
  - limit\_stress - optional; type *double*
  - name - optional; type *string*
  - parallel\_to\_surface - optional; type *solid\_contacts\_parallel\_to\_surface* - type *undefined* with restriction - enum { 'ELASTIC\_FRICTION', 'ELASTIC\_FRICTION\_LIMIT', 'ELASTIC\_SOLID', 'FAILURE\_IF\_CONTACT\_PERPENDICULAR\_TO\_SURFACES\_FAILED', 'FULL\_FORCE\_TRANSMISSION', 'RIGID\_FRICTION', 'RIGID\_FRICTION\_LIMIT' }
  - perpendicular\_to\_surface - optional; type *solid\_contacts\_perpendicular\_to\_surface* - type *undefined* with restriction - enum { 'FAILURE\_UNDER\_COMPRESSION', 'FAILURE\_UNDER\_TENSION', 'FULL\_FORCE\_TRANSMISSION' }
  - shear\_stiffness - optional; type *double*
  - solids - optional; type *array\_of\_int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 104. get\_solid\_gas

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid\_gas

**Input:** get\_solid\_gas\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_solid\_gas*

- no type *int*

**Output:** get\_solid\_gas\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_solid\_gasResponse*

- value type *solid\_gas*
  - no type *int*
  - comment - optional; type *string*
  - name - optional; type *string*
  - pressure - optional; type *double*
  - solids - optional; type *array\_of\_int*
  - temperature - optional; type *double*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 105. get\_solid\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid\_load

**Input:** get\_solid\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_solid\_load*

- no type *int*
- load\_case\_no type *int*

Output: `get_solid_load_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_loadResponse
 value type solid_load
 no type int
 load_type - optional; type solid_load_load_type - type undefined with restriction - enum {
 'LOAD_TYPE_BUOYANCY', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_GAS', 'LOAD_TYPE_ROTARY_MOTION',
 'LOAD_TYPE_STRAIN', 'LOAD_TYPE_TEMPERATURE' }
 solids - optional; type array_of_int
 load_case - optional; type int
 load_distribution - optional; type solid_load_load_distribution - type undefined with restriction - enum {
 'LOAD_DISTRIBUTION_LINEAR_IN_X', 'LOAD_DISTRIBUTION_LINEAR_IN_Y',
 'LOAD_DISTRIBUTION_LINEAR_IN_Z', 'LOAD_DISTRIBUTION_UNIFORM' }
 load_direction - optional; type solid_load_load_direction - type undefined with restriction - enum {
 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE',
 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE',
 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE' }
 load_direction_orientation - optional; type solid_load_load_direction_orientation - type undefined with restriction -
enum { 'LOAD_DIRECTION_FORWARD', 'LOAD_DIRECTION_REVERSED' }
 uniform_magnitude - optional; type double
 magnitude_1 - optional; type double
 magnitude_2 - optional; type double
 strain_uniform_magnitude_x - optional; type double
 strain_uniform_magnitude_y - optional; type double
 strain_uniform_magnitude_z - optional; type double
 strain_magnitude_x1 - optional; type double
 strain_magnitude_y1 - optional; type double
 strain_magnitude_z1 - optional; type double
 strain_magnitude_x2 - optional; type double
 strain_magnitude_y2 - optional; type double
 strain_magnitude_z2 - optional; type double
 node_1 - optional; type int
 node_2 - optional; type int
 is_density_defined_by_altitude - optional; type boolean
 altitude - optional; type double
 angular_velocity - optional; type double
 angular_acceleration - optional; type double
 axis_definition_type - optional; type solid_load_axis_definition_type - type undefined with restriction - enum {
 'AXIS_DEFINITION_POINT_AND_AXIS', 'AXIS_DEFINITION_TWO_POINTS' }
 axis_definition_p1 - optional; type vector_3d
 x type double
 y type double
 z type double
 axis_definition_p1_x - optional; type double
 axis_definition_p1_y - optional; type double
 axis_definition_p1_z - optional; type double
 axis_definition_p2 - optional; type vector_3d
 x type double
 y type double
 z type double
 axis_definition_p2_x - optional; type double
 axis_definition_p2_y - optional; type double
 axis_definition_p2_z - optional; type double
 axis_definition_axis - optional; type solid_load_axis_definition_axis - type undefined with restriction - enum {
 'AXIS_X', 'AXIS_Y', 'AXIS_Z' }
 axis_definition_axis_orientation - optional; type solid_load_axis_definition_axis_orientation - type undefined with
restriction - enum { 'AXIS_NEGATIVE', 'AXIS_POSITIVE' }
 gas_magnitude - optional; type double
 gas_behaviour - optional; type solid_load_gas_behaviour - type undefined with restriction - enum {
 'GAS_BEHAVIOUR_OVERPRESSURE_INCREMENT', 'GAS_BEHAVIOUR_RESULTING_OVERPRESSURE',
 'GAS_BEHAVIOUR_RESULTING_VOLUME', 'GAS_BEHAVIOUR_VOLUME_INCREMENT' }
 coordinate_system - optional; type int
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

#### 106. `get_solid_mesh_refinement`

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: `http://localhost:8082/get_solid_mesh_refinement`

Input: `get_solid_mesh_refinement_request` (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_mesh_refinement
 no type int
```

Output: `get_solid_mesh_refinement_response` (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_mesh_refinementResponse
 value type solid_mesh_refinement
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 solids - optional; type array_of_int
 target_length - optional; type double
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

#### 107. `get_solid_set`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid\_set

**Input:** get\_solid\_set\_request (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_set
 ▪ no type int
```

**Output:** get\_solid\_set\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_setResponse
 ▪ value type solid_set
 ▪ no type int
 ▪ user_defined_name_enabled - optional; type boolean
 ▪ name - optional; type string
 ▪ set_type - optional; type solid_set_set_type - type undefined with restriction - enum { 'SET_TYPE_CONTINUOUS', 'SET_TYPE_GROUP' }
 ▪ solids - optional; type array_of_int
 ▪ surface_area - optional; type double
 ▪ volume - optional; type double
 ▪ mass - optional; type double
 ▪ center_of_gravity - optional; type vector_3d
 ▪ x type double
 ▪ y type double
 ▪ z type double
 ▪ center_of_gravity_x - optional; type double
 ▪ center_of_gravity_y - optional; type double
 ▪ center_of_gravity_z - optional; type double
 ▪ stress_analysis_configuration - optional; type int
 ▪ comment - optional; type string
 ▪ is_generated - optional; type boolean
 ▪ generating_object_info - optional; type string
 ▪ id_for_export_import - optional; type string
 ▪ metadata_for_export_import - optional; type string
```

#### 108. get\_solid\_set\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_solid\_set\_load

**Input:** get\_solid\_set\_load\_request (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_set_load
 ▪ no type int
 ▪ load_case_no type int
```

**Output:** get\_solid\_set\_load\_response (soap:body, use = literal) [Source code](#)

```
parameters type get_solid_set_loadResponse
 ▪ value type solid_set_load
 ▪ no type int
 ▪ load_type - optional; type solid_set_load_load_type - type undefined with restriction - enum { 'LOAD_TYPE_BUOYANCY', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_GAS', 'LOAD_TYPE_ROTARY_MOTION', 'LOAD_TYPE_STRAIN', 'LOAD_TYPE_TEMPERATURE' }
 ▪ solid_sets - optional; type array_of_int
 ▪ load_case - optional; type int
 ▪ load_distribution - optional; type solid_set_load_load_distribution - type undefined with restriction - enum { 'LOAD_DISTRIBUTION_LINEAR_IN_X', 'LOAD_DISTRIBUTION_LINEAR_IN_Y', 'LOAD_DISTRIBUTION_LINEAR_IN_Z', 'LOAD_DISTRIBUTION_UNIFORM' }
 ▪ load_direction - optional; type solid_set_load_load_direction - type undefined with restriction - enum { 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE', 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE', 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE' }
 ▪ load_direction_orientation - optional; type solid_set_load_load_direction_orientation - type undefined with restriction - enum { 'LOAD_DIRECTION_FORWARD', 'LOAD_DIRECTION_REVERSED' }
 ▪ uniform_magnitude - optional; type double
 ▪ magnitude_1 - optional; type double
 ▪ magnitude_2 - optional; type double
 ▪ strain_uniform_magnitude_x - optional; type double
 ▪ strain_uniform_magnitude_y - optional; type double
 ▪ strain_uniform_magnitude_z - optional; type double
 ▪ strain_magnitude_x1 - optional; type double
 ▪ strain_magnitude_y1 - optional; type double
 ▪ strain_magnitude_z1 - optional; type double
 ▪ strain_magnitude_x2 - optional; type double
 ▪ strain_magnitude_y2 - optional; type double
 ▪ strain_magnitude_z2 - optional; type double
 ▪ node_1 - optional; type int
 ▪ node_2 - optional; type int
 ▪ is_density_defined_by_altitude - optional; type boolean
 ▪ altitude - optional; type double
 ▪ angular_acceleration - optional; type double
 ▪ angular_velocity - optional; type double
 ▪ axis_definition_type - optional; type solid_set_load_axis_definition_type - type undefined with restriction - enum { 'AXIS_DEFINITION_POINT_AND_AXIS', 'AXIS_DEFINITION_TWO_POINTS' }
 ▪ axis_definition_p1 - optional; type vector_3d
 ▪ x type double
 ▪ y type double
 ▪ z type double
 ▪ axis_definition_p1_x - optional; type double
 ▪ axis_definition_p1_y - optional; type double
 ▪ axis_definition_p1_z - optional; type double
 ▪ axis_definition_p2 - optional; type vector_3d
 ▪ x type double
 ▪ y type double
```

- z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *solid\_set\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *solid\_set\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- gas\_magnitude - optional; type *double*
- gas\_behaviour - optional; type *solid\_set\_load\_gas\_behaviour* - type *undefined* with restriction - enum { 'GAS\_BEHAVIOUR\_OVERPRESSURE\_INCREMENT', 'GAS\_BEHAVIOUR\_RESULTING\_OVERPRESSURE', 'GAS\_BEHAVIOUR\_VOLUME\_INCREMENT', 'GAS\_BEHAVIOUR\_VOLUME\_INCREMENT' }
- coordinate\_system - optional; type *int*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 109. get\_spectral\_analysis\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_spectral\_analysis\_settings

**Input:** get\_spectral\_analysis\_settings\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_spectral\_analysis\_settings*

- no type *int*

**Output:** get\_spectral\_analysis\_settings\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_spectral\_analysis\_settingsResponse*

- value type *spectral\_analysis\_settings*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - comment - optional; type *string*
  - assigned\_to - optional; type *string*
  - combination\_rule\_for\_periodic\_responses - optional; type *spectral\_analysis\_settings\_combination\_rule\_for\_periodic\_responses* - type *undefined* with restriction - enum { 'ABSOLUTE\_SUM', 'CQC', 'SRSS' }
  - use\_equivalent\_linear\_combination - optional; type *boolean*
  - signed\_results\_using\_dominant\_mode - optional; type *boolean*
  - include\_missing\_masses - optional; type *boolean*
  - combination\_rule\_for\_missing\_masses - optional; type *spectral\_analysis\_settings\_combination\_rule\_for\_missing\_masses* - type *undefined* with restriction - enum { 'ABSOLUTE\_SUM', 'SRSS' }
  - save\_results\_of\_all\_selected\_modes - optional; type *boolean*
  - combination\_rule\_for\_directional\_components - optional; type *spectral\_analysis\_settings\_combination\_rule\_for\_directional\_components* - type *undefined* with restriction - enum { 'ABSOLUTE\_SUM', 'SCALED\_SUM', 'SRSS' }
  - combination\_rule\_for\_directional\_components\_value - optional; type *double*
  - damping\_for\_cqc\_rule - optional; type *spectral\_analysis\_settings\_damping\_for\_cqc\_rule* - type *undefined* with restriction - enum { 'CONSTANT\_FOR\_EACH\_MODE', 'DIFFERENT\_FOR\_EACH\_MODE' }
  - constant\_d\_for\_each\_mode - optional; type *double*
  - zero\_periodic\_acceleration\_type - optional; type *spectral\_analysis\_settings\_zero\_periodic\_acceleration\_type* - type *undefined* with restriction - enum { 'ACCORDING\_TO\_RESPONSE\_SPECTRUM', 'SPECTRAL\_ACCELERATION\_OF\_LAST\_CALCULATED\_FREQUENCY', 'USER\_DEFINED' }
  - user\_defined\_zpa - optional; type *double*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 110. get\_static\_analysis\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_static\_analysis\_settings

**Input:** get\_static\_analysis\_settings\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_static\_analysis\_settings*

- no type *int*

**Output:** get\_static\_analysis\_settings\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_static\_analysis\_settingsResponse*

- value type *static\_analysis\_settings*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - analysis\_type - optional; type *static\_analysis\_settings\_analysis\_type* - type *undefined* with restriction - enum { 'GEOMETRICALLY\_LINEAR', 'LARGE\_DEFORMATIONS', 'SECOND\_ORDER\_P\_DELTA' }
  - assign\_reduce\_stiffness\_enabled - optional; type *static\_analysis\_settings\_assign\_reduce\_stiffness\_enabled* - type *undefined* with restriction - enum { 'ASSIGN\_REDUCED\_STIFFNESS\_TO\_FAILING\_MEMBERS', 'FAILING\_MEMBERS\_TO\_BE\_REMOVED\_INDIVIDUALLY\_DURING\_SUCCESSIVE\_ITERATIONS' }
  - comment - optional; type *string*
  - consider\_favorable\_effect\_due\_to\_tension\_in\_members - optional; type *boolean*
  - cutting\_patterns\_settings - optional; type *boolean*
  - deformation\_of\_failing\_members\_and\_reactivation\_enabled - optional; type *boolean*
  - displacements\_due\_to\_bourdon\_effect - optional; type *boolean*
  - divide\_results\_by\_loading\_factor - optional; type *boolean*
  - exceptional\_handling\_enabled - optional; type *boolean*
  - instability\_detection\_tolerance - optional; type *double*
  - integrate\_preliminary\_form\_finding\_enabled - optional; type *boolean*
  - iterative\_calculation\_robustness - optional; type *double*



- `iterative_method_for_nonlinear_analysis` - optional; type `static_analysis_settings_iterative_method_for_nonlinear_analysis` - type `undefined` with restriction - enum { 'DYNAMIC\_RELAXATION', 'NEWTON\_RAPHSON', 'NEWTON\_RAPHSON\_COMBINED\_WITH\_PICARD', 'NEWTON\_RAPHSON\_WITH\_CONSTANT\_STIFFNESS', 'NEWTON\_RAPHSON\_WITH\_POSTCRITICAL\_ANALYSIS', 'PICARD' }
- `loading_multiplier_factor` - optional; type `double`
- `mass_conversion_acceleration_in_direction_x` - optional; type `double`
- `mass_conversion_acceleration_in_direction_y` - optional; type `double`
- `mass_conversion_acceleration_in_direction_z` - optional; type `double`
- `mass_conversion_defined_as_acceleration` - optional; type `boolean`
- `mass_conversion_enabled` - optional; type `boolean`
- `mass_conversion_factor_in_direction_x` - optional; type `double`
- `mass_conversion_factor_in_direction_y` - optional; type `double`
- `mass_conversion_factor_in_direction_z` - optional; type `double`
- `max_number_of_iterations` - optional; type `int`
- `maximum_number_of_reactivations` - optional; type `int`
- `method_of_equation_system` - optional; type `static_analysis_settings_method_of_equation_system` - type `undefined` with restriction - enum { 'METHOD\_OF\_EQUATION\_SYSTEM\_DIRECT', 'METHOD\_OF\_EQUATION\_SYSTEM\_ITERATIVE' }
- `modify_loading_by_multiplier_factor` - optional; type `boolean`
- `nonsymmetric_direct_solver` - optional; type `boolean`
- `number_of_iterations_for_loading_prestress` - optional; type `int`
- `number_of_load_increments` - optional; type `int`
- `percentage_of_iteration` - optional; type `int`
- `plate_bending_theory` - optional; type `static_analysis_settings_plate_bending_theory` - type `undefined` with restriction - enum { 'PLATE\_BENDING\_THEORY\_KIRCHHOFF', 'PLATE\_BENDING\_THEORY\_MINDLIN' }
- `precision_of_convergence_criteria_for_nonlinear_calculation` - optional; type `double`
- `ratio_of_distance_of_cutting_lines_node_to_mesh` - optional; type `double`
- `reduction_factor_of_stiffness` - optional; type `int`
- `refer_internal_forces_to_deformed_structure` - optional; type `boolean`
- `refer_internal_forces_to_deformed_structure_for_moments` - optional; type `boolean`
- `refer_internal_forces_to_deformed_structure_for_normal_forces` - optional; type `boolean`
- `refer_internal_forces_to_deformed_structure_for_shear_forces` - optional; type `boolean`
- `relative_setting_of_time_step_for_dynamic_relaxation` - optional; type `double`
- `save_results_of_all_load_increments` - optional; type `boolean`
- `smoothness_of_boundary_lines` - optional; type `double`
- `speed_of_convergence` - optional; type `double`
- `standard_precision_and_tolerance_settings_enabled` - optional; type `boolean`
- `ignore_all_nonlinearities_enabled` - optional; type `boolean`
- `try_to_calculate_instabil_structure` - optional; type `boolean`
- `calculation_diagrams_enabled` - optional; type `boolean`
- `calculation_diagrams_list` - optional; type `array_of_static_analysis_settings_calculation_diagrams_list`
  - `static_analysis_settings_calculation_diagrams_list` - optional, unbounded; type `static_analysis_settings_calculation_diagrams_list`
    - `no` - optional; type `int`
    - `vertical_axis_result_type` - optional; type `vertical_axis_result_type` - type `undefined` with restriction - enum { 'INCREMENT', 'MAXIMUM\_DEFORMATION', 'MEMBER\_CONTACT\_FORCES', 'MEMBER\_GLOBAL\_DEFORMATIONS', 'MEMBER\_INTERNAL\_FORCES', 'MEMBER\_LOCAL\_DEFORMATIONS', 'MEMBER\_STRAINS', 'NODE\_GLOBAL\_DEFORMATIONS', 'SOLID\_BASIC\_PLASTIC\_STRAINS', 'SOLID\_BASIC\_STRESSES', 'SOLID\_BASIC\_TOTAL\_STRAINS', 'SOLID\_EQUIVALENT\_PLASTIC\_STRAINS', 'SOLID\_EQUIVALENT\_STRESSES', 'SOLID\_EQUIVALENT\_TOTAL\_STRAINS', 'SOLID\_GAS', 'SOLID\_GLOBAL\_DEFORMATIONS', 'SOLID\_PRINCIPAL\_PLASTIC\_STRAINS', 'SOLID\_PRINCIPAL\_STRESSES', 'SOLID\_PRINCIPAL\_TOTAL\_STRAINS', 'SUM\_OF\_SUPPORT\_FORCES', 'SUPPORT\_REACTIONS\_LINE\_SUPPORTS', 'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS', 'SURFACE\_BASIC\_INTERNAL\_FORCES', 'SURFACE\_BASIC\_PLASTIC\_STRAINS', 'SURFACE\_BASIC\_STRESSES', 'SURFACE\_BASIC\_TOTAL\_STRAINS', 'SURFACE\_CONTACT\_STRESSES', 'SURFACE\_DESIGN\_INTERNAL\_FORCES', 'SURFACE\_GLOBAL\_DEFORMATIONS', 'SURFACE\_LOCAL\_DEFORMATIONS', 'SURFACE\_MAXIMUM\_PLASTIC\_STRAINS', 'SURFACE\_MAXIMUM\_TOTAL\_STRAINS', 'SURFACE\_PLASTIC\_STRAINS\_BACH\_P', 'SURFACE\_PLASTIC\_STRAINS\_MISES\_P', 'SURFACE\_PLASTIC\_STRAINS\_RANKINE\_P', 'SURFACE\_PLASTIC\_STRAINS\_TRESCA\_P', 'SURFACE\_PRINCIPAL\_INTERNAL\_FORCES', 'SURFACE\_PRINCIPAL\_PLASTIC\_STRAINS', 'SURFACE\_PRINCIPAL\_STRESSES', 'SURFACE\_PRINCIPAL\_TOTAL\_STRAINS', 'SURFACE\_STRAINS\_BACH', 'SURFACE\_STRAINS\_MISES', 'SURFACE\_STRAINS\_RANKINE', 'SURFACE\_STRAINS\_TRESCA', 'SURFACE\_STRESSES\_SIGMA\_EQV\_BACH', 'SURFACE\_STRESSES\_SIGMA\_EQV\_MISES', 'SURFACE\_STRESSES\_SIGMA\_EQV\_RANKINE', 'SURFACE\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SURFACE\_STRESS\_COMPONENTS' }
    - `vertical_axis_value_type` - optional; type `vertical_axis_value_type` - type `undefined` with restriction - enum { 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_U', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_V', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Y', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Z', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_PRI', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_SEC', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MY', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MZ', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_M\_OMEGA', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_N', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_P', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_U', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_U', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VY', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VZ', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_U', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_V', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_U', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_V' }

'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_WARPING', 'MEMBER\_RESULTS\_STRAINS\_EPS\_X',  
'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XY', 'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_X', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Y',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Z',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XU',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XV',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_U',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_V',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'NODES\_GLOBAL\_DEFORMATIONS\_U',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_Z', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Z',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SOLID\_RESULTS\_GAS\_INITIAL\_PRESSURE', 'SOLID\_RESULTS\_GAS\_OVERPRESSURE',  
'SOLID\_RESULTS\_GAS\_PRESSURE', 'SOLID\_RESULTS\_GAS\_TEMPERATURE',  
'SOLID\_RESULTS\_GAS\_VOLUME', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_123',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_3',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_MISES',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Z',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_TRESCA',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XZ',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_YZ', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_123', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_3', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_MISES', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_TRESCA', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_X',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_Y', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_Z',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_XY', 'SOLID\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_YZ', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_1',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_123', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_2',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_3', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_BACH',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_MISES',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_RANKINE',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_X',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_Y', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_Z',  
'SOLID\_RESULTS\_STRESSES\_TAU\_MAX', 'SOLID\_RESULTS\_STRESSES\_TAU\_XY',  
'SOLID\_RESULTS\_STRESSES\_TAU\_XZ', 'SOLID\_RESULTS\_STRESSES\_TAU\_YZ',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Z',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VY',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_SIGMA\_Z',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_YZ',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NC\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NX\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NY\_D',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U',

'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_PLUS',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_M',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_BETA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_MT\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_M',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_ALPHA\_MINUS', 'SURFACE\_RESULTS\_STRAINS\_ALPHA\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_BACH', 'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_M',  
'SURFACE\_RESULTS\_STRESSES\_ALFA\_MINUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_M', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_B', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_B',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_M',

'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_TAU\_MAX',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_B', 'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_M',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_YZ', 'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_X',  
'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_Y', 'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_Z' }

- vertical\_axis\_object - optional; type *int*
- vertical\_axis\_node - optional; type *int*
- horizontal\_axis\_result\_type - optional; type *horizontal\_axis\_result\_type* - type *undefined* with restriction - enum { 'INCREMENT', 'MAXIMUM\_DEFORMATION', 'MEMBER\_CONTACT\_FORCES', 'MEMBER\_GLOBAL\_DEFORMATIONS', 'MEMBER\_INTERNAL\_FORCES', 'MEMBER\_LOCAL\_DEFORMATIONS', 'MEMBER\_STRAINS', 'NODE\_GLOBAL\_DEFORMATIONS', 'SOLID\_BASIC\_PLASTIC\_STRAINS', 'SOLID\_BASIC\_STRESSES', 'SOLID\_BASIC\_TOTAL\_STRAINS', 'SOLID\_EQUIVALENT\_PLASTIC\_STRAINS', 'SOLID\_EQUIVALENT\_STRESSES', 'SOLID\_EQUIVALENT\_TOTAL\_STRAINS', 'SOLID\_GAS', 'SOLID\_GLOBAL\_DEFORMATIONS', 'SOLID\_PRINCIPAL\_PLASTIC\_STRAINS', 'SOLID\_PRINCIPAL\_STRESSES', 'SOLID\_PRINCIPAL\_TOTAL\_STRAINS', 'SUM\_OF\_SUPPORT\_FORCES', 'SUPPORT\_REACTIONS\_LINE\_SUPPORTS', 'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS', 'SURFACE\_BASIC\_INTERNAL\_FORCES', 'SURFACE\_BASIC\_PLASTIC\_STRAINS', 'SURFACE\_BASIC\_STRESSES', 'SURFACE\_BASIC\_TOTAL\_STRAINS', 'SURFACE\_CONTACT\_STRESSES', 'SURFACE\_DESIGN\_INTERNAL\_FORCES', 'SURFACE\_GLOBAL\_DEFORMATIONS', 'SURFACE\_LOCAL\_DEFORMATIONS', 'SURFACE\_MAXIMUM\_PLASTIC\_STRAINS', 'SURFACE\_MAXIMUM\_TOTAL\_STRAINS', 'SURFACE\_PLASTIC\_STRAINS\_BACH\_P', 'SURFACE\_PLASTIC\_STRAINS\_MISES\_P', 'SURFACE\_PLASTIC\_STRAINS\_RANKINE\_P', 'SURFACE\_PLASTIC\_STRAINS\_TRESCA\_P', 'SURFACE\_PRINCIPAL\_INTERNAL\_FORCES', 'SURFACE\_PRINCIPAL\_PLASTIC\_STRAINS', 'SURFACE\_PRINCIPAL\_STRESSES', 'SURFACE\_PRINCIPAL\_TOTAL\_STRAINS', 'SURFACE\_STRAINS\_BACH', 'SURFACE\_STRAINS\_MISES', 'SURFACE\_STRAINS\_RANKINE', 'SURFACE\_STRAINS\_TRESCA', 'SURFACE\_STRESSES\_SIGMA\_EQV\_BACH', 'SURFACE\_STRESSES\_SIGMA\_EQV\_MISES', 'SURFACE\_STRESSES\_SIGMA\_EQV\_RANKINE', 'SURFACE\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SURFACE\_STRESS\_COMPONENTS' }
- horizontal\_axis\_value\_type - optional; type *horizontal\_axis\_value\_type* - type *undefined* with restriction - enum { 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_U', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_V', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Y', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Z', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_PRI', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_SEC', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MY', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MZ', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_M\_OMEGA', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_N', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_P', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_U', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_U', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VY', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VZ', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_U', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_V', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_U', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_V', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z', 'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_WARPING', 'MEMBER\_RESULTS\_STRAINS\_EPS\_X', 'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XY', 'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XZ', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_X', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Y', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Z', 'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XU', 'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XV', 'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_U', 'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_V', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'NODES\_GLOBAL\_DEFORMATIONS\_U', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_Y', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_Z', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_X', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Z', 'SOLID\_RESULTS\_GAS\_INITIAL\_PRESSURE', 'SOLID\_RESULTS\_GAS\_OVERPRESSURE', 'SOLID\_RESULTS\_GAS\_PRESSURE', 'SOLID\_RESULTS\_GAS\_TEMPERATURE', 'SOLID\_RESULTS\_GAS\_VOLUME', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_1', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_123', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_2', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_3', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_BACH', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_MISES', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Z', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_RANKINE', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_TRESCA', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XZ', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_YZ', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_1', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_123', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_2', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_3', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_BACH', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_MISES', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_RANKINE' }

'SOLID\_RESULTS\_STRAINS\_EPSILON\_TRESCA', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_X',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_Y', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_Z',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_XY', 'SOLID\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_YZ', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_1',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_123', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_2',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_3', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_BACH',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_MISES',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_RANKINE',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_X',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_Y', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_Z',  
'SOLID\_RESULTS\_STRESSES\_TAU\_MAX', 'SOLID\_RESULTS\_STRESSES\_TAU\_XY',  
'SOLID\_RESULTS\_STRESSES\_TAU\_XZ', 'SOLID\_RESULTS\_STRESSES\_TAU\_YZ',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Z',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VY',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_SIGMA\_Z',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_YZ',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NC\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NX\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NY\_D',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_PLUS',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_M',

```
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_BETA_B',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_M1',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_M2',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_MT_MAX_B',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_N1',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_N2',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_V_MAX_B',
'SURFACE_RESULTS_PRINCIPAL_INTERNAL_FORCES_V_MAX_M',
'SURFACE_RESULTS_STRAINS_ABS_EPSILON_MAX',
'SURFACE_RESULTS_STRAINS_ABS_EPSILON_MAX_MINUS',
'SURFACE_RESULTS_STRAINS_ABS_EPSILON_MAX_PLUS',
'SURFACE_RESULTS_STRAINS_ALPHA_MINUS', 'SURFACE_RESULTS_STRAINS_ALPHA_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_1_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_1_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_2_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_2_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_BACH', 'SURFACE_RESULTS_STRAINS_EPSILON_MAX',
'SURFACE_RESULTS_STRAINS_EPSILON_MAX_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_MAX_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_MIN',
'SURFACE_RESULTS_STRAINS_EPSILON_MINUS_BACH',
'SURFACE_RESULTS_STRAINS_EPSILON_MINUS_MISES',
'SURFACE_RESULTS_STRAINS_EPSILON_MINUS_RANKINE',
'SURFACE_RESULTS_STRAINS_EPSILON_MINUS_TRESCA',
'SURFACE_RESULTS_STRAINS_EPSILON_MIN_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_MIN_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_MISES',
'SURFACE_RESULTS_STRAINS_EPSILON_PLUS_BACH',
'SURFACE_RESULTS_STRAINS_EPSILON_PLUS_MISES',
'SURFACE_RESULTS_STRAINS_EPSILON_PLUS_RANKINE',
'SURFACE_RESULTS_STRAINS_EPSILON_PLUS_TRESCA',
'SURFACE_RESULTS_STRAINS_EPSILON_RANKINE',
'SURFACE_RESULTS_STRAINS_EPSILON_TRESCA',
'SURFACE_RESULTS_STRAINS_EPSILON_X_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_X_PLUS',
'SURFACE_RESULTS_STRAINS_EPSILON_Y_MINUS',
'SURFACE_RESULTS_STRAINS_EPSILON_Y_PLUS',
'SURFACE_RESULTS_STRAINS_GAMMA_XY_MINUS',
'SURFACE_RESULTS_STRAINS_GAMMA_XY_PLUS', 'SURFACE_RESULTS_STRESSES_ALFA_M',
'SURFACE_RESULTS_STRESSES_ALFA_MINUS', 'SURFACE_RESULTS_STRESSES_ALFA_PLUS',
'SURFACE_RESULTS_STRESSES_SIGMA_1_M', 'SURFACE_RESULTS_STRESSES_SIGMA_1_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_1_PLUS', 'SURFACE_RESULTS_STRESSES_SIGMA_2_M',
'SURFACE_RESULTS_STRESSES_SIGMA_2_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_2_PLUS',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MAX_BACH',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MAX_MISES',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MAX_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MAX_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MINUS_BACH',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MINUS_MISES',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MINUS_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_MINUS_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_BACH',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_MISES',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_BACH',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_MISES',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_X_B', 'SURFACE_RESULTS_STRESSES_SIGMA_X_M',
'SURFACE_RESULTS_STRESSES_SIGMA_X_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_X_PLUS', 'SURFACE_RESULTS_STRESSES_SIGMA_Y_B',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_M',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_PLUS', 'SURFACE_RESULTS_STRESSES_TAU_MAX',
'SURFACE_RESULTS_STRESSES_TAU_XY_B', 'SURFACE_RESULTS_STRESSES_TAU_XY_M',
'SURFACE_RESULTS_STRESSES_TAU_XY_MINUS',
'SURFACE_RESULTS_STRESSES_TAU_XY_PLUS', 'SURFACE_RESULTS_STRESSES_TAU_XZ',
'SURFACE_RESULTS_STRESSES_TAU_YZ', 'TYPE_SUM_OF_SUPPORT_FORCES_X',
'TYPE_SUM_OF_SUPPORT_FORCES_Y', 'TYPE_SUM_OF_SUPPORT_FORCES_Z'}
```

- horizontal\_axis\_object - optional; type *int*
- horizontal\_axis\_node - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 111. get\_structure\_modification

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_structure\_modification

**Input:** get\_structure\_modification\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_structure\_modification*

- no type *int*

**Output:** get\_structure\_modification\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_structure\_modificationResponse*

- value type *structure\_modification*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to - optional; type *string*
  - comment - optional; type *string*
  - modify\_stiffnesses\_gamma\_m - optional; type *boolean*
  - modify\_stiffnesses\_materials - optional; type *boolean*
  - modify\_stiffnesses\_sections - optional; type *boolean*
  - modify\_stiffnesses\_members - optional; type *boolean*

- `modify_stiffnesses_surfaces` - optional; type *boolean*
- `modify_stiffnesses_member_hinges` - optional; type *boolean*
- `modify_stiffnesses_line_hinges` - optional; type *boolean*
- `modify_stiffnesses_nodal_supports` - optional; type *boolean*
- `modify_stiffnesses_line_supports` - optional; type *boolean*
- `modify_stiffnesses_member_supports` - optional; type *boolean*
- `modify_stiffnesses_surface_supports` - optional; type *boolean*
- `modify_stiffness_member_reinforcement` - optional; type *boolean*
- `modify_stiffness_surface_reinforcement` - optional; type *boolean*
- `modify_stiffness_timber_members_due_moisture_class` - optional; type *boolean*
- `nonlinearities_disabled_material_nonlinearity_models` - optional; type *boolean*
- `nonlinearities_disabled_material_temperature_nonlinearities` - optional; type *boolean*
- `nonlinearities_disabled_line_hinges` - optional; type *boolean*
- `nonlinearities_disabled_member_types` - optional; type *boolean*
- `nonlinearities_disabled_member_hinges` - optional; type *boolean*
- `nonlinearities_disabled_member_nonlinearities` - optional; type *boolean*
- `nonlinearities_disabled_solid_types_contact_or_surfaces_contact` - optional; type *boolean*
- `nonlinearities_disabled_nodal_supports` - optional; type *boolean*
- `nonlinearities_disabled_line_supports` - optional; type *boolean*
- `nonlinearities_disabled_member_supports` - optional; type *boolean*
- `nonlinearities_disabled_surface_supports` - optional; type *boolean*
- `modify_stiffnesses_material_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_material\_table*
  - `structure_modification_modify_stiffnesses_material_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_material\_table*
    - `no` - optional; type *int*
    - `material_name` - optional; type *int*
    - `modification_type` - optional; type *modification\_type* - type *undefined* with restriction - enum { 'DIVISION\_FACTOR', 'MULTIPLY\_FACTOR' }
    - `E_and_G` - optional; type *double*
    - `comment` - optional; type *string*
- `modify_stiffnesses_section_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_section\_table*
  - `structure_modification_modify_stiffnesses_section_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_section\_table*
    - `no` - optional; type *int*
    - `section_name` - optional; type *string*
    - `A` - optional; type *double*
    - `A_y` - optional; type *double*
    - `A_z` - optional; type *double*
    - `J` - optional; type *double*
    - `I_y` - optional; type *double*
    - `I_z` - optional; type *double*
- `modify_stiffnesses_member_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_member\_table*
  - `structure_modification_modify_stiffnesses_member_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_member\_table*
    - `no` - optional; type *int*
    - `member_modification` - optional; type *int*
    - `members` - optional; type *array\_of\_int*
    - `comment` - optional; type *string*
- `modify_stiffnesses_surface_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_surface\_table*
  - `structure_modification_modify_stiffnesses_surface_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_surface\_table*
    - `no` - optional; type *int*
    - `surface_modification` - optional; type *int*
    - `surfaces` - optional; type *array\_of\_int*
    - `comment` - optional; type *string*
- `modify_stiffnesses_member_hinges_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_member\_hinges\_table*
  - `structure_modification_modify_stiffnesses_member_hinges_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_member\_hinges\_table*
    - `no` - optional; type *int*
    - `member_side` - optional; type *string*
    - `C_u_x` - optional; type *double*
    - `C_u_y` - optional; type *double*
    - `C_u_z` - optional; type *double*
    - `C_phi_x` - optional; type *double*
    - `C_phi_y` - optional; type *double*
    - `C_phi_z` - optional; type *double*
- `modify_stiffnesses_line_hinges_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_line\_hinges\_table*
  - `structure_modification_modify_stiffnesses_line_hinges_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_line\_hinges\_table*
    - `no` - optional; type *int*
    - `C_u_x` - optional; type *double*
    - `C_u_y` - optional; type *double*
    - `C_u_z` - optional; type *double*
    - `C_phi_x` - optional; type *double*
- `modify_stiffnesses_nodal_supports_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_nodal\_supports\_table*
  - `structure_modification_modify_stiffnesses_nodal_supports_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_nodal\_supports\_table*
    - `no` - optional; type *int*
    - `C_u_X` - optional; type *double*
    - `C_u_Y` - optional; type *double*
    - `C_u_Z` - optional; type *double*
    - `C_phi_X` - optional; type *double*
    - `C_phi_Y` - optional; type *double*
    - `C_phi_Z` - optional; type *double*
- `modify_stiffnesses_line_supports_table` - optional; type  
*array\_of\_structure\_modification\_modify\_stiffnesses\_line\_supports\_table*
  - `structure_modification_modify_stiffnesses_line_supports_table` - optional, unbounded; type  
*structure\_modification\_modify\_stiffnesses\_line\_supports\_table*
    - `no` - optional; type *int*
    - `C_u_X` - optional; type *double*

- C\_u\_Y - optional; type *double*
- C\_u\_Z - optional; type *double*
- C\_phi\_X - optional; type *double*
- C\_phi\_Y - optional; type *double*
- C\_phi\_Z - optional; type *double*
- modify\_stiffnesses\_member\_supports\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_member\_supports\_table*
  - structure\_modification\_modify\_stiffnesses\_member\_supports\_table - optional, unbounded; type *structure\_modification\_modify\_stiffnesses\_member\_supports\_table*
    - no - optional; type *int*
    - C\_u\_x - optional; type *double*
    - C\_u\_y - optional; type *double*
    - C\_u\_z - optional; type *double*
    - C\_s\_x - optional; type *double*
    - C\_s\_y - optional; type *double*
    - C\_s\_z - optional; type *double*
    - C\_phi\_x - optional; type *double*
- modify\_stiffnesses\_surface\_supports\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_surface\_supports\_table*
  - structure\_modification\_modify\_stiffnesses\_surface\_supports\_table - optional, unbounded; type *structure\_modification\_modify\_stiffnesses\_surface\_supports\_table*
    - no - optional; type *int*
    - C\_u\_X - optional; type *double*
    - C\_u\_Y - optional; type *double*
    - C\_u\_Z - optional; type *double*
    - C\_v\_xz - optional; type *double*
    - C\_v\_yz - optional; type *double*
- deactivate\_members\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_members - optional; type *int*
- deactivate\_surfaces\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_surfaces - optional; type *int*
- deactivate\_solids\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_solids - optional; type *int*
- deactivate\_support\_on\_nodes\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_support\_on\_nodes - optional; type *int*
- deactivate\_support\_on\_lines\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_support\_on\_lines - optional; type *int*
- deactivate\_support\_on\_members\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_support\_on\_members - optional; type *int*
- deactivate\_support\_on\_surfaces\_enabled - optional; type *boolean*
- object\_selection\_for\_deactivate\_support\_on\_surfaces - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 112. get\_surface

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface

**Input:** get\_surface\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface*

- no type *int*

**Output:** get\_surface\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surfaceResponse*

- value type *surface*
  - no type *int*
  - geometry - optional; type *surface\_geometry* - type *undefined* with restriction - enum { 'GEOMETRY\_NURBS', 'GEOMETRY\_PIPE', 'GEOMETRY\_PLANE', 'GEOMETRY\_QUADRANGLE', 'GEOMETRY\_ROTATED' }
  - type - optional; type *surface\_type* - type *undefined* with restriction - enum { 'TYPE\_GROUNDWATER', 'TYPE\_LOAD\_TRANSFER', 'TYPE\_MEMBRANE', 'TYPE\_RIGID', 'TYPE\_STANDARD', 'TYPE\_WITHOUT\_MEMBRANE\_TENSION', 'TYPE\_WITHOUT\_THICKNESS' }
  - boundary\_lines - optional; type *array\_of\_int*
  - thickness - optional; type *int*
  - material - optional; type *int*
  - analytical\_area - optional; type *double*
  - analytical\_volume - optional; type *double*
  - analytical\_mass - optional; type *double*
  - analytical\_center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - analytical\_center\_of\_gravity\_x - optional; type *double*
  - analytical\_center\_of\_gravity\_y - optional; type *double*
  - analytical\_center\_of\_gravity\_z - optional; type *double*
  - area - optional; type *double*
  - volume - optional; type *double*
  - mass - optional; type *double*
  - center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_of\_gravity\_x - optional; type *double*
  - center\_of\_gravity\_y - optional; type *double*
  - center\_of\_gravity\_z - optional; type *double*
  - position - optional; type *string*
  - position\_short - optional; type *string*
  - grid\_enabled - optional; type *boolean*
  - is\_deactivated\_for\_calculation - optional; type *boolean*
  - comment - optional; type *string*
  - design\_properties\_via\_surface - optional; type *boolean*
  - load\_transfer\_direction - optional; type *surface\_load\_transfer\_direction* - type *undefined* with restriction - enum { 'LOAD\_TRANSFER\_DIRECTION\_IN\_BOTH', 'LOAD\_TRANSFER\_DIRECTION\_IN\_X', 'LOAD\_TRANSFER\_DIRECTION\_IN\_Y' }



- is\_surface\_weight\_enabled - optional; type *boolean*
- surface\_weight - optional; type *double*
- consider\_member\_eccentricity - optional; type *boolean*
- consider\_section\_distribution - optional; type *boolean*
- excluded\_members - optional; type *array\_of\_int*
- excluded\_parallel\_to\_members - optional; type *array\_of\_int*
- excluded\_lines - optional; type *array\_of\_int*
- excluded\_parallel\_to\_lines - optional; type *array\_of\_int*
- loaded\_members - optional; type *array\_of\_int*
- loaded\_lines - optional; type *array\_of\_int*
- nurbs\_control\_point\_count\_in\_direction\_u - optional; type *int*
- nurbs\_control\_point\_count\_in\_direction\_v - optional; type *int*
- nurbs\_order\_in\_direction\_u - optional; type *int*
- nurbs\_order\_in\_direction\_v - optional; type *int*
- nurbs\_control\_points - optional; type *array\_of\_surface\_nurbs\_control\_points\_rows*
  - surface\_nurbs\_control\_points\_rows - optional, unbounded; type *surface\_nurbs\_control\_points\_row*
    - surface\_nurbs\_control\_points\_row - optional, unbounded; type *surface\_nurbs\_control\_points*
      - u - optional; type *int*
      - v - optional; type *int*
      - global\_coordinates - optional; type *vector\_3d*
        - x type *double*
        - y type *double*
        - z type *double*
      - coordinates - optional; type *vector\_3d*
        - x type *double*
        - y type *double*
        - z type *double*
      - weight type *double*
- quadrangle\_corner\_nodes - optional; type *array\_of\_int*
- quadrangle\_corner\_node\_1 - optional; type *int*
- quadrangle\_corner\_node\_2 - optional; type *int*
- quadrangle\_corner\_node\_3 - optional; type *int*
- quadrangle\_corner\_node\_4 - optional; type *int*
- pipe\_radius - optional; type *double*
- pipe\_center\_line - optional; type *int*
- pipe\_generated\_lines - optional; type *array\_of\_int*
- has\_line\_hinges - optional; type *boolean*
- support - optional; type *int*
- eccentricity - optional; type *int*
- mesh\_refinement - optional; type *int*
- meshing\_type - optional; type *surface\_meshing\_type* - type *undefined* with restriction - enum { 'MESHING\_TYPE\_FREE', 'MESHING\_TYPE\_MAPPED', 'MESHING\_TYPE\_USE\_GLOBAL\_SETTINGS' }
- input\_axes\_rotation\_specification\_type - optional; type *surface\_input\_axes\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'INPUT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_ANGULAR\_ROTATION', 'INPUT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_DIRECT\_TO\_POINT', 'INPUT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_PARALLEL\_TO\_COORDINATE\_SYSTEM', 'INPUT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_PARALLEL\_TO\_LINES' }
- input\_axes\_angular\_rotation - optional; type *double*
- input\_axes\_axis - optional; type *surface\_input\_axes\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y' }
- input\_axes\_lines - optional; type *array\_of\_int*
- input\_axes\_point\_1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- input\_axes\_point\_1\_x - optional; type *double*
- input\_axes\_point\_1\_y - optional; type *double*
- input\_axes\_point\_1\_z - optional; type *double*
- input\_axes\_point\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- input\_axes\_point\_2\_x - optional; type *double*
- input\_axes\_point\_2\_y - optional; type *double*
- input\_axes\_point\_2\_z - optional; type *double*
- input\_axes\_coordinate\_system - optional; type *int*
- result\_axes\_rotation\_specification\_type - optional; type *surface\_result\_axes\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'RESULT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_IDENTICAL\_TO\_INPUT\_AXES' }
- reversed\_normal - optional; type *boolean*
- grid\_type - optional; type *surface\_grid\_type* - type *undefined* with restriction - enum { 'GRID\_TYPE\_CARTESIAN', 'GRID\_TYPE\_POLAR' }
- grid\_origin - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- grid\_origin\_x - optional; type *double*
- grid\_origin\_y - optional; type *double*
- grid\_origin\_z - optional; type *double*
- grid\_point\_count\_negative\_x - optional; type *int*
- grid\_point\_count\_positive\_x - optional; type *int*
- grid\_point\_count\_negative\_y - optional; type *int*
- grid\_point\_count\_positive\_y - optional; type *int*
- grid\_numbering\_increment - optional; type *int*
- grid\_point\_count\_r - optional; type *int*
- grid\_distance\_x - optional; type *double*
- grid\_distance\_y - optional; type *double*
- grid\_distance\_r - optional; type *double*
- grid\_rotation\_alpha - optional; type *double*
- grid\_rotation\_beta - optional; type *double*
- grid\_angle\_gamma - optional; type *double*
- grid\_adapt\_automatically - optional; type *boolean*
- auto\_detection\_of\_integrated\_objects - optional; type *boolean*
- integrated\_nodes - optional; type *array\_of\_int*
- integrated\_lines - optional; type *array\_of\_int*
- integrated\_openings - optional; type *array\_of\_int*

- has\_integrated\_objects - optional; type *boolean*
- has\_input\_axes\_rotation - optional; type *boolean*
- has\_result\_axes\_rotation - optional; type *boolean*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- surface\_reinforcements - optional; type *array\_of\_int*
- is\_user\_defined\_concrete\_cover\_enabled - optional; type *boolean*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- user\_defined\_concrete\_cover\_top - optional; type *double*
- user\_defined\_concrete\_cover\_bottom - optional; type *double*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- reinforcement\_direction\_top - optional; type *int*
- reinforcement\_direction\_bottom - optional; type *int*
- deflection\_check\_surface\_type - optional; type *surface\_deflection\_check\_surface\_type* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_SURFACE\_TYPE\_CANTILEVER', 'DEFLECTION\_CHECK\_SURFACE\_TYPE\_DOUBLE\_SUPPORTED' }
- deflection\_check\_displacement\_reference - optional; type *surface\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_USER\_DEFINED\_REFERENCE\_PLANE', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_PARALLEL\_SURFACE', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_UNDEFORMED\_SYSTEM' }
- deflection\_check\_reference\_length\_z - optional; type *double*
- deflection\_check\_reference\_length\_z\_definition\_type - optional; type *surface\_deflection\_check\_reference\_length\_z\_definition\_type* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_BY\_MAXIMUM\_BOUNDARY\_LINE', 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_BY\_MINIMUM\_BOUNDARY\_LINE', 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_MANUALLY' }
- deflection\_check\_reference\_plane\_point\_1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_1\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_1\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_1\_z - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_2\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2\_z - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_3\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3\_z - optional; type *double*
- surface\_concrete\_design\_uls\_configuration - optional; type *int*
- surface\_concrete\_design\_sls\_configuration - optional; type *int*
- surface\_concrete\_design\_fr\_configuration - optional; type *int*
- surface\_concrete\_design\_seismic\_configuration - optional; type *int*
- rotated\_boundary\_line - optional; type *int*
- rotated\_angle\_of\_rotation - optional; type *double*
- rotated\_point\_p - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- rotated\_point\_p\_x - optional; type *double*
- rotated\_point\_p\_y - optional; type *double*
- rotated\_point\_p\_z - optional; type *double*
- rotated\_point\_r - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- rotated\_point\_r\_x - optional; type *double*
- rotated\_point\_r\_y - optional; type *double*
- rotated\_point\_r\_z - optional; type *double*
- rotated\_generated\_lines - optional; type *array\_of\_int*
- stress\_analysis\_configuration - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 113. **get\_surface\_eccentricity**

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_surface\\_eccentricity](http://localhost:8082/get_surface_eccentricity)

**Input:** *get\_surface\_eccentricity\_request* (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_eccentricity*

- no type *int*

**Output:** *get\_surface\_eccentricity\_response* (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_eccentricityResponse*

- value type *surface\_eccentricity*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*

- offset - optional; type *double*
- assigned\_to\_surfaces - optional; type *array\_of\_int*
- thickness\_alignment - optional; type *surface\_eccentricity\_thickness\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
- transverse\_offset\_reference\_type - optional; type *surface\_eccentricity\_transverse\_offset\_reference\_type* - type *undefined* with restriction - enum { 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_MEMBER\_SECTION', 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_SURFACE\_THICKNESS', 'TRANSVERSE\_OFFSET\_TYPE\_NONE' }
- transverse\_offset\_reference\_member - optional; type *int*
- transverse\_offset\_reference\_surface - optional; type *int*
- transverse\_offset\_member\_reference\_node - optional; type *int*
- transverse\_offset\_surface\_reference\_node - optional; type *int*
- transverse\_offset\_alignment - optional; type *surface\_eccentricity\_transverse\_offset\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 114. get\_surface\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface\_imperfection

**Input:** get\_surface\_imperfection\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_imperfection*

- no type *int*
- imperfection\_case\_no type *int*

**Output:** get\_surface\_imperfection\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_imperfectionResponse*

- value type *surface\_imperfection*
  - no type *int*
  - definition\_type - optional; type *surface\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'ABSOLUTE', 'RELATIVE' }
  - imperfection\_case - optional; type *int*
  - imperfection\_direction - optional; type *surface\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE' }
  - initial\_bow - optional; type *double*
  - initial\_bow\_relative - optional; type *double*
  - parameters - optional; type *array\_of\_int*
  - reference\_length - optional; type *double*
  - surfaces - optional; type *array\_of\_int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

#### 115. get\_surface\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface\_load

**Input:** get\_surface\_load\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_load*

- no type *int*
- load\_case\_no type *int*

**Output:** get\_surface\_load\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_loadResponse*

- value type *surface\_load*
  - no type *int*
  - load\_type - optional; type *surface\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_MASS', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY\_MOTION', 'LOAD\_TYPE\_TEMPERATURE' }
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - load\_distribution - optional; type *surface\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_X', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Y', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Z', 'LOAD\_DISTRIBUTION\_RADIAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }
  - coordinate\_system - optional; type *string*
  - load\_direction - optional; type *surface\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
  - uniform\_magnitude - optional; type *double*
  - magnitude\_1 - optional; type *double*
  - magnitude\_2 - optional; type *double*
  - magnitude\_3 - optional; type *double*
  - uniform\_magnitude\_t\_c - optional; type *double*
  - magnitude\_t\_c\_1 - optional; type *double*
  - magnitude\_t\_c\_2 - optional; type *double*
  - magnitude\_t\_c\_3 - optional; type *double*
  - uniform\_magnitude\_delta\_t - optional; type *double*
  - magnitude\_delta\_t\_1 - optional; type *double*

- magnitude\_delta\_t\_2 - optional; type *double*
- magnitude\_delta\_t\_3 - optional; type *double*
- magnitude\_axial\_strain\_x - optional; type *double*
- magnitude\_axial\_strain\_y - optional; type *double*
- magnitude\_axial\_strain\_1x - optional; type *double*
- magnitude\_axial\_strain\_1y - optional; type *double*
- magnitude\_axial\_strain\_2x - optional; type *double*
- magnitude\_axial\_strain\_2y - optional; type *double*
- magnitude\_axial\_strain\_3x - optional; type *double*
- magnitude\_axial\_strain\_3y - optional; type *double*
- angular\_velocity - optional; type *double*
- angular\_acceleration - optional; type *double*
- node\_1 - optional; type *int*
- node\_2 - optional; type *int*
- node\_3 - optional; type *int*
- axis\_definition\_type - optional; type *surface\_load\_axis\_definition\_type* - type *undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *surface\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *surface\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- varying\_load\_parameters - optional; type *array\_of\_surface\_load\_varying\_load\_parameters*
  - surface\_load\_varying\_load\_parameters - optional, unbounded; type *surface\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- form\_finding\_definition - optional; type *surface\_load\_form\_finding\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_DEFINITION\_FORCE', 'FORM\_FINDING\_DEFINITION\_STRESS' }
- magnitude\_force\_x - optional; type *double*
- magnitude\_force\_y - optional; type *double*
- magnitude\_force\_u - optional; type *double*
- magnitude\_force\_v - optional; type *double*
- magnitude\_force\_r - optional; type *double*
- magnitude\_force\_t - optional; type *double*
- magnitude\_stress\_x - optional; type *double*
- magnitude\_stress\_y - optional; type *double*
- magnitude\_stress\_u - optional; type *double*
- magnitude\_stress\_v - optional; type *double*
- magnitude\_stress\_r - optional; type *double*
- magnitude\_stress\_t - optional; type *double*
- form\_finding\_calculation\_method - optional; type *surface\_load\_form\_finding\_calculation\_method* - type *undefined* with restriction - enum { 'FORM\_FINDING\_CALCULATION\_METHOD\_PROJECTION', 'FORM\_FINDING\_CALCULATION\_METHOD\_STANDARD' }
- individual\_mass\_components - optional; type *boolean*
- magnitude\_mass\_global - optional; type *double*
- magnitude\_mass\_y - optional; type *double*
- magnitude\_mass\_z - optional; type *double*
- magnitude\_mass\_x - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 116. **get\_surface\_mesh\_refinement**

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_surface\\_mesh\\_refinement](http://localhost:8082/get_surface_mesh_refinement)

**Input:** *get\_surface\_mesh\_refinement\_request* (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_mesh\_refinement*

- no type *int*

**Output:** *get\_surface\_mesh\_refinement\_response* (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_mesh\_refinementResponse*

- value type *surface\_mesh\_refinement*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - surfaces - optional; type *array\_of\_int*
  - target\_length - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*

- metadata\_for\_export\_import - optional; type *string*

### 117. [get\\_surface\\_results\\_adjustment](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_surface\\_results\\_adjustment](http://localhost:8082/get_surface_results_adjustment)

**Input:** [get\\_surface\\_results\\_adjustment\\_request](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_surface\\_results\\_adjustment](#)

- no type *int*

**Output:** [get\\_surface\\_results\\_adjustment\\_response](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_surface\\_results\\_adjustmentResponse](#)

- value type [surface\\_results\\_adjustment](#)
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - surfaces - optional; type *array\_of\_int*
  - shape - optional; type [surface\\_results\\_adjustment\\_shape](#) - type *undefined* with restriction - enum { 'SHAPE\_CIRCLE', 'SHAPE\_ELLIPSE', 'SHAPE\_RECTANGLE' }
  - dimension\_1 - optional; type *double*
  - dimension\_2 - optional; type *double*
  - angular\_rotation - optional; type *double*
  - center\_position - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_position\_x - optional; type *double*
  - center\_position\_y - optional; type *double*
  - center\_position\_z - optional; type *double*
  - adjustment\_type\_in\_direction\_u - optional; type [surface\\_results\\_adjustment\\_adjustment\\_type\\_in\\_direction\\_u](#) - type *undefined* with restriction - enum { 'AVERAGING\_OF\_MX\_MXY\_VX\_NX\_NXY', 'AVERAGING\_OF\_MY\_MXY\_VY\_NY\_NXY', 'NONE', 'USER\_DEFINED', 'ZERO' }
  - adjustment\_type\_in\_direction\_v - optional; type [surface\\_results\\_adjustment\\_adjustment\\_type\\_in\\_direction\\_v](#) - type *undefined* with restriction - enum { 'AVERAGING\_OF\_MX\_MXY\_VX\_NX\_NXY', 'AVERAGING\_OF\_MY\_MXY\_VY\_NY\_NXY', 'NONE', 'USER\_DEFINED', 'ZERO' }
  - projection\_in\_direction\_type - optional; type [surface\\_results\\_adjustment\\_projection\\_in\\_direction\\_type](#) - type *undefined* with restriction - enum { 'GLOBAL\_IN\_X', 'GLOBAL\_IN\_Y', 'GLOBAL\_IN\_Z', 'PERPENDICULAR', 'VECTOR' }
  - vector\_of\_projection\_in\_direction\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - vector\_of\_projection\_in\_direction\_coordinates\_x - optional; type *double*
  - vector\_of\_projection\_in\_direction\_coordinates\_y - optional; type *double*
  - vector\_of\_projection\_in\_direction\_coordinates\_z - optional; type *double*
  - comment - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 118. [get\\_surface\\_set](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_surface\\_set](http://localhost:8082/get_surface_set)

**Input:** [get\\_surface\\_set\\_request](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_surface\\_set](#)

- no type *int*

**Output:** [get\\_surface\\_set\\_response](#) (soap:body, use = literal) [Source code](#)

**parameters** type [get\\_surface\\_setResponse](#)

- value type [surface\\_set](#)
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - set\_type - optional; type [surface\\_set\\_set\\_type](#) - type *undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
  - surfaces - optional; type *array\_of\_int*
  - surface\_area - optional; type *double*
  - volume - optional; type *double*
  - mass - optional; type *double*
  - center\_of\_gravity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_of\_gravity\_x - optional; type *double*
  - center\_of\_gravity\_y - optional; type *double*
  - center\_of\_gravity\_z - optional; type *double*
  - position - optional; type *string*
  - position\_short - optional; type *string*
  - stress\_analysis\_configuration - optional; type *int*
  - timber\_service\_class - optional; type *int*
  - timber\_moisture\_class - optional; type *int*
  - timber\_service\_conditions - optional; type *int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - surface\_concrete\_design\_uls\_configuration - optional; type *int*
  - surface\_concrete\_design\_sls\_configuration - optional; type *int*
  - surface\_concrete\_design\_fr\_configuration - optional; type *int*
  - surface\_concrete\_design\_seismic\_configuration - optional; type *int*

- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

#### 119. get\_surface\_set\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface\_set\_imperfection

**Input:** get\_surface\_set\_imperfection\_request (soap:body, use = literal) [Source code](#)

- parameters** type *get\_surface\_set\_imperfection*
- no type *int*
  - imperfection\_case\_no type *int*

**Output:** get\_surface\_set\_imperfection\_response (soap:body, use = literal) [Source code](#)

- parameters** type *get\_surface\_set\_imperfectionResponse*
- value type *surface\_set\_imperfection*
    - no type *int*
    - definition\_type - optional; type *surface\_set\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'ABSOLUTE', 'RELATIVE' }
    - imperfection\_case - optional; type *int*
    - imperfection\_direction - optional; type *surface\_set\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE' }
    - initial\_bow - optional; type *double*
    - initial\_bow\_relative - optional; type *double*
    - parameters - optional; type *array\_of\_int*
    - reference\_length - optional; type *double*
    - surface\_sets - optional; type *array\_of\_int*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

#### 120. get\_surface\_set\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface\_set\_load

**Input:** get\_surface\_set\_load\_request (soap:body, use = literal) [Source code](#)

- parameters** type *get\_surface\_set\_load*
- no type *int*
  - load\_case\_no type *int*

**Output:** get\_surface\_set\_load\_response (soap:body, use = literal) [Source code](#)

- parameters** type *get\_surface\_set\_loadResponse*
- value type *surface\_set\_load*
    - no type *int*
    - load\_type - optional; type *surface\_set\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_MASS', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY\_MOTION', 'LOAD\_TYPE\_TEMPERATURE' }
    - surface\_sets - optional; type *array\_of\_int*
    - load\_case - optional; type *int*
    - coordinate\_system - optional; type *string*
    - load\_distribution - optional; type *surface\_set\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_X', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Y', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Z', 'LOAD\_DISTRIBUTION\_RADIAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }
    - load\_direction - optional; type *surface\_set\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
    - individual\_mass\_components - optional; type *boolean*
    - uniform\_magnitude - optional; type *double*
    - magnitude\_1 - optional; type *double*
    - magnitude\_2 - optional; type *double*
    - magnitude\_3 - optional; type *double*
    - uniform\_magnitude\_t\_c - optional; type *double*
    - magnitude\_t\_c\_1 - optional; type *double*
    - magnitude\_t\_c\_2 - optional; type *double*
    - magnitude\_t\_c\_3 - optional; type *double*
    - uniform\_magnitude\_delta\_t - optional; type *double*
    - magnitude\_delta\_t\_1 - optional; type *double*
    - magnitude\_delta\_t\_2 - optional; type *double*
    - magnitude\_delta\_t\_3 - optional; type *double*
    - magnitude\_axial\_strain\_x - optional; type *double*
    - magnitude\_axial\_strain\_y - optional; type *double*
    - magnitude\_axial\_strain\_1x - optional; type *double*
    - magnitude\_axial\_strain\_1y - optional; type *double*
    - magnitude\_axial\_strain\_2x - optional; type *double*
    - magnitude\_axial\_strain\_2y - optional; type *double*
    - magnitude\_axial\_strain\_3x - optional; type *double*
    - magnitude\_axial\_strain\_3y - optional; type *double*
    - angular\_velocity - optional; type *double*
    - angular\_acceleration - optional; type *double*
    - node\_1 - optional; type *int*
    - node\_2 - optional; type *int*
    - node\_3 - optional; type *int*

- axis\_definition\_type - optional; type *surface\_set\_load\_axis\_definition\_type* - type *undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *surface\_set\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *surface\_set\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- varying\_load\_parameters - optional; type *array\_of\_surface\_set\_load\_varying\_load\_parameters*
  - *surface\_set\_load\_varying\_load\_parameters* - optional, unbounded; type *surface\_set\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- form\_finding\_definition - optional; type *surface\_set\_load\_form\_finding\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_DEFINITION\_FORCE', 'FORM\_FINDING\_DEFINITION\_STRESS' }
- magnitude\_force\_x - optional; type *double*
- magnitude\_force\_y - optional; type *double*
- magnitude\_force\_u - optional; type *double*
- magnitude\_force\_v - optional; type *double*
- magnitude\_force\_r - optional; type *double*
- magnitude\_force\_t - optional; type *double*
- magnitude\_mass\_x - optional; type *double*
- magnitude\_mass\_y - optional; type *double*
- magnitude\_mass\_z - optional; type *double*
- magnitude\_stress\_x - optional; type *double*
- magnitude\_stress\_y - optional; type *double*
- magnitude\_stress\_u - optional; type *double*
- magnitude\_stress\_v - optional; type *double*
- magnitude\_stress\_r - optional; type *double*
- magnitude\_stress\_t - optional; type *double*
- form\_finding\_calculation\_method - optional; type *surface\_set\_load\_form\_finding\_calculation\_method* - type *undefined* with restriction - enum { 'FORM\_FINDING\_CALCULATION\_METHOD\_PROJECTION', 'FORM\_FINDING\_CALCULATION\_METHOD\_STANDARD' }
- magnitude\_mass\_global - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 121. get\_surface\_stiffness\_modification

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_surface\_stiffness\_modification

**Input:** get\_surface\_stiffness\_modification\_request (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_stiffness\_modification*

- no type *int*

**Output:** get\_surface\_stiffness\_modification\_response (soap:body, use = literal) [Source code](#)

**parameters** type *get\_surface\_stiffness\_modificationResponse*

- value type *surface\_stiffness\_modification*
  - no type *int*
  - type - optional; type *surface\_stiffness\_modification\_type* - type *undefined* with restriction - enum { 'TYPE\_CONCRETE\_STRUCTURES\_ACI', 'TYPE\_CONCRETE\_STRUCTURES\_CSA', 'TYPE\_PARTIAL\_STIFFNESSES\_FACTORS', 'TYPE\_STIFFNESS\_MATRIX\_ELEMENTS\_FACTORS', 'TYPE\_TOTAL\_STIFFNESS\_FACTOR' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - factor\_of\_total\_stiffness - optional; type *double*
  - factor\_of\_bending\_stiffness - optional; type *double*
  - factor\_of\_shear\_stiffness - optional; type *double*
  - factor\_of\_membrane\_stiffness - optional; type *double*
  - factor\_of\_eccentric\_effects - optional; type *double*
  - factor\_of\_weight - optional; type *double*
  - kd11 - optional; type *double*
  - kd12 - optional; type *double*
  - kd13 - optional; type *double*
  - kd22 - optional; type *double*
  - kd23 - optional; type *double*
  - kd33 - optional; type *double*
  - kd44 - optional; type *double*
  - kd45 - optional; type *double*
  - kd55 - optional; type *double*
  - kd66 - optional; type *double*
  - kd67 - optional; type *double*
  - kd68 - optional; type *double*

- kd77 - optional; type *double*
- kd78 - optional; type *double*
- kd88 - optional; type *double*
- kd16 - optional; type *double*
- kd17 - optional; type *double*
- kd18 - optional; type *double*
- kd27 - optional; type *double*
- kd28 - optional; type *double*
- kd38 - optional; type *double*
- kd11\_note - optional; type *string*
- kd12\_note - optional; type *string*
- kd13\_note - optional; type *string*
- kd22\_note - optional; type *string*
- kd23\_note - optional; type *string*
- kd33\_note - optional; type *string*
- kd44\_note - optional; type *string*
- kd45\_note - optional; type *string*
- kd55\_note - optional; type *string*
- kd66\_note - optional; type *string*
- kd67\_note - optional; type *string*
- kd68\_note - optional; type *string*
- kd77\_note - optional; type *string*
- kd78\_note - optional; type *string*
- kd88\_note - optional; type *string*
- kd16\_note - optional; type *string*
- kd17\_note - optional; type *string*
- kd18\_note - optional; type *string*
- kd27\_note - optional; type *string*
- kd28\_note - optional; type *string*
- kd38\_note - optional; type *string*
- assigned\_to\_structural\_modification - optional; type *array\_of\_int*
- concrete\_structures\_component\_type - optional; type *surface\_stiffness\_modification\_concrete\_structures\_component\_type* - type *undefined* with restriction - enum { 'COMPONENT\_TYPE\_BEAMS', 'COMPONENT\_TYPE\_COLUMNS', 'COMPONENT\_TYPE\_FLAT\_PLATES\_AND\_FLAT\_SLABS', 'COMPONENT\_TYPE\_WALLS\_CRACKED', 'COMPONENT\_TYPE\_WALLS\_UNCRACKED' }
- concrete\_structures\_bending\_stiffness\_factor - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 122. `get_surface_support`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_surface_support`

**Input:** `get_surface_support_request` (soap:body, use = literal) [Source code](#)

parameters type *get\_surface\_support*

- no type *int*

**Output:** `get_surface_support_response` (soap:body, use = literal) [Source code](#)

parameters type *get\_surface\_supportResponse*

- value type *surface\_support*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - surfaces - optional; type *array\_of\_int*
  - translation - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - translation\_x - optional; type *double*
  - translation\_y - optional; type *double*
  - translation\_z - optional; type *double*
  - shear\_xz - optional; type *double*
  - shear\_yz - optional; type *double*
  - nonlinearity - optional; type *surface\_support\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_FAILURE\_IF\_NEGATIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_FAILURE\_IF\_POSITIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_NONE' }
  - negative\_nonlinearity\_type - optional; type *surface\_support\_negative\_nonlinearity\_type* - type *undefined* with restriction - enum { 'NONLINEARITY\_DEFINITION\_TYPE\_BASIC\_UNDIRECTIONAL\_ACTION', 'NONLINEARITY\_DEFINITION\_TYPE\_FRICTION\_PLANE\_XY', 'NONLINEARITY\_DEFINITION\_TYPE\_YIELDING\_CONTACT\_STRESS\_SIGMA\_Z' }
  - positive\_nonlinearity\_type - optional; type *surface\_support\_positive\_nonlinearity\_type* - type *undefined* with restriction - enum { 'NONLINEARITY\_DEFINITION\_TYPE\_BASIC\_UNDIRECTIONAL\_ACTION', 'NONLINEARITY\_DEFINITION\_TYPE\_FRICTION\_PLANE\_XY', 'NONLINEARITY\_DEFINITION\_TYPE\_YIELDING\_CONTACT\_STRESS\_SIGMA\_Z' }
  - negative\_friction\_coefficient - optional; type *double*
  - positive\_friction\_coefficient - optional; type *double*
  - negative\_contact\_stress - optional; type *double*
  - positive\_contact\_stress - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 123. `get_surfaces_contact`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.



**SOAP action:** [http://localhost:8082/get\\_surfaces\\_contact](http://localhost:8082/get_surfaces_contact)

**Input:** [get\\_surfaces\\_contact\\_request](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_surfaces_contact

- no type int

```

**Output:** [get\\_surfaces\\_contact\\_response](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_surfaces_contactResponse

- value type surfaces_contact
 - no type int
 - surfaces_contact_type - optional; type int
 - surfaces_group1 - optional; type array_of_int
 - surfaces_group2 - optional; type array_of_int
 - comment - optional; type string
 - id_for_export_import - optional; type string
 - metadata_for_export_import - optional; type string

```

#### 124. [get\\_surfaces\\_contact\\_type](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_surfaces\\_contact\\_type](http://localhost:8082/get_surfaces_contact_type)

**Input:** [get\\_surfaces\\_contact\\_type\\_request](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_surfaces_contact_type

- no type int

```

**Output:** [get\\_surfaces\\_contact\\_type\\_response](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_surfaces_contact_typeResponse

- value type surfaces_contact_type
 - no type int
 - user_defined_name_enabled - optional; type boolean
 - name - optional; type string
 - comment - optional; type string
 - surfaces_contacts - optional; type array_of_int
 - perpendicular_to_surface - optional; type surfaces_contact_type_perpendicular_to_surface - type undefined with restriction - enum { 'FAILURE_UNDER_COMPRESSION', 'FAILURE_UNDER_TENSION', 'FULL_FORCE_TRANSMISSION' }
 - parallel_to_surface - optional; type surfaces_contact_type_parallel_to_surface - type undefined with restriction - enum { 'ELASTIC_FRICTION', 'ELASTIC_SURFACE', 'FAILURE_IF_CONTACT_PERPENDICULAR_TO_SURFACES_FAILED', 'FULL_FORCE_TRANSMISSION', 'RIGID_FRICTION' }
 - rigid_friction_type - optional; type surfaces_contact_type_rigid_friction_type - type undefined with restriction - enum { 'FRICTION_COEFFICIENT', 'LIMIT_STRESS' }
 - rigid_friction_coefficient - optional; type double
 - rigid_friction_limit_stress - optional; type double
 - elastic_behavior_shear_stiffness - optional; type double
 - elastic_friction_type - optional; type surfaces_contact_type_elastic_friction_type - type undefined with restriction - enum { 'FRICTION_COEFFICIENT', 'LIMIT_STRESS' }
 - elastic_friction_coefficient - optional; type double
 - elastic_friction_limit_stress - optional; type double
 - elastic_friction_shear_stiffness - optional; type double
 - id_for_export_import - optional; type string
 - metadata_for_export_import - optional; type string

```

#### 125. [get\\_terrain](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/get\\_terrain](http://localhost:8082/get_terrain)

**Input:** [get\\_terrain\\_request](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_terrain

- no type int

```

**Output:** [get\\_terrain\\_response](#) (soap:body, use = literal) [Source code](#)

```
parameters type get_terrainResponse

- value type terrain
 - no type int
 - type - optional; type terrain_type - type undefined with restriction - enum { 'HORIZONTAL_PLANE', 'INCLINED_PLANE', 'NO_TERRAIN', 'SOIL_SAMPLES', 'TABLE' }
 - comment - optional; type string
 - bounding_box_offset_x - optional; type double
 - bounding_box_offset_y - optional; type double
 - center_of_terrain_z - optional; type double
 - rotation_around_Z - optional; type double
 - consider_soil_samples - optional; type boolean
 - coordinate_system - optional; type int
 - terrain_table - optional; type array_of_terrain_terrain_table
 - terrain_terrain_table - optional, unbounded; type terrain_terrain_table
 - no - optional; type int
 - global_x - optional; type double
 - global_y - optional; type double
 - global_z - optional; type double
 - id_for_export_import - optional; type string
 - metadata_for_export_import - optional; type string

```

#### 126. [get\\_thickness](#)

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_thickness

**Input:** get\_thickness\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_thickness*

- no type *int*

**Output:** get\_thickness\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_thicknessResponse*

- value type *thickness*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to\_surfaces - optional; type *array\_of\_int*
  - type - optional; type *thickness\_type* - type *undefined* with restriction - enum { 'TYPE\_LAYERS', 'TYPE\_SHAPE\_ORTHOTROPY', 'TYPE\_STIFFNESS\_MATRIX', 'TYPE\_THICKNESS\_PHASE', 'TYPE\_UNIFORM', 'TYPE\_VARIABLE\_CIRCLE', 'TYPE\_VARIABLE\_FOUR\_SURFACE\_CORNERS', 'TYPE\_VARIABLE\_THREE\_NODES', 'TYPE\_VARIABLE\_TWO\_NODES\_AND\_DIRECTION' }
  - material - optional; type *int*
  - uniform\_thickness - optional; type *double*
  - advanced\_time\_dependent\_properties\_of\_concrete\_enabled - optional; type *boolean*
  - creep\_enabled - optional; type *boolean*
  - shrinkage\_enabled - optional; type *boolean*
  - relative\_humidity - optional; type *double*
  - thickness\_1 - optional; type *double*
  - node\_1 - optional; type *int*
  - thickness\_2 - optional; type *double*
  - node\_2 - optional; type *int*
  - thickness\_3 - optional; type *double*
  - node\_3 - optional; type *int*
  - direction - optional; type *thickness\_direction* - type *undefined* with restriction - enum { 'THICKNESS\_DIRECTION\_IN\_SMALL\_X', 'THICKNESS\_DIRECTION\_IN\_SMALL\_Y', 'THICKNESS\_DIRECTION\_IN\_X', 'THICKNESS\_DIRECTION\_IN\_Y', 'THICKNESS\_DIRECTION\_IN\_Z' }
  - thickness\_4 - optional; type *double*
  - node\_4 - optional; type *int*
  - thickness\_circle\_center - optional; type *double*
  - thickness\_circle\_line - optional; type *double*
  - comment - optional; type *string*
  - layers\_reference\_table - optional; type *array\_of\_thickness\_layers\_reference\_table*
    - thickness\_layers\_reference\_table - optional, unbounded; type *thickness\_layers\_reference\_table*
      - no - optional; type *int*
      - layer\_no - optional; type *int*
      - layer\_type - optional; type *layer\_type* - type *undefined* with restriction
      - thickness\_type - optional; type *int*
      - material - optional; type *int*
      - thickness - optional; type *double*
      - angle - optional; type *double*
      - connection\_with\_other\_topological\_elements - optional; type *boolean*
      - comment - optional; type *string*
      - specific\_weight - optional; type *double*
      - weight - optional; type *double*
  - layers\_solid\_model\_enabled - optional; type *boolean*
  - layers\_gas\_enabled - optional; type *boolean*
  - layers\_total\_thickness - optional; type *double*
  - layers\_total\_weight - optional; type *double*
  - stiffness\_reduction\_enabled - optional; type *boolean*
  - K33 - optional; type *double*
  - K44 - optional; type *double*
  - K55 - optional; type *double*
  - K88 - optional; type *double*
  - K33\_note - optional; type *string*
  - K44\_note - optional; type *string*
  - K55\_note - optional; type *string*
  - K88\_note - optional; type *string*
  - function\_data\_function\_type - optional; type *thickness\_function\_data\_function\_type* - type *undefined* with restriction - enum { 'FUNCTION\_TYPE\_CREEP\_COEFFICIENT', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CA', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CD', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CS' }
  - function\_data\_layer\_id - optional; type *int*
  - function\_data\_age\_of\_concrete\_at\_the\_considered\_moment - optional; type *double*
  - function\_data\_number\_of\_steps - optional; type *int*
  - function\_data\_coefficients - optional; type *array\_of\_thickness\_function\_data\_coefficients*
    - thickness\_function\_data\_coefficients - optional, unbounded; type *thickness\_function\_data\_coefficients*
      - no - optional; type *int*
      - time - optional; type *double*
      - coefficient - optional; type *double*
  - parent\_thickness - optional; type *int*
  - parent\_thickness\_layers - optional; type *array\_of\_int*
  - timber\_frame\_wall\_sheathing\_front\_material\_enabled - optional; type *boolean*
  - timber\_frame\_wall\_sheathing\_front\_material - optional; type *int*
  - timber\_frame\_wall\_sheathing\_front\_thickness - optional; type *double*
  - timber\_frame\_wall\_sheathing\_back\_material\_enabled - optional; type *boolean*
  - timber\_frame\_wall\_sheathing\_back\_material - optional; type *int*
  - timber\_frame\_wall\_sheathing\_back\_thickness - optional; type *double*
  - timber\_frame\_wall\_horizontal\_framing\_members\_cross\_section - optional; type *int*
  - timber\_frame\_wall\_vertical\_framing\_members\_cross\_section - optional; type *int*
  - timber\_frame\_wall\_framing\_members\_connector\_type - optional; type *thickness\_timber\_frame\_wall\_framing\_members\_connector\_type* - type *undefined* with restriction - enum { 'TYPE\_NAIL', 'TYPE\_STAPLE' }
  - timber\_frame\_wall\_framing\_members\_connector\_diameter - optional; type *double*
  - timber\_frame\_wall\_framing\_members\_connector\_dimension\_type - optional; type *thickness\_timber\_frame\_wall\_framing\_members\_connector\_dimension\_type* - type *undefined* with restriction - enum { 'TYPE\_0075\_X\_1060', 'TYPE\_0075\_X\_1130', 'TYPE\_0075\_X\_1300', 'TYPE\_0110\_X\_0610', 'TYPE\_0120\_X\_0200', 'TYPE\_0125\_X\_0620', 'TYPE\_0125\_X\_0770', 'TYPE\_0125\_X\_1060', 'TYPE\_0125\_X\_1130', 'TYPE\_0125\_X\_1250', 'TYPE\_0125\_X\_1300', 'TYPE\_0165\_X\_0280' }
  - timber\_frame\_wall\_framing\_members\_connector\_spacing - optional; type *double*
  - timber\_frame\_wall\_vertical\_studs\_cross\_section - optional; type *int*

- timber\_frame\_wall\_vertical\_studs\_distribution - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_distribution* - type *undefined* with restriction - enum { 'TYPE\_CENTER\_TO\_CENTER', 'TYPE\_GAP', 'TYPE\_UNIFORM', 'TYPE\_USER\_DEFINED' }
- timber\_frame\_wall\_vertical\_studs\_reverse\_distribution\_enabled - optional; type *boolean*
- timber\_frame\_wall\_vertical\_studs\_spacing - optional; type *double*
- timber\_frame\_wall\_vertical\_studs\_spacing\_table - optional; type *array\_of\_thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table*
  - *thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table* - optional, unbounded; type *thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table*
    - no - optional; type *int*
    - spacing - optional; type *double*
    - note - optional; type *string*
- timber\_frame\_wall\_vertical\_studs\_spacing\_definition\_relative - optional; type *boolean*
- timber\_frame\_wall\_vertical\_studs\_connector\_type - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_connector\_type* - type *undefined* with restriction - enum { 'TYPE\_NAIL', 'TYPE\_STAPLE' }
- timber\_frame\_wall\_vertical\_studs\_connector\_diameter - optional; type *double*
- timber\_frame\_wall\_vertical\_studs\_connector\_dimension\_type - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_connector\_dimension\_type* - type *undefined* with restriction - enum { 'TYPE\_0075\_X\_1060', 'TYPE\_0075\_X\_1130', 'TYPE\_0075\_X\_1300', 'TYPE\_0110\_X\_0610', 'TYPE\_0120\_X\_0200', 'TYPE\_0125\_X\_0620', 'TYPE\_0125\_X\_0770', 'TYPE\_0125\_X\_1060', 'TYPE\_0125\_X\_1130', 'TYPE\_0125\_X\_1250', 'TYPE\_0125\_X\_1300', 'TYPE\_0165\_X\_0280' }
- timber\_frame\_wall\_vertical\_studs\_connector\_spacing - optional; type *double*
- timber\_frame\_wall\_blockings\_cross\_section\_enabled - optional; type *boolean*
- timber\_frame\_wall\_blockings\_cross\_section - optional; type *int*
- timber\_frame\_wall\_blockings\_distribution - optional; type *thickness\_timber\_frame\_wall\_blockings\_distribution* - type *undefined* with restriction - enum { 'TYPE\_ALTERNATING', 'TYPE\_INLINE', 'TYPE\_STAGGERED' }
- timber\_frame\_wall\_blockings\_offset - optional; type *double*
- orthotropy\_type - optional; type *thickness\_orthotropy\_type* - type *undefined* with restriction - enum { 'ORTHOTROPIC\_THICKNESS\_TYPE\_BIDIRECTIONAL\_RIBBED\_PLATE', 'ORTHOTROPIC\_THICKNESS\_TYPE\_COUPLING', 'ORTHOTROPIC\_THICKNESS\_TYPE\_EFFECTIVE\_THICKNESS', 'ORTHOTROPIC\_THICKNESS\_TYPE\_GRILLAGE', 'ORTHOTROPIC\_THICKNESS\_TYPE\_HOLLOW\_CORE\_SLAB', 'ORTHOTROPIC\_THICKNESS\_TYPE\_TRAPEZOIDAL\_SHEET', 'ORTHOTROPIC\_THICKNESS\_TYPE\_UNIDIRECTIONAL\_RIBBED\_PLATE' }
- orthotropy\_rotation\_beta - optional; type *double*
- orthotropy\_fictitious\_thickness - optional; type *double*
- shape\_orthotropy\_self\_weight\_definition\_type - optional; type *thickness\_shape\_orthotropy\_self\_weight\_definition\_type* - type *undefined* with restriction - enum { 'SELF\_WEIGHT\_COMPUTED\_FROM\_PARAMETERS', 'SELF\_WEIGHT\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS', 'SELF\_WEIGHT\_DEFINED\_VIA\_WEIGHT' }
- shape\_orthotropy\_effective\_thickness\_x - optional; type *double*
- shape\_orthotropy\_effective\_thickness\_y - optional; type *double*
- shape\_orthotropy\_self\_weight - optional; type *double*
- stiffness\_matrix\_self\_weight\_definition\_type - optional; type *thickness\_stiffness\_matrix\_self\_weight\_definition\_type* - type *undefined* with restriction - enum { 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_BULK\_DENSITY\_AND\_AREA\_DENSITY', 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS\_AND\_AREA\_DENSITY', 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS\_AND\_BULK\_DENSITY' }
- stiffness\_matrix\_bulk\_density - optional; type *double*
- stiffness\_matrix\_area\_density - optional; type *double*
- stiffness\_matrix\_coefficient\_of\_thermal\_expansion - optional; type *double*
- coupling\_thickness - optional; type *double*
- coupling\_spacing - optional; type *double*
- coupling\_width - optional; type *double*
- slab\_thickness - optional; type *double*
- rib\_height - optional; type *double*
- rib\_spacing - optional; type *double*
- rib\_width - optional; type *double*
- rib\_spacing\_x - optional; type *double*
- rib\_spacing\_y - optional; type *double*
- rib\_width\_x - optional; type *double*
- rib\_width\_y - optional; type *double*
- sheet\_thickness - optional; type *double*
- total\_profile\_height - optional; type *double*
- top\_flange\_width - optional; type *double*
- bottom\_flange\_width - optional; type *double*
- void\_spacing - optional; type *double*
- void\_diameter - optional; type *double*
- rib\_height\_x - optional; type *double*
- rib\_height\_y - optional; type *double*
- D11 - optional; type *double*
- D11\_note - optional; type *string*
- D12 - optional; type *double*
- D12\_note - optional; type *string*
- D13 - optional; type *double*
- D13\_note - optional; type *string*
- D16 - optional; type *double*
- D16\_note - optional; type *string*
- D17 - optional; type *double*
- D17\_note - optional; type *string*
- D18 - optional; type *double*
- D18\_note - optional; type *string*
- D22 - optional; type *double*
- D22\_note - optional; type *string*
- D23 - optional; type *double*
- D23\_note - optional; type *string*
- D27 - optional; type *double*
- D27\_note - optional; type *string*
- D28 - optional; type *double*
- D28\_note - optional; type *string*
- D33 - optional; type *double*
- D33\_note - optional; type *string*
- D38 - optional; type *double*
- D38\_note - optional; type *string*
- D44 - optional; type *double*
- D44\_note - optional; type *string*
- D45 - optional; type *double*

- D45\_note - optional; type *string*
- D55 - optional; type *double*
- D55\_note - optional; type *string*
- D66 - optional; type *double*
- D66\_note - optional; type *string*
- D67 - optional; type *double*
- D67\_note - optional; type *string*
- D68 - optional; type *double*
- D68\_note - optional; type *string*
- D77 - optional; type *double*
- D77\_note - optional; type *string*
- D78 - optional; type *double*
- D78\_note - optional; type *string*
- D88 - optional; type *double*
- D88\_note - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 127. get\_visual\_object

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_visual\_object

**Input:** get\_visual\_object\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_visual\_object*

- no type *int*

**Output:** get\_visual\_object\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_visual\_objectResponse*

- value type *visual\_object*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - filename - optional; type *string*
  - coordinate\_system - optional; type *int*
  - insert\_point - optional; type *visual\_object\_insert\_point* - type *undefined* with restriction - enum { 'INSERT\_CENTER', 'INSERT\_MX', 'INSERT\_MXMYMZ', 'INSERT\_MXMYPZ', 'INSERT\_MXPYMZ', 'INSERT\_MXPYPZ', 'INSERT\_MY', 'INSERT\_MZ', 'INSERT\_PX', 'INSERT\_PXMYMZ', 'INSERT\_PXMYPZ', 'INSERT\_PXPYMZ', 'INSERT\_PXPYPZ', 'INSERT\_PY', 'INSERT\_PZ' }
  - origin\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - origin\_coordinate\_x - optional; type *double*
  - origin\_coordinate\_y - optional; type *double*
  - origin\_coordinate\_z - optional; type *double*
  - rotation\_angles\_sequence - optional; type *visual\_object\_rotation\_angles\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_UVW', 'SEQUENCE\_UWV', 'SEQUENCE\_VUW', 'SEQUENCE\_VWU', 'SEQUENCE\_WUV', 'SEQUENCE\_WVU', 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - rotation\_angle\_1 - optional; type *double*
  - rotation\_angle\_2 - optional; type *double*
  - rotation\_angle\_3 - optional; type *double*
  - scale\_is\_nonuniform - optional; type *boolean*
  - scale\_is\_defined\_as\_relative - optional; type *boolean*
  - scale\_absolute - optional; type *double*
  - scale\_relative - optional; type *double*
  - scale\_absolute\_x - optional; type *double*
  - scale\_absolute\_y - optional; type *double*
  - scale\_absolute\_z - optional; type *double*
  - scale\_relative\_x - optional; type *double*
  - scale\_relative\_y - optional; type *double*
  - scale\_relative\_z - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

## 128. get\_wind\_profile

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_wind\_profile

**Input:** get\_wind\_profile\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_wind\_profile*

- no type *int*

**Output:** get\_wind\_profile\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_wind\_profileResponse*

- value type *wind\_profile*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - type - optional; type *wind\_profile\_type* - type *undefined* with restriction - enum { 'ACCORDING\_TO\_STANDARD', 'USER\_DEFINED' }
  - user\_defined\_wind\_profile\_step\_enabled - optional; type *boolean*
  - user\_defined\_wind\_profile\_sorted - optional; type *boolean*
  - user\_defined\_wind\_profile\_uniform\_intensity\_enabled - optional; type *boolean*

- user\_defined\_wind\_profile\_uniform\_intensity - optional; type *double*
- user\_defined\_wind\_profile\_period\_step - optional; type *double*
- user\_defined\_wind\_profile - optional; type *array\_of\_wind\_profile\_user\_defined\_wind\_profile*
  - wind\_profile\_user\_defined\_wind\_profile - optional, unbounded; type *wind\_profile\_user\_defined\_wind\_profile*
    - no - optional; type *int*
    - height - optional; type *double*
    - velocity - optional; type *double*
    - turbulence\_intensity - optional; type *double*
- comment - optional; type *string*
- load\_zone - optional; type *wind\_profile\_load\_zone* - type *undefined* with restriction - enum {
  - 'E\_LOAD\_ZONE\_NOT\_DEFINED', 'E\_LOAD\_ZONE\_TYPE\_0\_18\_TO\_0\_38', 'E\_LOAD\_ZONE\_TYPE\_0\_39\_TO\_0\_48',
  - 'E\_LOAD\_ZONE\_TYPE\_0\_4', 'E\_LOAD\_ZONE\_TYPE\_0\_49\_TO\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_0\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_0\_61\_TO\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_0\_7',
  - 'E\_LOAD\_ZONE\_TYPE\_0\_71\_TO\_0\_8', 'E\_LOAD\_ZONE\_TYPE\_0\_9', 'E\_LOAD\_ZONE\_TYPE\_0\_M',
  - 'E\_LOAD\_ZONE\_TYPE\_1', 'E\_LOAD\_ZONE\_TYPE\_10', 'E\_LOAD\_ZONE\_TYPE\_100', 'E\_LOAD\_ZONE\_TYPE\_15',
  - 'E\_LOAD\_ZONE\_TYPE\_19', 'E\_LOAD\_ZONE\_TYPE\_1\_0\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_1\_2',
  - 'E\_LOAD\_ZONE\_TYPE\_1\_25', 'E\_LOAD\_ZONE\_TYPE\_1\_5', 'E\_LOAD\_ZONE\_TYPE\_1\_5\_COMMA',
  - 'E\_LOAD\_ZONE\_TYPE\_1\_75', 'E\_LOAD\_ZONE\_TYPE\_1\_8', 'E\_LOAD\_ZONE\_TYPE\_1\_A',
  - 'E\_LOAD\_ZONE\_TYPE\_1\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_1\_A\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_1\_B',
  - 'E\_LOAD\_ZONE\_TYPE\_1\_TO\_2', 'E\_LOAD\_ZONE\_TYPE\_1\_V', 'E\_LOAD\_ZONE\_TYPE\_2',
  - 'E\_LOAD\_ZONE\_TYPE\_20', 'E\_LOAD\_ZONE\_TYPE\_20\_9\_LESS', 'E\_LOAD\_ZONE\_TYPE\_21',
  - 'E\_LOAD\_ZONE\_TYPE\_22', 'E\_LOAD\_ZONE\_TYPE\_23', 'E\_LOAD\_ZONE\_TYPE\_23\_6', 'E\_LOAD\_ZONE\_TYPE\_24',
  - 'E\_LOAD\_ZONE\_TYPE\_24\_2\_LESS', 'E\_LOAD\_ZONE\_TYPE\_25', 'E\_LOAD\_ZONE\_TYPE\_26',
  - 'E\_LOAD\_ZONE\_TYPE\_27', 'E\_LOAD\_ZONE\_TYPE\_27\_5\_LESS', 'E\_LOAD\_ZONE\_TYPE\_28',
  - 'E\_LOAD\_ZONE\_TYPE\_29', 'E\_LOAD\_ZONE\_TYPE\_2\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_0\_2\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_0\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_2\_1', 'E\_LOAD\_ZONE\_TYPE\_2\_3',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_4', 'E\_LOAD\_ZONE\_TYPE\_2\_5', 'E\_LOAD\_ZONE\_TYPE\_2\_5\_2\_75',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_5\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_2\_7',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_75', 'E\_LOAD\_ZONE\_TYPE\_2\_75\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_75\_3\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_A', 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK\_2',
  - 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK\_3', 'E\_LOAD\_ZONE\_TYPE\_2\_B', 'E\_LOAD\_ZONE\_TYPE\_2\_V',
  - 'E\_LOAD\_ZONE\_TYPE\_3', 'E\_LOAD\_ZONE\_TYPE\_30', 'E\_LOAD\_ZONE\_TYPE\_30\_8\_LESS',
  - 'E\_LOAD\_ZONE\_TYPE\_32', 'E\_LOAD\_ZONE\_TYPE\_33', 'E\_LOAD\_ZONE\_TYPE\_34\_1\_LESS',
  - 'E\_LOAD\_ZONE\_TYPE\_35', 'E\_LOAD\_ZONE\_TYPE\_36\_0', 'E\_LOAD\_ZONE\_TYPE\_37\_4\_LESS',
  - 'E\_LOAD\_ZONE\_TYPE\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_3\_0\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_3\_0\_COMMA',
  - 'E\_LOAD\_ZONE\_TYPE\_3\_3', 'E\_LOAD\_ZONE\_TYPE\_3\_4', 'E\_LOAD\_ZONE\_TYPE\_3\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_3\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_3\_6', 'E\_LOAD\_ZONE\_TYPE\_3\_A',
  - 'E\_LOAD\_ZONE\_TYPE\_3\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_3\_A\_GREATER', 'E\_LOAD\_ZONE\_TYPE\_4',
  - 'E\_LOAD\_ZONE\_TYPE\_40', 'E\_LOAD\_ZONE\_TYPE\_40\_7\_LESS', 'E\_LOAD\_ZONE\_TYPE\_44\_0\_LESS',
  - 'E\_LOAD\_ZONE\_TYPE\_45', 'E\_LOAD\_ZONE\_TYPE\_47\_3\_LESS', 'E\_LOAD\_ZONE\_TYPE\_48',
  - 'E\_LOAD\_ZONE\_TYPE\_4\_0', 'E\_LOAD\_ZONE\_TYPE\_4\_5', 'E\_LOAD\_ZONE\_TYPE\_4\_5\_COMMA',
  - 'E\_LOAD\_ZONE\_TYPE\_5', 'E\_LOAD\_ZONE\_TYPE\_50', 'E\_LOAD\_ZONE\_TYPE\_50\_6\_LESS',
  - 'E\_LOAD\_ZONE\_TYPE\_5\_0', 'E\_LOAD\_ZONE\_TYPE\_5\_5', 'E\_LOAD\_ZONE\_TYPE\_5\_5\_COMMA',
  - 'E\_LOAD\_ZONE\_TYPE\_6', 'E\_LOAD\_ZONE\_TYPE\_60', 'E\_LOAD\_ZONE\_TYPE\_6\_0', 'E\_LOAD\_ZONE\_TYPE\_6\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_7', 'E\_LOAD\_ZONE\_TYPE\_70', 'E\_LOAD\_ZONE\_TYPE\_7\_0', 'E\_LOAD\_ZONE\_TYPE\_7\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_8', 'E\_LOAD\_ZONE\_TYPE\_80', 'E\_LOAD\_ZONE\_TYPE\_8\_0', 'E\_LOAD\_ZONE\_TYPE\_9',
  - 'E\_LOAD\_ZONE\_TYPE\_90', 'E\_LOAD\_ZONE\_TYPE\_A', 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_200\_M',
  - 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_400\_M', 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_500\_M', 'E\_LOAD\_ZONE\_TYPE\_A\_1',
  - 'E\_LOAD\_ZONE\_TYPE\_A\_2', 'E\_LOAD\_ZONE\_TYPE\_A\_2\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_A\_3',
  - 'E\_LOAD\_ZONE\_TYPE\_A\_3\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_A\_4', 'E\_LOAD\_ZONE\_TYPE\_A\_4\_ASTERISK',
  - 'E\_LOAD\_ZONE\_TYPE\_B', 'E\_LOAD\_ZONE\_TYPE\_B\_1', 'E\_LOAD\_ZONE\_TYPE\_B\_2', 'E\_LOAD\_ZONE\_TYPE\_CS',
  - 'E\_LOAD\_ZONE\_TYPE\_C\_1', 'E\_LOAD\_ZONE\_TYPE\_C\_2', 'E\_LOAD\_ZONE\_TYPE\_D', 'E\_LOAD\_ZONE\_TYPE\_E',
  - 'E\_LOAD\_ZONE\_TYPE\_GAMMA', 'E\_LOAD\_ZONE\_TYPE\_GAMMA\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_GEQ\_8',
  - 'E\_LOAD\_ZONE\_TYPE\_GUADELOUPE', 'E\_LOAD\_ZONE\_TYPE\_GUYANE', 'E\_LOAD\_ZONE\_TYPE\_H',
  - 'E\_LOAD\_ZONE\_TYPE\_I', 'E\_LOAD\_ZONE\_TYPE\_II', 'E\_LOAD\_ZONE\_TYPE\_III',
  - 'E\_LOAD\_ZONE\_TYPE\_III\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_II\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_IV',
  - 'E\_LOAD\_ZONE\_TYPE\_IX', 'E\_LOAD\_ZONE\_TYPE\_I\_A', 'E\_LOAD\_ZONE\_TYPE\_I\_ASTERISK\_A',
  - 'E\_LOAD\_ZONE\_TYPE\_I\_ASTERISK\_M', 'E\_LOAD\_ZONE\_TYPE\_I\_M', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_25',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_30', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_35', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_40',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_45', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_50',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_55', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_60',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_75', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_8',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_9', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_95', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_100',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_110', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_120', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_130',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_140', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_150', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_160',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_170', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_180', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_0',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_1', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_2', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_5',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_75', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_200', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_0',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_25', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_75',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_4\_0',
  - 'E\_LOAD\_ZONE\_TYPE\_LEQ\_4\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_95', 'E\_LOAD\_ZONE\_TYPE\_MARTINIQUE',
  - 'E\_LOAD\_ZONE\_TYPE\_MAYOTTE', 'E\_LOAD\_ZONE\_TYPE\_MEQ\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_M\_1',
  - 'E\_LOAD\_ZONE\_TYPE\_N\_A', 'E\_LOAD\_ZONE\_TYPE\_REUNION', 'E\_LOAD\_ZONE\_TYPE\_SR',
  - 'E\_LOAD\_ZONE\_TYPE\_UNDER\_200\_M', 'E\_LOAD\_ZONE\_TYPE\_V', 'E\_LOAD\_ZONE\_TYPE\_VI',
  - 'E\_LOAD\_ZONE\_TYPE\_VII', 'E\_LOAD\_ZONE\_TYPE\_VIII', 'E\_LOAD\_ZONE\_TYPE\_Z1', 'E\_LOAD\_ZONE\_TYPE\_Z2',
  - 'E\_LOAD\_ZONE\_TYPE\_Z3', 'E\_LOAD\_ZONE\_TYPE\_ZERO' }
- definition\_type - optional; type *wind\_profile\_definition\_type* - type *undefined* with restriction - enum {
  - 'E\_DEFINITION\_TYPE\_PARAMETERS\_FROM\_MAP', 'E\_DEFINITION\_TYPE\_USER\_DEFINED' }
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

## 129. get\_wind\_simulation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/get\_wind\_simulation

**Input:** get\_wind\_simulation\_request (soap:body, use = literal) [Source code](#)

parameters type *get\_wind\_simulation*

- no type *int*

**Output:** get\_wind\_simulation\_response (soap:body, use = literal) [Source code](#)

parameters type *get\_wind\_simulationResponse*

- value type *wind\_simulation*
  - no type *int*
  - type - optional; type *wind\_simulation\_type* - type *undefined* with restriction - enum { 'STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*

- name - optional; type *string*
- active - optional; type *boolean*
- wind\_profile - optional; type *int*
- wind\_simulation\_analysis\_settings - optional; type *int*
- wind\_direction\_type - optional; type *wind\_simulation\_wind\_direction\_type* - type *undefined* with restriction - enum { 'UNIFORM', 'USER\_DEFINED' }
- uniform\_wind\_direction\_step - optional; type *double*
- uniform\_wind\_direction\_range\_start - optional; type *double*
- uniform\_wind\_direction\_range\_end - optional; type *double*
- user\_defined\_list\_of\_wind\_directions - optional; type *array\_of\_int*
- generate\_into\_load\_cases - optional; type *array\_of\_wind\_simulation\_generate\_into\_load\_cases*
  - wind\_simulation\_generate\_into\_load\_cases - optional, unbounded; type *wind\_simulation\_generate\_into\_load\_cases*
    - no - optional; type *int*
    - direction - optional; type *double*
    - load\_case - optional; type *int*
- consider\_initial\_state - optional; type *boolean*
- initial\_state\_case - optional; type *int*
- initial\_state\_definition\_type - optional; type *wind\_simulation\_initial\_state\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_FINAL\_STATE', 'DEFINITION\_TYPE\_STIFFNESS', 'DEFINITION\_TYPE\_STRAINS', 'DEFINITION\_TYPE\_STRAINS\_WITH\_USER\_DEFINED\_FACTORS' }
- individual\_factors\_of\_selected\_objects\_table - optional; type *array\_of\_wind\_simulation\_individual\_factors\_of\_selected\_objects\_table*
  - wind\_simulation\_individual\_factors\_of\_selected\_objects\_table - optional, unbounded; type *wind\_simulation\_individual\_factors\_of\_selected\_objects\_table*
    - no - optional; type *int*
    - object\_type - optional; type *object\_type* - type *undefined* with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
    - object\_list - optional; type *array\_of\_int*
    - strain\_type - optional; type *strain\_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
    - factor - optional; type *double*
    - comment - optional; type *string*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

### 130. `get_wind_simulation_analysis_settings`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/get_wind_simulation_analysis_settings`

**Input:** `get_wind_simulation_analysis_settings_request` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_wind\_simulation\_analysis\_settings*

- no type *int*

**Output:** `get_wind_simulation_analysis_settings_response` (soap:body, use = literal) [Source code](#)

**parameters** type *get\_wind\_simulation\_analysis\_settingsResponse*

- value type *wind\_simulation\_analysis\_settings*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - comment - optional; type *string*
  - assigned\_to - optional; type *string*
  - density - optional; type *double*
  - kinematic\_viscosity - optional; type *double*
  - numerical\_solver - optional; type *wind\_simulation\_analysis\_settings\_numerical\_solver* - type *undefined* with restriction - enum { 'OPEN\_FOAM' }
  - finite\_volume\_mesh\_density - optional; type *double*
  - maximum\_number\_of\_iterations - optional; type *int*
  - target\_residue - optional; type *double*
  - mesh\_refinement\_type - optional; type *wind\_simulation\_analysis\_settings\_mesh\_refinement\_type* - type *undefined* with restriction - enum { 'DISTANCE\_FROM\_SURFACE', 'SURFACE\_CURVATURE' }
  - snap\_to\_model\_edges - optional; type *boolean*
  - boundary\_layers\_checked - optional; type *boolean*
  - boundary\_layers\_value - optional; type *int*
  - consider\_turbulence - optional; type *boolean*
  - slip\_boundary\_condition\_on\_bottom\_boundary - optional; type *boolean*
  - use\_potential\_flow\_for\_initial\_condition - optional; type *boolean*
  - use\_second\_order\_numerical\_scheme - optional; type *boolean*
  - consider\_wall\_roughness - optional; type *boolean*
  - user\_defined\_dimensions\_of\_wind\_tunnel - optional; type *boolean*
  - member\_load\_distribution - optional; type *wind\_simulation\_analysis\_settings\_member\_load\_distribution* - type *undefined* with restriction - enum { 'CONCENTRATED', 'TRAPEZOIDAL', 'UNIFORM' }
  - turbulence\_model\_type - optional; type *wind\_simulation\_analysis\_settings\_turbulence\_model\_type* - type *undefined* with restriction - enum { 'TURBULENCE\_TYPE\_EPSILON', 'TURBULENCE\_TYPE\_OMEGA' }
  - sand\_grain\_roughness\_height - optional; type *double*
  - roughness\_constant - optional; type *double*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

### 131. `plausibility_check`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/plausibility_check`

**Input:** `plausibility_check_request` (soap:body, use = literal) [Source code](#)

```
parameters type plausibility_check
 ■ skip_warnings type boolean
```

**Output:** `plausibility_check_response` (soap:body, use = literal) [Source code](#)

```
parameters type plausibility_checkResponse
 ■ value type plausibility_check_result
 ■ succeeded - optional; type boolean
 ■ messages - optional; type plausibility_check_messages
 ■ message - optional, unbounded; type plausibility_check_message
 ■ message_type type plausibility_check_message_type - type undefined with restriction - enum {
 'ERROR', 'INFORMATION', 'QUESTION', 'WARNING'
 }
 ■ object - optional; type string
 ■ input_field - optional; type string
 ■ current_value - optional; type string
 ■ message - optional; type string
 ■ result - optional; type boolean
```

### 132. `reset`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/reset`

**Input:** `reset_request` (soap:body, use = literal) [Source code](#)

```
parameters type reset
```

**Output:** `reset_response` (soap:body, use = literal) [Source code](#)

```
parameters type resetResponse
```

### 133. `run_script`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/run_script`

**Input:** `run_script_request` (soap:body, use = literal) [Source code](#)

```
parameters type run_script
 ■ script_file_path type string
```

**Output:** `run_script_response` (soap:body, use = literal) [Source code](#)

```
parameters type run_scriptResponse
```

### 134. `save`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/save`

**Input:** `save_request` (soap:body, use = literal) [Source code](#)

```
parameters type save
 ■ file_path type string
```

**Output:** `save_response` (soap:body, use = literal) [Source code](#)

```
parameters type saveResponse
 ■ value type boolean
```

### 135. `set_action`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_action`

**Input:** `set_action_request` (soap:body, use = literal) [Source code](#)

```
parameters type set_action
 ■ value type action
 ■ no type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ is_active - optional; type boolean
 ■ action_category - optional; type string
 ■ action_type - optional; type action_action_type - type undefined with restriction - enum {
 'ACTING_ALTERNATIVELY', 'ACTING_DIFFERENTLY', 'ACTING_SIMULTANEOUSLY', 'DYNAMIC_LOAD_CASE'
 }
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ items - optional; type array_of_action_items
 ■ action_items - optional, unbounded; type action_items
 ■ no - optional; type int
 ■ load_case_item - optional; type int
 ■ acting_group_number - optional; type int
 ■ has_short_duration - optional; type boolean
 ■ has_duration_shorter_than_one_month - optional; type boolean
 ■ imposed_load_category - optional; type action_imposed_load_category - type undefined with restriction - enum {
 'IMPOSED_LOADS_CATEGORY_A', 'IMPOSED_LOADS_CATEGORY_B', 'IMPOSED_LOADS_CATEGORY_C',
 'IMPOSED_LOADS_CATEGORY_D', 'IMPOSED_LOADS_CATEGORY_E'
 }
 ■ has_short_duration_according_to_5132 - optional; type boolean
 ■ for_temperature_apply_coefficients - optional; type boolean
```

- short\_time\_variable\_action - optional; type *boolean*
- has\_inclusive\_action - optional; type *boolean*
- inclusive\_action - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_action\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_actionResponse*

### 136. set\_action\_combination

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_action\_combination

**Input:** set\_action\_combination\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_action\_combination*

- value type *action\_combination*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - attribute\_always\_editable - optional; type *string*
  - comment - optional; type *string*
  - design\_situation - optional; type *int*
  - items - optional; type *array\_of\_action\_combination\_items*
    - action\_combination\_items - optional, unbounded; type *action\_combination\_items*
      - no - optional; type *int*
      - action\_item - optional; type *int*
      - operator\_type - optional; type *operator\_type* - type *undefined* with restriction - enum { 'OPERATOR\_AND', 'OPERATOR\_NONE', 'OPERATOR\_OR' }
      - left\_parenthesis - optional; type *boolean*
      - right\_parenthesis - optional; type *boolean*
      - group\_factor - optional; type *double*
      - action\_factor - optional; type *double*
      - action\_load\_type - optional; type *action\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
      - group\_load\_type - optional; type *group\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
      - action - optional; type *int*
      - is\_leading - optional; type *boolean*
      - gamma - optional; type *double*
      - psi - optional; type *double*
      - xi - optional; type *double*
      - k\_fi - optional; type *double*
      - c\_esl - optional; type *double*
      - k\_def - optional; type *double*
      - psi\_0 - optional; type *double*
      - psi\_1 - optional; type *double*
      - psi\_2 - optional; type *double*
      - fi - optional; type *double*
      - gamma\_0 - optional; type *double*
      - alfa - optional; type *double*
      - k\_f - optional; type *double*
      - phi - optional; type *double*
      - rho - optional; type *double*
      - omega\_0 - optional; type *double*
      - gamma\_l\_1 - optional; type *double*
      - k\_creep - optional; type *double*
  - active - optional; type *boolean*
  - construction\_stage - optional; type *int*
  - combination\_type - optional; type *action\_combination\_combination\_type* - type *undefined* with restriction - enum { 'ENVELOPE\_PERMANENT', 'ENVELOPE\_TRANSIENT', 'GENERAL', 'SUPERPOSITION' }
  - generated\_load\_combinations - optional; type *array\_of\_int*
  - generated\_result\_combinations - optional; type *array\_of\_int*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_action\_combination\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_action\_combinationResponse*

### 137. set\_addon\_statuses

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_addon\_statuses

**Input:** set\_addon\_statuses\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_addon\_statuses*

- addon\_statuses type *addon\_list\_type*
  - design\_addons - optional; type *addon\_list\_design\_addons\_list\_type*
    - stress\_analysis\_active - optional; type *boolean*
    - concrete\_design\_active - optional; type *boolean*
    - steel\_design\_active - optional; type *boolean*
    - timber\_design\_active - optional; type *boolean*
    - aluminum\_design\_active - optional; type *boolean*
    - steel\_joints\_active - optional; type *boolean*
    - timber\_joints\_active - optional; type *boolean*
    - craneway\_design\_active - optional; type *boolean*
  - masonry\_design\_active - optional; type *boolean*
  - multilayer\_surfaces\_design\_active - optional; type *boolean*



- analysis - optional; type *addon\_list\_analysis\_list\_type*
  - material\_nonlinear\_analysis\_active - optional; type *boolean*
  - structure\_stability\_active - optional; type *boolean*
  - construction\_stages\_active - optional; type *boolean*
  - time\_dependent\_active - optional; type *boolean*
  - form\_finding\_active - optional; type *boolean*
  - cutting\_patterns\_active - optional; type *boolean*
  - torsional\_warping\_active - optional; type *boolean*
  - cost\_estimation\_active - optional; type *boolean*
- dynamic\_analysis\_settings - optional; type *addon\_list\_dynamic\_analysis\_settings\_list\_type*
  - modal\_active - optional; type *boolean*
  - spectral\_active - optional; type *boolean*
  - time\_history\_active - optional; type *boolean*
  - pushover\_active - optional; type *boolean*
  - harmonic\_response\_active - optional; type *boolean*
- special\_solutions - optional; type *addon\_list\_special\_solutions\_list\_type*
  - building\_model\_active - optional; type *boolean*
  - wind\_simulation\_active - optional; type *boolean*
  - geotechnical\_analysis\_active - optional; type *boolean*

**Output:** set\_addon\_statuses\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_addon\_statusesResponse*

### 138. set\_building\_story

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_building\_story

**Input:** set\_building\_story\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_building\_story*
- value type *building\_story*
    - no type *int*
    - type - optional; type *building\_story\_type* - type *undefined* with restriction - enum { 'TYPE\_STANDARD' }
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - story\_no - optional; type *int*
    - elevation - optional; type *double*
    - bottom\_elevation - optional; type *double*
    - height - optional; type *double*
    - modified\_height - optional; type *double*
    - thickness - optional; type *double*
    - comment - optional; type *string*
    - mass - optional; type *double*
    - center\_of\_gravity\_x - optional; type *double*
    - center\_of\_gravity\_y - optional; type *double*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_building\_story\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_building\_storyResponse*

### 139. set\_clipping\_box

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_clipping\_box

**Input:** set\_clipping\_box\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_clipping\_box*
- value type *clipping\_box*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - coordinate\_system - optional; type *int*
    - origin\_coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - origin\_coordinate\_x - optional; type *double*
    - origin\_coordinate\_y - optional; type *double*
    - origin\_coordinate\_z - optional; type *double*
    - dimensions - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - dimension\_x - optional; type *double*
    - dimension\_y - optional; type *double*
    - dimension\_z - optional; type *double*
    - comment - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_clipping\_box\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_clipping\_boxResponse*

### 140. set\_clipping\_plane

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_clipping\_plane

**Input:** set\_clipping\_plane\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_clipping_plane
 value type clipping_plane
 no type int
 type - optional; type clipping_plane_type - type undefined with restriction - enum { 'TYPE_2_POINTS_AND_ANGLE', 'TYPE_3_POINTS', 'TYPE_OFFSET_XYZ', 'TYPE_POINT_AND_3_ANGLES' }
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 coordinate_system - optional; type int
 invert_clipping_side - optional; type boolean
 origin_coordinates - optional; type vector_3d
 x type double
 y type double
 z type double
 origin_coordinate_x - optional; type double
 origin_coordinate_y - optional; type double
 origin_coordinate_z - optional; type double
 u_axis_point_coordinates - optional; type vector_3d
 x type double
 y type double
 z type double
 u_axis_point_coordinate_x - optional; type double
 u_axis_point_coordinate_y - optional; type double
 u_axis_point_coordinate_z - optional; type double
 clipping_plane_point_coordinates - optional; type vector_3d
 x type double
 y type double
 z type double
 clipping_plane_point_coordinate_x - optional; type double
 clipping_plane_point_coordinate_y - optional; type double
 clipping_plane_point_coordinate_z - optional; type double
 clipping_plane_angle - optional; type double
 rotation_angles_sequence - optional; type clipping_plane_rotation_angles_sequence - type undefined with restriction - enum { 'SEQUENCE_UVW', 'SEQUENCE_UWV', 'SEQUENCE_VUW', 'SEQUENCE_VWU', 'SEQUENCE_WUV', 'SEQUENCE_WVU', 'SEQUENCE_XYZ', 'SEQUENCE_XZY', 'SEQUENCE_YXZ', 'SEQUENCE_YZX', 'SEQUENCE_ZXY', 'SEQUENCE_ZYX' }
 rotation_angle_1 - optional; type double
 rotation_angle_2 - optional; type double
 rotation_angle_3 - optional; type double
 orientation - optional; type clipping_plane_orientation - type undefined with restriction - enum { 'ORIENTATION_PARALLEL_TO_XY', 'ORIENTATION_PARALLEL_TO_XZ', 'ORIENTATION_PARALLEL_TO_YZ' }
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_clipping\_plane\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_clipping_planeResponse
```

#### 141. set\_combination\_wizard

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_combination\_wizard

**Input:** set\_combination\_wizard\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_combination_wizard
 value type combination_wizard
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 static_analysis_settings - optional; type int
 generate_combinations - optional; type combination_wizard_generate_combinations - type undefined with restriction - enum { 'GENERATE_LOAD_COMBINATIONS', 'GENERATE_RESULT_COMBINATIONS' }
 has_stability_analysis - optional; type boolean
 stability_analysis_settings - optional; type int
 consider_imperfection_case - optional; type boolean
 generate_same_CO_without_IC - optional; type boolean
 consider_construction_stages - optional; type boolean
 user_defined_action_combinations - optional; type boolean
 favorable_permanent_actions - optional; type boolean
 consider_inclusive_or_exclusive_load_cases - optional; type boolean
 reduce_number_of_generated_combinations - optional; type boolean
 auxiliary_combinations - optional; type boolean
 generate_subcombinations_of_type_superposition - optional; type boolean
 comment - optional; type string
 consider_initial_state - optional; type boolean
 initial_state_case - optional; type int
 initial_state_definition_type - optional; type combination_wizard_initial_state_definition_type - type undefined with restriction - enum { 'DEFINITION_TYPE_FINAL_STATE', 'DEFINITION_TYPE_STIFFNESS', 'DEFINITION_TYPE_STRAINS', 'DEFINITION_TYPE_STRAINS_WITH_USER_DEFINED_FACTORS' }
 individual_factors_of_selected_objects_table - optional; type array_of_combination_wizard_individual_factors_of_selected_objects_table
 combination_wizard_individual_factors_of_selected_objects_table - optional, unbounded; type combination_wizard_individual_factors_of_selected_objects_table
 no - optional; type int
 object_type - optional; type object_type - type undefined with restriction - enum { 'INITIAL_STATE_FACTORS_OBJECT_TYPE_LINE_HINGE', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_LINE_WITH_SUPPORT', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_MEMBER', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_MEMBER_HINGE', 'INITIAL_STATE_FACTORS_OBJECT_TYPE_NODE_WITH_SUPPORT',
```

- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
- object\_list - optional; type *array\_of\_int*
- strain\_type - optional; type *strain\_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
- factor - optional; type *double*
- comment - optional; type *string*
- structure\_modification\_enabled - optional; type *boolean*
- structure\_modification - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_combination\_wizard\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_combination\_wizardResponse*

#### 142. set\_coordinate\_system

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_coordinate\_system

**Input:** set\_coordinate\_system\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_coordinate\_system*

- value type *coordinate\_system*
  - no type *int*
  - type - optional; type *coordinate\_system\_type* - type *undefined* with restriction - enum { 'TYPE\_2\_POINTS\_AND\_ANGLE', 'TYPE\_3\_POINTS', 'TYPE\_GLOBAL\_XYZ', 'TYPE\_OFFSET\_XYZ', 'TYPE\_POINT\_AND\_3\_ANGLES' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - origin\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - origin\_coordinate\_x - optional; type *double*
  - origin\_coordinate\_y - optional; type *double*
  - origin\_coordinate\_z - optional; type *double*
  - u\_axis\_point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - u\_axis\_point\_coordinate\_x - optional; type *double*
  - u\_axis\_point\_coordinate\_y - optional; type *double*
  - u\_axis\_point\_coordinate\_z - optional; type *double*
  - uw\_plane\_point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - uw\_plane\_point\_coordinate\_x - optional; type *double*
  - uw\_plane\_point\_coordinate\_y - optional; type *double*
  - uw\_plane\_point\_coordinate\_z - optional; type *double*
  - uw\_plane\_angle - optional; type *double*
  - rotation\_angles\_sequence - optional; type *coordinate\_system\_rotation\_angles\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - rotation\_angle\_1 - optional; type *double*
  - rotation\_angle\_2 - optional; type *double*
  - rotation\_angle\_3 - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_coordinate\_system\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_coordinate\_systemResponse*

#### 143. set\_cutting\_line\_setting

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_cutting\_line\_setting

**Input:** set\_cutting\_line\_setting\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_cutting\_line\_setting*

- value type *cutting\_line\_setting*
  - no type *int*
  - type - optional; type *cutting\_line\_setting\_type* - type *undefined* with restriction - enum { 'TYPE\_BOUNDARY\_LINE', 'TYPE\_WELDING\_LINE' }
  - comment - optional; type *string*
  - cutting\_line\_compensation - optional; type *double*
  - cutting\_line\_compensation\_enabled - optional; type *boolean*
  - cutting\_pattern\_allowance - optional; type *double*
  - name - optional; type *string*
  - weld\_allowance\_assignment - optional; type *cutting\_line\_setting\_weld\_allowance\_assignment* - type *undefined* with restriction - enum { 'WELD\_ALLOWANCE\_TO\_LEFT\_PATTERN', 'WELD\_ALLOWANCE\_TO\_RIGHT\_PATTERN', 'WELD\_ALLOWANCE\_TO\_NEIGHBOURING\_PATTERNS' }
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_cutting\_line\_setting\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_cutting\_line\_settingResponse*

#### 144. set\_cutting\_pattern

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_cutting\_pattern

**Input:** set\_cutting\_pattern\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_cutting_pattern
 value type cutting_pattern
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 material_change_enabled - optional; type boolean
 material - optional; type int
 pattern_orientation_category - optional; type cutting_pattern_pattern_orientation_category - type undefined with
restriction - enum { 'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_ANGULAR_ROTATION',
'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_COORDINATE_SYSTEM',
'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_LINES' }
 angular_rotation - optional; type double
 axis - optional; type cutting_pattern_axis - type undefined with restriction - enum { 'AXIS_X', 'AXIS_Y' }
 parallel_to_lines - optional; type array_of_int
 coordinate_system - optional; type int
 boundary_lines - optional; type array_of_int
 cutting_line_settings_table - optional; type array_of_cutting_pattern_cutting_line_settings_table
 cutting_pattern_cutting_line_settings_table - optional, unbounded; type
cutting_pattern_cutting_line_settings_table
 no - optional; type int
 line - optional; type int
 cutting_line_settings - optional; type int
 comment - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_cutting\_pattern\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_cutting_patternResponse
```

#### 145. set\_design\_situation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_design\_situation

**Input:** set\_design\_situation\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_design_situation
 value type design_situation
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 design_situation_type - optional; type int
 active - optional; type boolean
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 combination_wizard - optional; type int
 case_objects - optional; type array_of_design_situation_case_objects
 design_situation_case_objects - optional, unbounded; type int
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_design\_situation\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_design_situationResponse
```

#### 146. set\_design\_support

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_design\_support

**Input:** set\_design\_support\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_design_support
 value type design_support
 no type int
 type - optional; type design_support_type - type undefined with restriction - enum {
'DESIGN_SUPPORT_TYPE_CONCRETE', 'DESIGN_SUPPORT_TYPE_GENERAL',
'DESIGN_SUPPORT_TYPE_TIMBER' }
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 assigned_to_members - optional; type array_of_int
 assigned_to_member_sets - optional; type array_of_int
 assigned_to_nodes - optional; type array_of_int
 assigned_to_objects - optional; type string
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 activate_in_y - optional; type boolean
 activate_in_z - optional; type boolean
 concrete_monolithic_connection_y_enabled - optional; type boolean
 concrete_monolithic_connection_z_enabled - optional; type boolean
 concrete_ratio_of_moment_redistribution_y - optional; type double
 concrete_ratio_of_moment_redistribution_z - optional; type double
```

- design\_support\_orientation\_y - optional; type *design\_support\_design\_support\_orientation\_y* - type *undefined* with restriction - enum { 'DESIGN\_SUPPORT\_ORIENTATION\_Y\_AXIS\_BOTH', 'DESIGN\_SUPPORT\_ORIENTATION\_Y\_AXIS\_NEGATIVE', 'DESIGN\_SUPPORT\_ORIENTATION\_Y\_AXIS\_POSITIVE' }
- design\_support\_orientation\_z - optional; type *design\_support\_design\_support\_orientation\_z* - type *undefined* with restriction - enum { 'DESIGN\_SUPPORT\_ORIENTATION\_Z\_AXIS\_BOTH', 'DESIGN\_SUPPORT\_ORIENTATION\_Z\_AXIS\_NEGATIVE', 'DESIGN\_SUPPORT\_ORIENTATION\_Z\_AXIS\_POSITIVE' }
- direct\_support\_y\_enabled - optional; type *boolean*
- direct\_support\_z\_enabled - optional; type *boolean*
- inner\_support\_y\_enabled - optional; type *boolean*
- inner\_support\_z\_enabled - optional; type *boolean*
- limit\_of\_high\_bending\_stresses\_y - optional; type *double*
- limit\_of\_high\_bending\_stresses\_z - optional; type *double*
- support\_depth\_by\_section\_width\_of\_member\_y\_enabled - optional; type *boolean*
- support\_depth\_by\_section\_width\_of\_member\_z\_enabled - optional; type *boolean*
- support\_depth\_y - optional; type *double*
- support\_depth\_z - optional; type *double*
- support\_width\_y - optional; type *double*
- support\_width\_z - optional; type *double*
- timber\_allow\_higher\_deformation\_y\_enabled - optional; type *boolean*
- timber\_allow\_higher\_deformation\_z\_enabled - optional; type *boolean*
- timber\_calculation\_method\_y - optional; type *design\_support\_timber\_calculation\_method\_y* - type *undefined* with restriction - enum { 'CALCULATION\_METHOD\_ACC\_TO\_4\_2\_2\_2', 'CALCULATION\_METHOD\_ACC\_TO\_ANNEX\_C' }
- timber\_calculation\_method\_z - optional; type *design\_support\_timber\_calculation\_method\_z* - type *undefined* with restriction - enum { 'CALCULATION\_METHOD\_ACC\_TO\_4\_2\_2\_2', 'CALCULATION\_METHOD\_ACC\_TO\_ANNEX\_C' }
- timber\_check\_critical\_bearing\_y\_enabled - optional; type *boolean*
- timber\_check\_critical\_bearing\_z\_enabled - optional; type *boolean*
- timber\_compression\_design\_value\_y - optional; type *design\_support\_timber\_compression\_design\_value\_y* - type *undefined* with restriction - enum { 'COMPRESSION\_DESIGN\_VALUE\_0\_0\_2', 'COMPRESSION\_DESIGN\_VALUE\_0\_0\_4' }
- timber\_compression\_design\_value\_z - optional; type *design\_support\_timber\_compression\_design\_value\_z* - type *undefined* with restriction - enum { 'COMPRESSION\_DESIGN\_VALUE\_0\_0\_2', 'COMPRESSION\_DESIGN\_VALUE\_0\_0\_4' }
- timber\_factor\_of\_compression\_y - optional; type *double*
- timber\_factor\_of\_compression\_z - optional; type *double*
- timber\_load\_bearing\_capacity\_of\_screw\_y - optional; type *double*
- timber\_load\_bearing\_capacity\_of\_screw\_z - optional; type *double*
- timber\_number\_of\_screws\_in\_grain\_direction\_y - optional; type *int*
- timber\_number\_of\_screws\_in\_grain\_direction\_z - optional; type *int*
- timber\_number\_of\_screws\_in\_vertical\_direction\_y - optional; type *int*
- timber\_number\_of\_screws\_in\_vertical\_direction\_z - optional; type *int*
- timber\_spacing\_of\_rows\_y - optional; type *double*
- timber\_spacing\_of\_rows\_z - optional; type *double*
- timber\_spacing\_of\_screws\_y - optional; type *double*
- timber\_spacing\_of\_screws\_z - optional; type *double*
- timber\_stiffener\_parameters\_specification\_type\_y - optional; type *design\_support\_timber\_stiffener\_parameters\_specification\_type\_y* - type *undefined* with restriction - enum { 'STIFFENER\_PARAMETERS\_SPECIFICATION\_TYPE\_AUTOMATICALLY', 'STIFFENER\_PARAMETERS\_SPECIFICATION\_TYPE\_MANUALLY' }
- timber\_stiffener\_parameters\_specification\_type\_z - optional; type *design\_support\_timber\_stiffener\_parameters\_specification\_type\_z* - type *undefined* with restriction - enum { 'STIFFENER\_PARAMETERS\_SPECIFICATION\_TYPE\_AUTOMATICALLY', 'STIFFENER\_PARAMETERS\_SPECIFICATION\_TYPE\_MANUALLY' }
- timber\_stiffening\_elements\_type\_y - optional; type *design\_support\_timber\_stiffening\_elements\_type\_y* - type *undefined* with restriction - enum { 'STIFFENING\_ELEMENTS\_TYPE\_SCREWS' }
- timber\_stiffening\_elements\_type\_z - optional; type *design\_support\_timber\_stiffening\_elements\_type\_z* - type *undefined* with restriction - enum { 'STIFFENING\_ELEMENTS\_TYPE\_SCREWS' }
- timber\_stiffening\_elements\_y\_enabled - optional; type *boolean*
- timber\_stiffening\_elements\_z\_enabled - optional; type *boolean*
- timber\_ultimate\_strength\_of\_screw\_y - optional; type *double*
- timber\_ultimate\_strength\_of\_screw\_z - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_design\_support\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_design\_supportResponse*

## 147. set\_dimension

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_dimension

**Input:** set\_dimension\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_dimension*

- value type *dimension*
  - no type *int*
  - type - optional; type *dimension\_type* - type *undefined* with restriction - enum { 'DIMENSION\_TYPE\_ANGULAR', 'DIMENSION\_TYPE\_ARC\_LENGTH', 'DIMENSION\_TYPE\_DIAMETER', 'DIMENSION\_TYPE\_ELEVATION', 'DIMENSION\_TYPE\_LINEAR', 'DIMENSION\_TYPE\_RADIUS', 'DIMENSION\_TYPE\_SLOPE' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - display\_properties\_index - optional; type *int*
  - dimension\_line\_offset - optional; type *double*
  - is\_custom\_vertical\_position - optional; type *boolean*
  - custom\_vertical\_position - optional; type *dimension\_custom\_vertical\_position* - type *undefined* with restriction - enum { 'VERTICAL\_POSITION\_ABOVE', 'VERTICAL\_POSITION\_CENTER', 'VERTICAL\_POSITION\_UNDER' }
  - is\_custom\_horizontal\_position - optional; type *boolean*
  - custom\_horizontal\_position - optional; type *dimension\_custom\_horizontal\_position* - type *undefined* with restriction - enum { 'HORIZONTAL\_POSITION\_CENTER', 'HORIZONTAL\_POSITION\_LEFT', 'HORIZONTAL\_POSITION\_RIGHT' }
  - symbol - optional; type *string*
  - comment - optional; type *string*
  - measured\_length - optional; type *double*

- measured\_lengths - optional; type *array\_of\_int*
- measured\_angle - optional; type *double*
- measured\_angles - optional; type *array\_of\_int*
- measured\_slope - optional; type *double*
- is\_global\_dimension\_line\_offset - optional; type *boolean*
- linear\_coordinate\_system - optional; type *int*
- linear\_reference - optional; type *dimension\_linear\_reference* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH', 'REFERENCE\_PROJECTION\_X', 'REFERENCE\_PROJECTION\_Y', 'REFERENCE\_PROJECTION\_Z' }
- linear\_plane - optional; type *dimension\_linear\_plane* - type *undefined* with restriction - enum { 'PLANE\_FIRST', 'PLANE\_SECOND' }
- linear\_global\_dimension\_line\_offset - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- linear\_global\_dimension\_line\_offset\_x - optional; type *double*
- linear\_global\_dimension\_line\_offset\_y - optional; type *double*
- linear\_global\_dimension\_line\_offset\_z - optional; type *double*
- linear\_reference\_table - optional; type *array\_of\_dimension\_linear\_reference\_table*
  - dimension\_linear\_reference\_table - optional, unbounded; type *dimension\_linear\_reference\_table*
    - no - optional; type *int*
    - reference\_object\_type - optional; type *reference\_object\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_CONTROL\_POINT', 'REFERENCE\_TYPE\_LINE', 'REFERENCE\_TYPE\_MEMBER', 'REFERENCE\_TYPE\_NODE', 'REFERENCE\_TYPE\_POINT', 'REFERENCE\_TYPE\_POINT\_ON\_LINE', 'REFERENCE\_TYPE\_SURFACE' }
    - reference\_object - optional; type *int*
    - line\_relative\_distance - optional; type *double*
    - coordinate\_x - optional; type *double*
    - coordinate\_y - optional; type *double*
    - coordinate\_z - optional; type *double*
- arc\_length\_reference\_line - optional; type *int*
- arc\_length\_reference\_table - optional; type *array\_of\_dimension\_arc\_length\_reference\_table*
  - dimension\_arc\_length\_reference\_table - optional, unbounded; type *dimension\_arc\_length\_reference\_table*
    - no - optional; type *int*
    - reference\_object\_type - optional; type *reference\_object\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_CONTROL\_POINT', 'REFERENCE\_TYPE\_LINE', 'REFERENCE\_TYPE\_MEMBER', 'REFERENCE\_TYPE\_NODE', 'REFERENCE\_TYPE\_POINT', 'REFERENCE\_TYPE\_POINT\_ON\_LINE', 'REFERENCE\_TYPE\_SURFACE' }
    - reference\_object - optional; type *int*
    - line\_relative\_distance - optional; type *double*
    - coordinate\_x - optional; type *double*
    - coordinate\_y - optional; type *double*
    - coordinate\_z - optional; type *double*
- arc\_length\_angle\_greater\_than\_180 - optional; type *boolean*
- angular\_reference\_table - optional; type *array\_of\_dimension\_angular\_reference\_table*
  - dimension\_angular\_reference\_table - optional, unbounded; type *dimension\_angular\_reference\_table*
    - no - optional; type *int*
    - reference\_object\_type - optional; type *reference\_object\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_CONTROL\_POINT', 'REFERENCE\_TYPE\_LINE', 'REFERENCE\_TYPE\_MEMBER', 'REFERENCE\_TYPE\_NODE', 'REFERENCE\_TYPE\_POINT', 'REFERENCE\_TYPE\_POINT\_ON\_LINE', 'REFERENCE\_TYPE\_SURFACE' }
    - reference\_object - optional; type *int*
    - line\_relative\_distance - optional; type *double*
    - coordinate\_x - optional; type *double*
    - coordinate\_y - optional; type *double*
    - coordinate\_z - optional; type *double*
- angular\_quadrant - optional; type *dimension\_angular\_quadrant* - type *undefined* with restriction - enum { 'ANGULAR\_QUADRANT\_LEFT', 'ANGULAR\_QUADRANT\_NEGATIVE', 'ANGULAR\_QUADRANT\_POSITIVE', 'ANGULAR\_QUADRANT\_RIGHT' }
- angular\_angle\_greater\_than\_180 - optional; type *boolean*
- radius\_diameter\_reference\_line - optional; type *int*
- radius\_diameter\_is\_target\_point - optional; type *boolean*
- radius\_diameter\_target\_point\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- radius\_diameter\_target\_point\_coordinate\_x - optional; type *double*
- radius\_diameter\_target\_point\_coordinate\_y - optional; type *double*
- radius\_diameter\_target\_point\_coordinate\_z - optional; type *double*
- radius\_diameter\_position\_on\_line - optional; type *double*
- slope\_coordinate\_system - optional; type *int*
- slope\_plane - optional; type *dimension\_slope\_plane* - type *undefined* with restriction - enum { 'SLOPE\_PLANE\_XY', 'SLOPE\_PLANE\_XZ', 'SLOPE\_PLANE\_YZ' }
- slope\_reference\_line - optional; type *int*
- slope\_direction - optional; type *dimension\_slope\_direction* - type *undefined* with restriction - enum { 'SLOPE\_DIRECTION\_DOWNWARD', 'SLOPE\_DIRECTION\_UPWARD' }
- slope\_refer\_distance\_from\_line\_end - optional; type *boolean*
- slope\_position\_is\_relative - optional; type *boolean*
- slope\_position\_absolute - optional; type *double*
- slope\_position\_relative - optional; type *double*
- elevation\_reference\_object\_type - optional; type *dimension\_elevation\_reference\_object\_type* - type *undefined* with restriction - enum { 'ELEVATION\_REFERENCE\_OBJECT\_TYPE\_CONTROL\_POINT', 'ELEVATION\_REFERENCE\_OBJECT\_TYPE\_NODE', 'ELEVATION\_REFERENCE\_OBJECT\_TYPE\_POINT', 'ELEVATION\_REFERENCE\_OBJECT\_TYPE\_SURFACE' }
- elevation\_reference\_surface - optional; type *int*
- elevation\_reference\_node - optional; type *int*
- elevation\_reference\_control\_point - optional; type *int*
- elevation\_reference\_point\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elevation\_reference\_point\_coordinate\_x - optional; type *double*
- elevation\_reference\_point\_coordinate\_y - optional; type *double*
- elevation\_reference\_point\_coordinate\_z - optional; type *double*
- elevation\_distance\_from\_picked\_position - optional; type *double*
- elevation\_rotation\_around\_z - optional; type *double*
- elevation\_reference\_level\_height - optional; type *double*

- elevation\_is\_altitude - optional; type *boolean*
- elevation\_altitude - optional; type *double*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_dimension\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_dimensionResponse*

#### 148. set\_formula

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_formula

**Input:** set\_formula\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_formula*

- object\_location type *object\_location*
  - type type *object\_types* - type *undefined* with restriction - enum { 'E\_OBJECT\_TYPE\_ACTION', 'E\_OBJECT\_TYPE\_ACTION\_COMBINATION', 'E\_OBJECT\_TYPE\_BUILDING\_STORY', 'E\_OBJECT\_TYPE\_CLIPPING\_BOX', 'E\_OBJECT\_TYPE\_CLIPPING\_PLANE', 'E\_OBJECT\_TYPE\_COMBINATION\_WIZARD', 'E\_OBJECT\_TYPE\_COORDINATE\_SYSTEM', 'E\_OBJECT\_TYPE\_CUTTING\_LINE\_SETTING', 'E\_OBJECT\_TYPE\_CUTTING\_PATTERN', 'E\_OBJECT\_TYPE\_DESIGN\_SITUATION', 'E\_OBJECT\_TYPE\_DESIGN\_SUPPORT', 'E\_OBJECT\_TYPE\_DIMENSION', 'E\_OBJECT\_TYPE\_FREE\_CIRCULAR\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_CONCENTRATED\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_POLYGON\_LOAD', 'E\_OBJECT\_TYPE\_FREE\_RECTANGULAR\_LOAD', 'E\_OBJECT\_TYPE\_GLOBAL\_PARAMETER', 'E\_OBJECT\_TYPE\_IMPERFECTION\_CASE', 'E\_OBJECT\_TYPE\_IMPOSED\_LINE\_DEFORMATION', 'E\_OBJECT\_TYPE\_IMPOSED\_NODAL\_DEFORMATION', 'E\_OBJECT\_TYPE\_INTERSECTION', 'E\_OBJECT\_TYPE\_LINE', 'E\_OBJECT\_TYPE\_LINE\_GRID', 'E\_OBJECT\_TYPE\_LINE\_HINGE', 'E\_OBJECT\_TYPE\_LINE\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_LINE\_SET', 'E\_OBJECT\_TYPE\_LINE\_SET\_LOAD', 'E\_OBJECT\_TYPE\_LINE\_SUPPORT', 'E\_OBJECT\_TYPE\_LINE\_WELDED\_JOINT', 'E\_OBJECT\_TYPE\_LOAD\_CASE', 'E\_OBJECT\_TYPE\_LOAD\_COMBINATION', 'E\_OBJECT\_TYPE\_MATERIAL', 'E\_OBJECT\_TYPE\_MEMBER', 'E\_OBJECT\_TYPE\_MEMBER\_DEFINABLE\_STIFFNESS', 'E\_OBJECT\_TYPE\_MEMBER\_ECCENTRICITY', 'E\_OBJECT\_TYPE\_MEMBER\_HINGE', 'E\_OBJECT\_TYPE\_MEMBER\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_NONLINEARITY', 'E\_OBJECT\_TYPE\_MEMBER\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_RESULT\_INTERMEDIATE\_POINT', 'E\_OBJECT\_TYPE\_MEMBER\_SET', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_IMPERFECTION', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_LOAD', 'E\_OBJECT\_TYPE\_MEMBER\_SET\_REPRESENTATIVE', 'E\_OBJECT\_TYPE\_MEMBER\_STIFFNESS\_MODIFICATION', 'E\_OBJECT\_TYPE\_MEMBER\_SUPPORT', 'E\_OBJECT\_TYPE\_MEMBER\_TRANSVERSE\_STIFFENER', 'E\_OBJECT\_TYPE\_NODAL\_LOAD', 'E\_OBJECT\_TYPE\_NODAL\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_NODAL\_SUPPORT', 'E\_OBJECT\_TYPE\_NODE', 'E\_OBJECT\_TYPE\_NOTE', 'E\_OBJECT\_TYPE\_OBJECT\_SNAP', 'E\_OBJECT\_TYPE\_OPENING', 'E\_OBJECT\_TYPE\_OPENING\_LOAD', 'E\_OBJECT\_TYPE\_RESULT\_COMBINATION', 'E\_OBJECT\_TYPE\_RESULT\_SECTION', 'E\_OBJECT\_TYPE\_RIGID\_LINK', 'E\_OBJECT\_TYPE\_SECTION', 'E\_OBJECT\_TYPE\_SOIL\_MASSIF', 'E\_OBJECT\_TYPE\_SOIL\_SAMPLE', 'E\_OBJECT\_TYPE\_SOLID', 'E\_OBJECT\_TYPE\_SOLID\_CONTACTS', 'E\_OBJECT\_TYPE\_SOLID\_GAS', 'E\_OBJECT\_TYPE\_SOLID\_LOAD', 'E\_OBJECT\_TYPE\_SOLID\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_SOLID\_SET', 'E\_OBJECT\_TYPE\_SOLID\_SET\_LOAD', 'E\_OBJECT\_TYPE\_SPECTRAL\_ANALYSIS\_SETTINGS', 'E\_OBJECT\_TYPE\_STATIC\_ANALYSIS\_SETTINGS', 'E\_OBJECT\_TYPE\_STRUCTURE\_MODIFICATION', 'E\_OBJECT\_TYPE\_SURFACE', 'E\_OBJECT\_TYPE\_SURFACES\_CONTACT', 'E\_OBJECT\_TYPE\_SURFACES\_CONTACT\_TYPE', 'E\_OBJECT\_TYPE\_SURFACE\_ECCENTRICITY', 'E\_OBJECT\_TYPE\_SURFACE\_IMPERFECTION', 'E\_OBJECT\_TYPE\_SURFACE\_LOAD', 'E\_OBJECT\_TYPE\_SURFACE\_MESH\_REFINEMENT', 'E\_OBJECT\_TYPE\_SURFACE\_RESULTS\_ADJUSTMENT', 'E\_OBJECT\_TYPE\_SURFACE\_SET', 'E\_OBJECT\_TYPE\_SURFACE\_SET\_IMPERFECTION', 'E\_OBJECT\_TYPE\_SURFACE\_SET\_LOAD', 'E\_OBJECT\_TYPE\_SURFACE\_STIFFNESS\_MODIFICATION', 'E\_OBJECT\_TYPE\_SURFACE\_SUPPORT', 'E\_OBJECT\_TYPE\_TERRAIN', 'E\_OBJECT\_TYPE\_THICKNESS', 'E\_OBJECT\_TYPE\_VISUAL\_OBJECT', 'E\_OBJECT\_TYPE\_WIND\_PROFILE', 'E\_OBJECT\_TYPE\_WIND\_SIMULATION', 'E\_OBJECT\_TYPE\_WIND\_SIMULATION\_ANALYSIS\_SETTINGS' }
  - no type *int*
  - parent\_no - optional; type *int*
- object\_parameter\_location type *object\_parameter\_location\_type*
  - attribute type *string*
  - parameter\_path\_in\_nested\_models\_hierarchy - optional; type *parameter\_path\_in\_nested\_models\_hierarchy\_type*
    - node - optional, unbounded; type *node\_of\_parameter\_path\_in\_nested\_models\_hierarchy\_type*
      - row\_path type *string*
      - column\_string\_id type *string*
  - formula type *string*

**Output:** set\_formula\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_formulaResponse*

#### 149. set\_free\_circular\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_free\_circular\_load

**Input:** set\_free\_circular\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_circular\_load*

- load\_case\_no type *int*
- value type *free\_circular\_load*
  - no type *int*
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_circular\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_circular\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE' }

```
'LOAD_DIRECTION_LOCAL_X', 'LOAD_DIRECTION_LOCAL_Y', 'LOAD_DIRECTION_LOCAL_Z',
'LOAD_DIRECTION_USER_DEFINED_U_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_U_TRUE',
'LOAD_DIRECTION_USER_DEFINED_V_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_V_TRUE',
'LOAD_DIRECTION_USER_DEFINED_W_PROJECTED', 'LOAD_DIRECTION_USER_DEFINED_W_TRUE' }
```

- load\_acting\_region\_from - optional; type *double*
- load\_acting\_region\_to - optional; type *double*
- load\_distribution - optional; type *free\_circular\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_UNIFORM' }
- magnitude\_uniform - optional; type *double*
- magnitude\_center - optional; type *double*
- magnitude\_radius - optional; type *double*
- load\_location\_x - optional; type *double*
- load\_location\_y - optional; type *double*
- load\_location\_radius - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_free\_circular\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_circular\_loadResponse*

#### 150. set\_free\_concentrated\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_free\_concentrated\_load

**Input:** set\_free\_concentrated\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_concentrated\_load*

- load\_case\_no type *int*
- value type *free\_concentrated\_load*
  - no type *int*
  - load\_type - optional; type *free\_concentrated\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MOMENT' }
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_concentrated\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_concentrated\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X', 'LOAD\_DIRECTION\_GLOBAL\_Y', 'LOAD\_DIRECTION\_GLOBAL\_Z', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U', 'LOAD\_DIRECTION\_USER\_DEFINED\_V', 'LOAD\_DIRECTION\_USER\_DEFINED\_W' }
  - load\_acting\_region\_from - optional; type *double*
  - load\_acting\_region\_to - optional; type *double*
  - magnitude - optional; type *double*
  - load\_location\_x - optional; type *double*
  - load\_location\_y - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_free\_concentrated\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_concentrated\_loadResponse*

#### 151. set\_free\_line\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_free\_line\_load

**Input:** set\_free\_line\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_line\_load*

- load\_case\_no type *int*
- value type *free\_line\_load*
  - no type *int*
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_line\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_line\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }
  - load\_acting\_region\_from - optional; type *double*
  - load\_acting\_region\_to - optional; type *double*
  - load\_distribution - optional; type *free\_line\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_UNIFORM' }
  - magnitude\_uniform - optional; type *double*
  - magnitude\_first - optional; type *double*
  - magnitude\_second - optional; type *double*
  - load\_location\_first\_x - optional; type *double*
  - load\_location\_first\_y - optional; type *double*



- load\_location\_second\_x - optional; type *double*
- load\_location\_second\_y - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_free\_line\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_line\_loadResponse*

## 152. set\_free\_polygon\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_free\_polygon\_load

**Input:** set\_free\_polygon\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_polygon\_load*

- load\_case\_no type *int*
- value type *free\_polygon\_load*
  - no type *int*
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_polygon\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_polygon\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }
  - load\_acting\_region\_from - optional; type *double*
  - load\_acting\_region\_to - optional; type *double*
  - load\_distribution - optional; type *free\_polygon\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR', 'LOAD\_DISTRIBUTION\_LINEAR\_FIRST', 'LOAD\_DISTRIBUTION\_LINEAR\_SECOND', 'LOAD\_DISTRIBUTION\_UNIFORM' }
  - magnitude\_uniform - optional; type *double*
  - magnitude\_linear\_1 - optional; type *double*
  - magnitude\_linear\_2 - optional; type *double*
  - magnitude\_linear\_3 - optional; type *double*
  - magnitude\_linear\_location\_1 - optional; type *int*
  - magnitude\_linear\_location\_2 - optional; type *int*
  - magnitude\_linear\_location\_3 - optional; type *int*
  - load\_location - optional; type *array\_of\_free\_polygon\_load\_load\_location*
    - free\_polygon\_load\_load\_location - optional, unbounded; type *free\_polygon\_load\_load\_location*
      - no - optional; type *int*
      - first\_coordinate - optional; type *double*
      - second\_coordinate - optional; type *double*
      - magnitude - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_free\_polygon\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_polygon\_loadResponse*

## 153. set\_free\_rectangular\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_free\_rectangular\_load

**Input:** set\_free\_rectangular\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_rectangular\_load*

- load\_case\_no type *int*
- value type *free\_rectangular\_load*
  - no type *int*
  - surfaces - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *int*
  - load\_projection - optional; type *free\_rectangular\_load\_load\_projection* - type *undefined* with restriction - enum { 'LOAD\_PROJECTION\_XY\_OR\_UV', 'LOAD\_PROJECTION\_XZ\_OR\_UW', 'LOAD\_PROJECTION\_YZ\_OR\_VW' }
  - load\_direction - optional; type *free\_rectangular\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_USER\_DEFINED\_W\_TRUE' }
  - load\_acting\_region\_from - optional; type *double*
  - load\_acting\_region\_to - optional; type *double*
  - load\_distribution - optional; type *free\_rectangular\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR\_FIRST', 'LOAD\_DISTRIBUTION\_LINEAR\_SECOND', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_VARYING\_ALONG\_PERIMETER', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z\_AND\_ALONG\_PERIMETER' }
  - magnitude\_uniform - optional; type *double*

- magnitude\_linear\_first - optional; type *double*
- magnitude\_linear\_second - optional; type *double*
- load\_location\_first\_x - optional; type *double*
- load\_location\_first\_y - optional; type *double*
- load\_location\_second\_x - optional; type *double*
- load\_location\_second\_y - optional; type *double*
- load\_location\_rectangle - optional; type *free\_rectangular\_load\_load\_location\_rectangle* - type *undefined* with restriction - enum { 'LOAD\_LOCATION\_RECTANGLE\_CENTER\_AND\_SIDES', 'LOAD\_LOCATION\_RECTANGLE\_CORNER\_POINTS' }
- load\_location\_center\_x - optional; type *double*
- load\_location\_center\_y - optional; type *double*
- load\_location\_center\_side\_a - optional; type *double*
- load\_location\_center\_side\_b - optional; type *double*
- load\_location\_rotation - optional; type *double*
- load\_varying\_in\_z\_parameters - optional; type *array\_of\_free\_rectangular\_load\_load\_varying\_in\_z\_parameters*
  - free\_rectangular\_load\_load\_varying\_in\_z\_parameters - optional, unbounded; type *free\_rectangular\_load\_load\_varying\_in\_z\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - recalculated\_magnitude - optional; type *double*
    - factor - optional; type *double*
    - note - optional; type *string*
- load\_varying\_along\_perimeter\_parameters - optional; type *array\_of\_free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters*
  - free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters - optional, unbounded; type *free\_rectangular\_load\_load\_varying\_along\_perimeter\_parameters*
    - no - optional; type *int*
    - alpha - optional; type *double*
    - recalculated\_magnitude - optional; type *double*
    - factor - optional; type *double*
    - note - optional; type *string*
- load\_varying\_in\_z\_parameters\_sorted - optional; type *boolean*
- load\_varying\_along\_perimeter\_parameters\_sorted - optional; type *boolean*
- load\_varying\_along\_perimeter\_z\_index - optional; type *int*
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_start\_angle - optional; type *double*
- comment - optional; type *string*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_free\_rectangular\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_free\_rectangular\_loadResponse*

#### 154. set\_global\_parameter

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_global\_parameter

**Input:** set\_global\_parameter\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_global\_parameter*

- value type *global\_parameter*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - attribute\_always\_editable - optional; type *string*
  - symbol - optional; type *string*
  - unit\_group - optional; type *global\_parameter\_unit\_group* - type *undefined* with restriction - enum { 'ANGLE', 'AREA', 'DENSITY', 'DIMENSIONLESS', 'DYNAMIC\_INCREASE\_FACTOR', 'EG\_MODULE', 'FRICTION\_COEFFICIENT', 'GEOGRAPHIC\_COORDINATES', 'GRAVITATIONAL\_ACCELERATION', 'LENGTH', 'LOADS\_ANGULAR\_ACCELERATION', 'LOADS\_ANGULAR\_VELOCITY', 'LOADS\_AREA\_MASS', 'LOADS\_AXIAL\_STRAIN', 'LOADS\_DENSITY', 'LOADS\_DISPLACEMENT', 'LOADS\_DISPLACEMENT\_PER\_UNIT\_LENGTH', 'LOADS\_FORCE', 'LOADS\_FORCE\_PER\_UNIT\_LENGTH', 'LOADS\_IMPOSED\_DISPLACEMENT', 'LOADS\_IMPOSED\_ROTATION', 'LOADS\_KINEMATIC\_VISCOSITY', 'LOADS\_KINETIC\_ENERGY', 'LOADS\_LENGTH', 'LOADS\_MASS', 'LOADS\_MOMENT', 'LOADS\_MOMENT\_PER\_UNIT\_LENGTH', 'LOADS\_PRECAMBER', 'LOADS\_PRESSURE', 'LOADS\_RELATIVE\_LENGTH', 'LOADS\_ROTATION', 'LOADS\_ROTATION\_PER\_UNIT\_LENGTH', 'LOADS\_SOLID\_TYPE\_LOAD', 'LOADS\_SPECIFIC\_ENERGY', 'LOADS\_SURFACE\_TYPE\_LOAD', 'LOADS\_TEMPERATURE', 'LOADS\_TEMPERATURE\_CHANGE', 'LOADS\_TURBULENCE DISSIPATION\_RATE', 'LOADS\_VELOCITY', 'MASS', 'MASS\_MOMENT\_PER\_UNIT\_AREA', 'MATERIAL\_ANGLE', 'MATERIAL\_DEFORMATION', 'MATERIAL\_FACTOR', 'MATERIAL\_QUANTITY\_FLOAT\_PRECISION\_2', 'MATERIAL\_QUANTITY\_INTEGER', 'MATERIAL\_SPECIFIC\_WEIGHT', 'MATERIAL\_THICKNESS', 'PARTIAL\_FACTOR', 'POISSONS\_RATIO', 'PRECISION\_FACTOR', 'QUANTITY', 'QUANTITY\_INTEGER', 'RATIO', 'RELATIVE\_LENGTH', 'SECTION\_ANGLE', 'SECTION\_AREA', 'SECTION\_BIMOMENT', 'SECTION\_COMPLIANCE', 'SECTION\_DIMENSION', 'SECTION\_EFFECTIVE\_AREA', 'SECTION\_EFFECTIVE\_SECOND\_MOMENT\_OF\_AREA', 'SECTION\_FORCE', 'SECTION\_MOMENT', 'SECTION\_MOMENT\_OF\_INERTIA', 'SECTION\_NORMALIZED\_WARPING\_CONSTANT', 'SECTION\_PERIMETER', 'SECTION\_QUANTITY', 'SECTION\_SECTION\_FACTOR', 'SECTION\_SECTION\_MODULUS', 'SECTION\_STATIC\_MOMENT\_OF\_AREA', 'SECTION\_TENSION\_FIELD\_COEFFICIENT\_1', 'SECTION\_TENSION\_FIELD\_COEFFICIENT\_2', 'SECTION\_UNIT\_STRESSES', 'SECTION\_UNIT\_WARPING\_FUNCTION', 'SECTION\_WARPING\_CONSTANT',

```
'SECTION_WARPING_STATICAL_MOMENT', 'SELF_WEIGHT_FACTOR', 'STIFFNESS_MULTIPLICATION_FACTOR',
'STRAIN', 'STRAIN_RATE', 'STRESSES', 'THERMAL_EXPANSION_COEFFICIENT', 'THICKNESS', 'TIME', 'VOLUME',
'WEIGHT_AND_KNOT' }
▪ definition_type - optional; type global_parameter_definition_type - type undefined with restriction - enum {
'DEFINITION_TYPE_FORMULA', 'DEFINITION_TYPE_VALUE' }
▪ value - optional; type double
▪ unit - optional; type string
▪ formula - optional; type string
▪ min - optional; type double
▪ max - optional; type double
▪ increment - optional; type double
▪ steps - optional; type int
▪ comment - optional; type string
▪ id_for_export_import - optional; type string
▪ metadata_for_export_import - optional; type string
```

**Output:** set\_global\_parameter\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_global\_parameterResponse*

## 155. set\_imperfection\_case

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_imperfection\_case

**Input:** set\_imperfection\_case\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_imperfection_case
▪ value type imperfection_case
▪ no type int
▪ type - optional; type imperfection_case_type - type undefined with restriction - enum {
'IMPERFECTION_TYPE_BUCKLING_MODE', 'IMPERFECTION_TYPE_DYNAMIC_EIGENMODE',
'IMPERFECTION_TYPE_IMPERFECTION_CASES_GROUP', 'IMPERFECTION_TYPE_INITIAL_SWAY_VIA_TABLE',
'IMPERFECTION_TYPE_LOCAL_IMPERFECTIONS',
'IMPERFECTION_TYPE_NOTIONAL_LOADS_FROM_LOAD_CASE', 'IMPERFECTION_TYPE_STATIC_DEFORMATION'
}
▪ user_defined_name_enabled - optional; type boolean
▪ name - optional; type string
▪ assigned_to_load_cases - optional; type array_of_int
▪ assigned_to_load_combinations - optional; type array_of_int
▪ is_active - optional; type boolean
▪ assign_to_combinations_without_assigned_imperfection_case - optional; type boolean
▪ direction - optional; type imperfection_case_direction - type undefined with restriction - enum {
'IMPERFECTION_CASE_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_NEGATIVE',
'IMPERFECTION_CASE_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE',
'IMPERFECTION_CASE_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_NEGATIVE',
'IMPERFECTION_CASE_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE',
'IMPERFECTION_CASE_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_NEGATIVE',
'IMPERFECTION_CASE_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE',
'IMPERFECTION_CASE_DIRECTION_LOCAL_X', 'IMPERFECTION_CASE_DIRECTION_LOCAL_Y',
'IMPERFECTION_CASE_DIRECTION_LOCAL_Y_NEGATIVE', 'IMPERFECTION_CASE_DIRECTION_LOCAL_Z',
'IMPERFECTION_CASE_DIRECTION_LOCAL_Z_NEGATIVE', 'IMPERFECTION_CASE_DIRECTION_SPATIAL' }
▪ direction_for_level_direction - optional; type imperfection_case_direction_for_level_direction - type undefined
with restriction - enum { 'DIRECTION_X', 'DIRECTION_XY', 'DIRECTION_XZ', 'DIRECTION_Y', 'DIRECTION_YZ',
'DIRECTION_Z' }
▪ coordinate_system - optional; type int
▪ load_case_for_notional_loads - optional; type int
▪ sway_coefficients_reciprocal - optional; type boolean
▪ level_imperfections - optional; type array_of_imperfection_case_level_imperfections
▪ imperfection_case_level_imperfections - optional, unbounded; type imperfection_case_level_imperfections
▪ no - optional; type int
▪ level - optional; type double
▪ e_1 - optional; type double
▪ theta_1 - optional; type double
▪ e_2 - optional; type double
▪ theta_2 - optional; type double
▪ comment - optional; type string
▪ source - optional; type imperfection_case_source - type undefined with restriction - enum {
'SOURCE_TYPE_AUTOMATICALLY', 'SOURCE_TYPE_LOAD_CASE', 'SOURCE_TYPE_LOAD_COMBINATION',
'SOURCE_TYPE_OWN_LOAD_CASE_OR_COMBINATION' }
▪ shape_from_load_case - optional; type int
▪ shape_from_load_combination - optional; type int
▪ buckling_shape - optional; type int
▪ delta_zero - optional; type double
▪ magnitude_assignment_type - optional; type imperfection_case_magnitude_assignment_type - type undefined
with restriction - enum { 'MAGNITUDE_ASSIGNMENT_LOCATION_WITH_LARGEST_DISPLACEMENT',
'MAGNITUDE_ASSIGNMENT_SPECIFIC_NODE' }
▪ reference_node - optional; type int
▪ amount_of_modes_to_investigate - optional; type int
▪ eigenmode_automatically - optional; type boolean
▪ imperfection_cases_items - optional; type array_of_imperfection_case_imperfection_cases_items
▪ imperfection_case_imperfection_cases_items - optional, unbounded; type
imperfection_case_imperfection_cases_items
▪ no - optional; type int
▪ name - optional; type int
▪ factor - optional; type double
▪ operator_type - optional; type operator_type - type undefined with restriction - enum {
'OPERATOR_AND', 'OPERATOR_NONE', 'OPERATOR_OR' }
▪ comment - optional; type string
▪ is_generated - optional; type boolean
▪ generating_object_info - optional; type string
▪ id_for_export_import - optional; type string
▪ metadata_for_export_import - optional; type string
```

**Output:** set\_imperfection\_case\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_imperfection\\_caseResponse](#)

#### 156. set\_imposed\_line\_deformation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/set\\_imposed\\_line\\_deformation](http://localhost:8082/set_imposed_line_deformation)

**Input:** set\_imposed\_line\_deformation\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_imposed_line_deformation
 ■ load_case_no type int
 ■ value type imposed_line_deformation
 ■ no type int
 ■ lines - optional; type array_of_int
 ■ load_case - optional; type int
 ■ imposed_displacement_line_start - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ imposed_displacement_line_start_x - optional; type double
 ■ imposed_displacement_line_start_y - optional; type double
 ■ imposed_displacement_line_start_z - optional; type double
 ■ imposed_displacement_line_end - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ imposed_displacement_line_end_x - optional; type double
 ■ imposed_displacement_line_end_y - optional; type double
 ■ imposed_displacement_line_end_z - optional; type double
 ■ imposed_rotation_line_start - optional; type double
 ■ imposed_rotation_line_end - optional; type double
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

**Output:** set\_imposed\_line\_deformation\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_imposed\\_line\\_deformationResponse](#)

#### 157. set\_imposed\_nodal\_deformation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/set\\_imposed\\_nodal\\_deformation](http://localhost:8082/set_imposed_nodal_deformation)

**Input:** set\_imposed\_nodal\_deformation\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_imposed_nodal_deformation
 ■ load_case_no type int
 ■ value type imposed_nodal_deformation
 ■ no type int
 ■ nodes - optional; type array_of_int
 ■ load_case - optional; type int
 ■ imposed_displacement - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ imposed_displacement_x - optional; type double
 ■ imposed_displacement_y - optional; type double
 ■ imposed_displacement_z - optional; type double
 ■ imposed_rotation - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ imposed_rotation_x - optional; type double
 ■ imposed_rotation_y - optional; type double
 ■ imposed_rotation_z - optional; type double
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

**Output:** set\_imposed\_nodal\_deformation\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_imposed\\_nodal\\_deformationResponse](#)

#### 158. set\_intersection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/set\\_intersection](http://localhost:8082/set_intersection)

**Input:** set\_intersection\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_intersection
 ■ value type intersection
 ■ no type int
 ■ comment - optional; type string
 ■ generated_lines - optional; type array_of_int
 ■ generated_nodes - optional; type array_of_int
 ■ surface_a - optional; type int
```

- surface\_b - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_intersection\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_intersectionResponse*

## 159. set\_line

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_line

**Input:** set\_line\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line*

- value type *line*
  - no type *int*
  - type - optional; type *line\_type* - type *undefined* with restriction - enum { 'TYPE\_ARC', 'TYPE\_CIRCLE', 'TYPE\_CUT\_VIA\_SECTION', 'TYPE\_CUT\_VIA\_TWO\_LINES', 'TYPE\_ELLIPSE', 'TYPE\_ELLIPTICAL\_ARC', 'TYPE\_NURBS', 'TYPE\_PARABOLA', 'TYPE\_POLYLINE', 'TYPE\_SPLINE' }
  - definition\_nodes - optional; type *array\_of\_int*
  - length - optional; type *double*
  - comment - optional; type *string*
  - arc\_first\_node - optional; type *int*
  - arc\_second\_node - optional; type *int*
  - arc\_control\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - arc\_control\_point\_x - optional; type *double*
  - arc\_control\_point\_y - optional; type *double*
  - arc\_control\_point\_z - optional; type *double*
  - arc\_center - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - arc\_center\_x - optional; type *double*
  - arc\_center\_y - optional; type *double*
  - arc\_center\_z - optional; type *double*
  - arc\_radius - optional; type *double*
  - arc\_height - optional; type *double*
  - arc\_alpha - optional; type *double*
  - arc\_alpha\_adjustment\_target - optional; type *line\_arc\_alpha\_adjustment\_target* - type *undefined* with restriction - enum { 'ALPHA\_ADJUSTMENT\_TARGET\_ARC\_CONTROL\_POINT', 'ALPHA\_ADJUSTMENT\_TARGET\_BEGINNING\_OF\_ARC', 'ALPHA\_ADJUSTMENT\_TARGET\_END\_OF\_ARC' }
  - circle\_center - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - circle\_center\_coordinate\_1 - optional; type *double*
  - circle\_center\_coordinate\_2 - optional; type *double*
  - circle\_center\_coordinate\_3 - optional; type *double*
  - circle\_normal - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - circle\_normal\_coordinate\_1 - optional; type *double*
  - circle\_normal\_coordinate\_2 - optional; type *double*
  - circle\_normal\_coordinate\_3 - optional; type *double*
  - circle\_rotation - optional; type *double*
  - circle\_node - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - circle\_radius - optional; type *double*
  - ellipse\_first\_node - optional; type *int*
  - ellipse\_second\_node - optional; type *int*
  - ellipse\_control\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - ellipse\_control\_point\_x - optional; type *double*
  - ellipse\_control\_point\_y - optional; type *double*
  - ellipse\_control\_point\_z - optional; type *double*
  - elliptical\_arc\_first\_node - optional; type *int*
  - elliptical\_arc\_second\_node - optional; type *int*
  - elliptical\_arc\_alpha - optional; type *double*
  - elliptical\_arc\_beta - optional; type *double*
  - elliptical\_arc\_normal - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - elliptical\_arc\_normal\_x - optional; type *double*
  - elliptical\_arc\_normal\_y - optional; type *double*
  - elliptical\_arc\_normal\_z - optional; type *double*
  - elliptical\_arc\_major\_radius - optional; type *double*
  - elliptical\_arc\_minor\_radius - optional; type *double*
  - elliptical\_arc\_center - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - elliptical\_arc\_center\_x - optional; type *double*
  - elliptical\_arc\_center\_y - optional; type *double*
  - elliptical\_arc\_center\_z - optional; type *double*

- elliptical\_arc\_focus\_1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_focus\_1\_x - optional; type *double*
- elliptical\_arc\_focus\_1\_y - optional; type *double*
- elliptical\_arc\_focus\_1\_z - optional; type *double*
- elliptical\_arc\_focus\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_focus\_2\_x - optional; type *double*
- elliptical\_arc\_focus\_2\_y - optional; type *double*
- elliptical\_arc\_focus\_2\_z - optional; type *double*
- elliptical\_arc\_first\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_first\_control\_point\_x - optional; type *double*
- elliptical\_arc\_first\_control\_point\_y - optional; type *double*
- elliptical\_arc\_first\_control\_point\_z - optional; type *double*
- elliptical\_arc\_second\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_second\_control\_point\_x - optional; type *double*
- elliptical\_arc\_second\_control\_point\_y - optional; type *double*
- elliptical\_arc\_second\_control\_point\_z - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- elliptical\_arc\_perimeter\_control\_point\_x - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point\_y - optional; type *double*
- elliptical\_arc\_perimeter\_control\_point\_z - optional; type *double*
- parabola\_first\_node - optional; type *int*
- parabola\_second\_node - optional; type *int*
- parabola\_control\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- parabola\_control\_point\_x - optional; type *double*
- parabola\_control\_point\_y - optional; type *double*
- parabola\_control\_point\_z - optional; type *double*
- parabola\_alpha - optional; type *double*
- parabola\_focus\_directrix\_distance - optional; type *double*
- parabola\_focus - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- parabola\_focus\_x - optional; type *double*
- parabola\_focus\_y - optional; type *double*
- parabola\_focus\_z - optional; type *double*
- nurbs\_order - optional; type *int*
- nurbs\_control\_points\_by\_components - optional; type *array\_of\_line\_nurbs\_control\_points\_by\_components*
  - line\_nurbs\_control\_points\_by\_components - optional, unbounded; type *line\_nurbs\_control\_points\_by\_components*
    - no - optional; type *int*
    - global\_coordinate\_x - optional; type *double*
    - global\_coordinate\_y - optional; type *double*
    - global\_coordinate\_z - optional; type *double*
    - weight - optional; type *double*
- nurbs\_control\_points - optional; type *array\_of\_line\_nurbs\_control\_points*
  - line\_nurbs\_control\_points - optional, unbounded; type *line\_nurbs\_control\_points*
    - no - optional; type *int*
    - global\_coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - weight type *double*
- nurbs\_knots - optional; type *array\_of\_line\_nurbs\_knots*
  - line\_nurbs\_knots - optional, unbounded; type *double*
- rotation\_specification\_type - optional; type *line\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE' }
- rotation\_angle - optional; type *double*
- rotation\_help\_node - optional; type *int*
- rotation\_plane - optional; type *line\_rotation\_plane* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- is\_rotated - optional; type *boolean*
- member - optional; type *int*
- support - optional; type *int*
- mesh\_refinement - optional; type *int*
- line\_weld\_assignment - optional; type *array\_of\_line\_line\_weld\_assignment*
  - line\_line\_weld\_assignment - optional, unbounded; type *line\_line\_weld\_assignment*
    - no - optional; type *int*
    - weld - optional; type *int*
    - surface1 - optional; type *int*
    - surface2 - optional; type *int*
    - surface3 - optional; type *int*
- has\_line\_welds - optional; type *boolean*
- is\_generated - optional; type *boolean*

- `generating_object_info` - optional; type *string*
- `id_for_export_import` - optional; type *string*
- `metadata_for_export_import` - optional; type *string*

**Output:** `set_line_response` (soap:body, use = literal) [Source code](#)

parameters type `set_lineResponse`

## 160. `set_line_grid`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_line_grid`

**Input:** `set_line_grid_request` (soap:body, use = literal) [Source code](#)

parameters type `set_line_grid`

- value type `line_grid`
  - no type `int`
  - type - optional; type `line_grid_type` - type *undefined* with restriction - enum { 'TYPE\_STANDARD' }
  - `user_defined_name_enabled` - optional; type `boolean`
  - name - optional; type `string`
  - `coordinate_system_type` - optional; type `line_grid_coordinate_system_type` - type *undefined* with restriction - enum { 'CARTESIAN', 'CYLINDRICAL', 'INCLINED', 'SPHERICAL' }
  - `lock_line_grid` - optional; type `boolean`
  - `include_in_view` - optional; type `boolean`
  - `line_grid_points` - optional; type `boolean`
  - `line_grid_lines` - optional; type `boolean`
  - `label_text` - optional; type `boolean`
  - `dimensions` - optional; type `boolean`
  - `lock_points_to_line_grid` - optional; type `boolean`
  - `origin_type` - optional; type `line_grid_origin_type` - type *undefined* with restriction - enum { 'ORIGIN\_FREE\_POINT', 'ORIGIN\_NODE' }
  - `origin_node` - optional; type `int`
  - `origin_free_point_coordinates` - optional; type `vector_3d`
    - x type `double`
    - y type `double`
    - z type `double`
  - `origin_free_point_coordinate_x` - optional; type `double`
  - `origin_free_point_coordinate_y` - optional; type `double`
  - `origin_free_point_coordinate_z` - optional; type `double`
  - `alpha_ux` - optional; type `double`
  - `alpha_vy` - optional; type `double`
  - `alpha_wz` - optional; type `double`
  - `x_input_type` - optional; type `line_grid_x_input_type` - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
  - `x_direction` - optional; type `line_grid_x_direction` - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
  - `x_assignment` - optional; type `array_of_line_grid_x_assignment`
    - `line_grid_x_assignment` - optional, unbounded; type `line_grid_x_assignment`
      - no - optional; type `int`
      - name - optional; type `string`
      - `coordinates` - optional; type `double`
      - `spans` - optional; type `double`
      - `spans_count` - optional; type `int`
  - `y_input_type` - optional; type `line_grid_y_input_type` - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
  - `y_direction` - optional; type `line_grid_y_direction` - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
  - `y_assignment` - optional; type `array_of_line_grid_y_assignment`
    - `line_grid_y_assignment` - optional, unbounded; type `line_grid_y_assignment`
      - no - optional; type `int`
      - name - optional; type `string`
      - `coordinates` - optional; type `double`
      - `spans` - optional; type `double`
      - `spans_count` - optional; type `int`
  - `z_input_type` - optional; type `line_grid_z_input_type` - type *undefined* with restriction - enum { 'INPUT\_TYPE\_COORDINATES', 'INPUT\_TYPE\_SPANS' }
  - `z_direction` - optional; type `line_grid_z_direction` - type *undefined* with restriction - enum { 'DIRECTION\_NEGATIVE', 'DIRECTION\_POSITIVE' }
  - `z_assignment` - optional; type `array_of_line_grid_z_assignment`
    - `line_grid_z_assignment` - optional, unbounded; type `line_grid_z_assignment`
      - no - optional; type `int`
      - name - optional; type `string`
      - `coordinates` - optional; type `double`
      - `spans` - optional; type `double`
      - `spans_count` - optional; type `int`
  - `coordinate_system` - optional; type `int`
  - `rotation_coordinate_system` - optional; type `int`
  - `has_specific_direction` - optional; type `boolean`
  - `specific_direction_type` - optional; type `line_grid_specific_direction_type` - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
  - `axes_sequence` - optional; type `line_grid_axes_sequence` - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - `rotated_about_angle_x` - optional; type `double`
  - `rotated_about_angle_y` - optional; type `double`
  - `rotated_about_angle_z` - optional; type `double`
  - `rotated_about_angle_1` - optional; type `double`
  - `rotated_about_angle_2` - optional; type `double`
  - `rotated_about_angle_3` - optional; type `double`
  - `directed_to_node_direction_node` - optional; type `int`
  - `directed_to_node_plane_node` - optional; type `int`
  - `directed_to_node_first_axis` - optional; type `line_grid_directed_to_node_first_axis` - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }

- directed\_to\_node\_second\_axis - optional; type *line\_grid\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *line\_grid\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_axis - optional; type *line\_grid\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_line\_grid\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_gridResponse*

## 161. set\_line\_hinge

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_line\_hinge

**Input:** set\_line\_hinge\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_hinge*

- value type *line\_hinge*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to - optional; type *string*
  - translational\_release\_u\_x - optional; type *double*
  - translational\_release\_u\_x\_nonlinearity - optional; type *line\_hinge\_translational\_release\_u\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - translational\_release\_u\_y - optional; type *double*
  - translational\_release\_u\_y\_nonlinearity - optional; type *line\_hinge\_translational\_release\_u\_y\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - translational\_release\_u\_z - optional; type *double*
  - translational\_release\_u\_z\_nonlinearity - optional; type *line\_hinge\_translational\_release\_u\_z\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - rotational\_release\_phi\_x - optional; type *double*
  - rotational\_release\_phi\_x\_nonlinearity - optional; type *line\_hinge\_rotational\_release\_phi\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
  - force\_moment\_diagram\_around\_x\_table - optional; type *array\_of\_line\_hinge\_force\_moment\_diagram\_around\_x\_table*
    - line\_hinge\_force\_moment\_diagram\_around\_x\_table - optional, unbounded; type *line\_hinge\_force\_moment\_diagram\_around\_x\_table*
      - no - optional; type *int*
      - force - optional; type *double*
      - max\_moment - optional; type *double*
      - min\_moment - optional; type *double*
      - note - optional; type *string*
  - force\_moment\_diagram\_around\_x\_symmetric - optional; type *boolean*
  - force\_moment\_diagram\_around\_x\_is\_sorted - optional; type *boolean*
  - force\_moment\_diagram\_around\_x\_start - optional; type *line\_hinge\_force\_moment\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - force\_moment\_diagram\_around\_x\_end - optional; type *line\_hinge\_force\_moment\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'FORCE\_MOMENT\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - force\_moment\_diagram\_around\_x\_depends\_on - optional; type *line\_hinge\_force\_moment\_diagram\_around\_x\_depends\_on* - type *undefined* with restriction - enum { 'FORCE\_MOMENT\_DIAGRAM\_DEPENDS\_ON\_N', 'FORCE\_MOMENT\_DIAGRAM\_DEPENDS\_ON\_VY', 'FORCE\_MOMENT\_DIAGRAM\_DEPENDS\_ON\_VZ' }
  - slab\_wall\_connection - optional; type *boolean*
  - slab\_wall\_with\_slab\_edge\_block - optional; type *boolean*
  - slab\_wall\_connection\_offset - optional; type *double*
  - slab\_edge\_block\_width - optional; type *double*
  - generated\_line\_hinges - optional; type *array\_of\_line\_hinge\_generated\_line\_hinges*
    - line\_hinge\_generated\_line\_hinges - optional, unbounded; type *line\_hinge\_generated\_line\_hinges*



- no - optional; type *int*
- generated\_by - optional; type *int*
- generated\_line\_hinge - optional; type *int*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_line\_hinge\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_hingeResponse*

## 162. set\_line\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_line\_load

**Input:** set\_line\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_load*

- load\_case\_no type *int*
- value type *line\_load*
  - no type *int*
  - load\_type - optional; type *line\_load\_load\_type* - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MOMENT' }
  - lines - optional; type *array\_of\_int*
  - load\_case - optional; type *int*
  - coordinate\_system - optional; type *string*
  - load\_distribution - optional; type *line\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING' }
  - load\_direction - optional; type *line\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
  - load\_direction\_orientation - optional; type *line\_load\_load\_direction\_orientation* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
  - magnitude - optional; type *double*
  - magnitude\_1 - optional; type *double*
  - magnitude\_2 - optional; type *double*
  - magnitude\_3 - optional; type *double*
  - individual\_mass\_components - optional; type *boolean*
  - mass\_global - optional; type *double*
  - mass\_x - optional; type *double*
  - mass\_y - optional; type *double*
  - mass\_z - optional; type *double*
  - distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_a\_absolute - optional; type *double*
  - distance\_a\_relative - optional; type *double*
  - distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_b\_absolute - optional; type *double*
  - distance\_b\_relative - optional; type *double*
  - distance\_c\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_c\_absolute - optional; type *double*
  - distance\_c\_relative - optional; type *double*
  - count\_n - optional; type *int*
  - varying\_load\_parameters\_are\_defined\_as\_relative - optional; type *boolean*
  - varying\_load\_parameters - optional; type *array\_of\_line\_load\_varying\_load\_parameters*
    - line\_load\_varying\_load\_parameters - optional, unbounded; type *line\_load\_varying\_load\_parameters*
      - no - optional; type *int*
      - distance - optional; type *double*
      - delta\_distance - optional; type *double*
      - magnitude - optional; type *double*
      - note - optional; type *string*
    - varying\_load\_parameters\_sorted - optional; type *boolean*
    - reference\_to\_list\_of\_lines - optional; type *boolean*
    - distance\_from\_line\_end - optional; type *boolean*
    - load\_is\_over\_total\_length - optional; type *boolean*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_line\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_loadResponse*

## 163. set\_line\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_line\_mesh\_refinement

**Input:** set\_line\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_mesh\_refinement*

- value type *line\_mesh\_refinement*

- no *type int*
- *type* - optional; *type line\_mesh\_refinement\_type* - *type undefined* with restriction - enum { 'TYPE\_ELEMENTS', 'TYPE\_GRADUAL', 'TYPE\_LENGTH' }
- *comment* - optional; *type string*
- *elements\_finite\_elements* - optional; *type int*
- *generating\_object\_info* - optional; *type string*
- *gradual\_rows* - optional; *type int*
- *is\_generated* - optional; *type boolean*
- *lines* - optional; *type array\_of\_int*
- *name* - optional; *type string*
- *number\_of\_layers* - optional; *type int*
- *target\_length* - optional; *type double*
- *user\_defined\_name\_enabled* - optional; *type boolean*
- *id\_for\_export\_import* - optional; *type string*
- *metadata\_for\_export\_import* - optional; *type string*

**Output:** `set_line_mesh_refinement_response` (soap:body, use = literal) [Source code](#)

parameters *type set\_line\_mesh\_refinementResponse*

#### 164. set\_line\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_line_set`

**Input:** `set_line_set_request` (soap:body, use = literal) [Source code](#)

parameters *type set\_line\_set*

- value *type line\_set*
  - no *type int*
  - *user\_defined\_name\_enabled* - optional; *type boolean*
  - *name* - optional; *type string*
  - *set\_type* - optional; *type line\_set\_set\_type* - *type undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
  - *lines* - optional; *type array\_of\_int*
  - *length* - optional; *type double*
  - *center\_of\_gravity* - optional; *type vector\_3d*
    - *x* *type double*
    - *y* *type double*
    - *z* *type double*
  - *center\_of\_gravity\_x* - optional; *type double*
  - *center\_of\_gravity\_y* - optional; *type double*
  - *center\_of\_gravity\_z* - optional; *type double*
  - *position* - optional; *type string*
  - *position\_short* - optional; *type string*
  - *comment* - optional; *type string*
  - *is\_generated* - optional; *type boolean*
  - *generating\_object\_info* - optional; *type string*
  - *id\_for\_export\_import* - optional; *type string*
  - *metadata\_for\_export\_import* - optional; *type string*

**Output:** `set_line_set_response` (soap:body, use = literal) [Source code](#)

parameters *type set\_line\_setResponse*

#### 165. set\_line\_set\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_line_set_load`

**Input:** `set_line_set_load_request` (soap:body, use = literal) [Source code](#)

parameters *type set\_line\_set\_load*

- *load\_case\_no* *type int*
- value *type line\_set\_load*
  - no *type int*
  - *load\_type* - optional; *type line\_set\_load\_load\_type* - *type undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MOMENT' }
  - *line\_sets* - optional; *type array\_of\_int*
  - *load\_case* - optional; *type int*
  - *coordinate\_system* - optional; *type string*
  - *load\_distribution* - optional; *type line\_set\_load\_load\_distribution* - *type undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING' }
  - *load\_direction* - optional; *type line\_set\_load\_load\_direction* - *type undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
  - *load\_direction\_orientation* - optional; *type line\_set\_load\_load\_direction\_orientation* - *type undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
  - *magnitude* - optional; *type double*
  - *magnitude\_1* - optional; *type double*
  - *magnitude\_2* - optional; *type double*
  - *magnitude\_3* - optional; *type double*
  - *mass\_global* - optional; *type double*
  - *mass\_x* - optional; *type double*
  - *mass\_y* - optional; *type double*
  - *mass\_z* - optional; *type double*

- count\_n - optional; type *int*
- varying\_load\_parameters\_are\_defined\_as\_relative - optional; type *boolean*
- varying\_load\_parameters - optional; type *array\_of\_line\_set\_load\_varying\_load\_parameters*
  - line\_set\_load\_varying\_load\_parameters - optional, unbounded; type *line\_set\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_a\_absolute - optional; type *double*
- distance\_a\_relative - optional; type *double*
- distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_b\_absolute - optional; type *double*
- distance\_b\_relative - optional; type *double*
- distance\_c\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_c\_absolute - optional; type *double*
- distance\_c\_relative - optional; type *double*
- reference\_to\_list\_of\_line\_sets - optional; type *boolean*
- distance\_from\_line\_set\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- individual\_mass\_components - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_line\_set\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_set\_loadResponse*

## 166. set\_line\_support

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_line\_support

**Input:** set\_line\_support\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_support*

- value type *line\_support*
  - no type *int*
  - comment - optional; type *string*
  - coordinate\_system - optional; type *line\_support\_coordinate\_system* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_GLOBAL', 'COORDINATE\_SYSTEM\_LOCAL' }
  - diagram\_along\_x\_end - optional; type *line\_support\_diagram\_along\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_x\_is\_sorted - optional; type *boolean*
  - diagram\_along\_x\_start - optional; type *line\_support\_diagram\_along\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_x\_symmetric - optional; type *boolean*
  - diagram\_along\_x\_table - optional; type *array\_of\_line\_support\_diagram\_along\_x\_table*
    - line\_support\_diagram\_along\_x\_table - optional, unbounded; type *line\_support\_diagram\_along\_x\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_along\_y\_end - optional; type *line\_support\_diagram\_along\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_y\_is\_sorted - optional; type *boolean*
  - diagram\_along\_y\_start - optional; type *line\_support\_diagram\_along\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_y\_symmetric - optional; type *boolean*
  - diagram\_along\_y\_table - optional; type *array\_of\_line\_support\_diagram\_along\_y\_table*
    - line\_support\_diagram\_along\_y\_table - optional, unbounded; type *line\_support\_diagram\_along\_y\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_along\_z\_end - optional; type *line\_support\_diagram\_along\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_z\_is\_sorted - optional; type *boolean*
  - diagram\_along\_z\_start - optional; type *line\_support\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_along\_z\_symmetric - optional; type *boolean*
  - diagram\_along\_z\_table - optional; type *array\_of\_line\_support\_diagram\_along\_z\_table*
    - line\_support\_diagram\_along\_z\_table - optional, unbounded; type *line\_support\_diagram\_along\_z\_table*
      - no - optional; type *int*
      - displacement - optional; type *double*
      - force - optional; type *double*
      - spring - optional; type *double*
      - note - optional; type *string*
  - diagram\_around\_x\_end - optional; type *line\_support\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
  - diagram\_around\_x\_is\_sorted - optional; type *boolean*





- generating\_object\_info - optional; type *string*
- fictitious\_wall\_enabled - optional; type *boolean*
- fictitious\_wall\_width - optional; type *double*
- fictitious\_wall\_height - optional; type *double*
- fictitious\_wall\_head\_support\_type - optional; type *line\_support\_fictitious\_wall\_head\_support\_type* - type *undefined* with restriction - enum { 'HEAD\_SUPPORT\_TYPE\_HINGED', 'HEAD\_SUPPORT\_TYPE\_SEMI\_RIGID' }
- fictitious\_wall\_base\_support\_type - optional; type *line\_support\_fictitious\_wall\_base\_support\_type* - type *undefined* with restriction - enum { 'BASE\_SUPPORT\_TYPE\_ELASTIC', 'BASE\_SUPPORT\_TYPE\_HINGED', 'BASE\_SUPPORT\_TYPE\_RIGID' }
- fictitious\_wall\_base\_elastic - optional; type *double*
- fictitious\_wall\_shear\_stiffness - optional; type *boolean*
- fictitious\_wall\_material - optional; type *int*
- fictitious\_wall\_spring\_x - optional; type *double*
- fictitious\_wall\_spring\_y - optional; type *double*
- fictitious\_wall\_spring\_z - optional; type *double*
- fictitious\_wall\_rotational\_restraint\_about\_line\_axis - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_line\_support\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_supportResponse*

## 167. set\_line\_welded\_joint

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_line\_welded\_joint

Input: set\_line\_welded\_joint\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_welded\_joint*

- value type *line\_welded\_joint*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - joint\_type - optional; type *line\_welded\_joint\_joint\_type* - type *undefined* with restriction - enum { 'BUTT\_JOINT', 'CORNER\_JOINT', 'LAP\_JOINT', 'TEE\_JOINT' }
  - weld\_type - optional; type *line\_welded\_joint\_weld\_type* - type *undefined* with restriction - enum { 'WELD\_BEVEL\_AND\_FILLET', 'WELD\_DOUBLE\_BEVEL', 'WELD\_DOUBLE\_FILLET', 'WELD\_DOUBLE\_J', 'WELD\_DOUBLE\_SQUARE', 'WELD\_DOUBLE\_U', 'WELD\_DOUBLE\_V', 'WELD\_J\_AND\_FILLET', 'WELD\_SINGLE\_BEVEL', 'WELD\_SINGLE\_FILLET', 'WELD\_SINGLE\_J', 'WELD\_SINGLE\_SQUARE', 'WELD\_SINGLE\_U', 'WELD\_SINGLE\_V', 'WELD\_V\_AND\_FILLET' }
  - longitudinal\_arrangement - optional; type *line\_welded\_joint\_longitudinal\_arrangement* - type *undefined* with restriction - enum { 'CHAIN\_INTERMITTENT', 'CONTINUOUS', 'STAGGERED\_INTERMITTENT' }
  - weld\_size\_a1 - optional; type *double*
  - weld\_size\_a2 - optional; type *double*
  - weld\_length - optional; type *double*
  - pitch - optional; type *double*
  - first\_weld\_position - optional; type *double*
  - stress\_analysis\_configuration - optional; type *int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

Output: set\_line\_welded\_joint\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_line\_welded\_jointResponse*

## 168. set\_load\_case

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_load\_case

Input: set\_load\_case\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_load\_case*

- value type *load\_case*
  - no type *int*
  - analysis\_type - optional; type *load\_case\_analysis\_type* - type *undefined* with restriction - enum { 'ANALYSIS\_TYPE\_CREEP\_AND\_SHRINKAGE', 'ANALYSIS\_TYPE\_CUTTING\_PATTERN', 'ANALYSIS\_TYPE\_MODAL', 'ANALYSIS\_TYPE\_RESPONSE\_SPECTRUM', 'ANALYSIS\_TYPE\_STATIC', 'ANALYSIS\_TYPE\_TIME\_DEPENDENT', 'ANALYSIS\_TYPE\_TIME\_HISTORY', 'ANALYSIS\_TYPE\_WIND\_SIMULATION' }
  - name - optional; type *string*
  - static\_analysis\_settings - optional; type *int*
  - stability\_analysis\_settings - optional; type *int*
  - modal\_analysis\_settings - optional; type *int*
  - spectral\_analysis\_settings - optional; type *int*
  - calculate\_critical\_load - optional; type *boolean*
  - consider\_imperfection - optional; type *boolean*
  - imperfection\_case - optional; type *int*
  - consider\_initial\_state - optional; type *boolean*
  - initial\_state\_case - optional; type *int*
  - initial\_state\_definition\_type - optional; type *load\_case\_initial\_state\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_FINAL\_STATE', 'DEFINITION\_TYPE\_STIFFNESS', 'DEFINITION\_TYPE\_STRAINS', 'DEFINITION\_TYPE\_STRAINS\_WITH\_USER\_DEFINED\_FACTORS' }
  - creep\_loading\_case - optional; type *int*
  - individual\_factors\_of\_selected\_objects\_table - optional; type *array\_of\_load\_case\_individual\_factors\_of\_selected\_objects\_table*
    - load\_case\_individual\_factors\_of\_selected\_objects\_table - optional, unbounded; type *load\_case\_individual\_factors\_of\_selected\_objects\_table*
      - no - optional; type *int*
      - object\_type - optional; type *object\_type* - type *undefined* with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE',

- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID',
- 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
- object\_list - optional; type *array\_of\_int*
- strain\_type - optional; type *strain\_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
- factor - optional; type *double*
- comment - optional; type *string*
- consider\_construction\_stage - optional; type *boolean*
- construction\_stage\_case - optional; type *int*
- to\_solve - optional; type *boolean*
- action\_category - optional; type *string*
- self\_weight\_active - optional; type *boolean*
- self\_weight\_factors - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- self\_weight\_factor\_x - optional; type *double*
- self\_weight\_factor\_y - optional; type *double*
- self\_weight\_factor\_z - optional; type *double*
- factor\_phi - optional; type *load\_case\_factor\_phi* - type *undefined* with restriction - enum { 'FACTOR\_PHI\_1', 'FACTOR\_PHI\_2' }
- load\_duration - optional; type *int*
- loading\_start - optional; type *double*
- time\_being\_investigated - optional; type *double*
- has\_inclusive\_load\_cases - optional; type *boolean*
- inclusive\_load\_cases - optional; type *array\_of\_load\_case\_inclusive\_load\_cases*
  - load\_case\_inclusive\_load\_cases - optional, unbounded; type *int*
- has\_exclusive\_load\_cases - optional; type *boolean*
- exclusive\_load\_cases - optional; type *array\_of\_load\_case\_exclusive\_load\_cases*
  - load\_case\_exclusive\_load\_cases - optional, unbounded; type *int*
- import\_masses\_from - optional; type *int*
- import\_modal\_analysis\_from - optional; type *int*
- response\_spectrum\_is\_enabled\_in\_any\_direction - optional; type *boolean*
- response\_spectrum\_is\_enabled\_in\_direction\_x - optional; type *boolean*
- response\_spectrum\_is\_enabled\_in\_direction\_y - optional; type *boolean*
- response\_spectrum\_is\_enabled\_in\_direction\_z - optional; type *boolean*
- response\_spectrum\_in\_direction\_x - optional; type *int*
- response\_spectrum\_in\_direction\_y - optional; type *int*
- response\_spectrum\_in\_direction\_z - optional; type *int*
- response\_spectrum\_scale\_factor\_in\_direction\_x - optional; type *double*
- response\_spectrum\_scale\_factor\_in\_direction\_y - optional; type *double*
- response\_spectrum\_scale\_factor\_in\_direction\_z - optional; type *double*
- response\_spectrum\_rotation\_angle - optional; type *double*
- response\_spectrum\_consider\_accidental\_torsion - optional; type *boolean*
- response\_spectrum\_eccentricity\_for\_x\_direction\_absolute - optional; type *double*
- response\_spectrum\_eccentricity\_for\_y\_direction\_absolute - optional; type *double*
- response\_spectrum\_eccentricity\_for\_x\_direction\_relative - optional; type *double*
- response\_spectrum\_eccentricity\_for\_y\_direction\_relative - optional; type *double*
- response\_spectrum\_building\_length\_in\_x - optional; type *double*
- response\_spectrum\_building\_length\_in\_y - optional; type *double*
- response\_spectrum\_eccentricity\_for\_x\_direction\_is\_defined\_as\_relative - optional; type *boolean*
- response\_spectrum\_eccentricity\_for\_y\_direction\_is\_defined\_as\_relative - optional; type *boolean*
- response\_spectrum\_user\_defined\_building\_lengths - optional; type *boolean*
- selection\_of\_modes\_mode\_activation\_table - optional; type *array\_of\_load\_case\_selection\_of\_modes\_mode\_activation\_table*
  - load\_case\_selection\_of\_modes\_mode\_activation\_table - optional, unbounded; type *load\_case\_selection\_of\_modes\_mode\_activation\_table*
    - no - optional; type *int*
    - to\_generate - optional; type *boolean*
    - natural\_period\_t - optional; type *double*
    - natural\_frequency\_f - optional; type *double*
    - acceleration\_s\_ax - optional; type *double*
    - acceleration\_s\_ay - optional; type *double*
    - acceleration\_s\_az - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mex - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mey - optional; type *double*
    - effective\_modal\_mass\_factor\_f\_mez - optional; type *double*
    - damping - optional; type *double*
- selection\_of\_modes\_deselect\_modes\_according\_to\_criterion\_is\_enabled - optional; type *boolean*
- selection\_of\_modes\_deselect\_modes\_modes\_according\_to\_criterion\_value - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mex - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mey - optional; type *double*
- selection\_of\_modes\_total\_effective\_modal\_mass\_factor\_f\_mez - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- structure\_modification\_enabled - optional; type *boolean*
- structure\_modification - optional; type *int*
- possibility\_of\_crowds - optional; type *boolean*
- specification\_for\_load\_case\_gr1a - optional; type *load\_case\_specification\_for\_load\_case\_gr1a* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR1A\_PEDESTRIAN\_AND\_CYCLE\_TRACK', 'SPECIFICATION\_GR1A\_TS1', 'SPECIFICATION\_GR1A\_UDL' }
- specification\_for\_load\_case\_gr2 - optional; type *load\_case\_specification\_for\_load\_case\_gr2* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR2\_HORIZONTAL\_FORCES\_BRAKING\_AND\_ACCELERATION', 'SPECIFICATION\_GR2\_HORIZONTAL\_FORCES\_CENTRIFUGAL\_FORCES' }
- specification\_for\_load\_case\_gr5 - optional; type *load\_case\_specification\_for\_load\_case\_gr5* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR5\_SPECIAL\_VEHICLE', 'SPECIFICATION\_GR5\_TS', 'SPECIFICATION\_GR5\_UDL' }
- specification\_for\_load\_case\_gr6 - optional; type *load\_case\_specification\_for\_load\_case\_gr6* - type *undefined* with restriction - enum { 'SPECIFICATION\_GR6\_HORIZONTAL\_FORCES\_BRAKING\_AND\_ACCELERATION', 'SPECIFICATION\_GR6\_HORIZONTAL\_FORCES\_CENTRIFUGAL\_FORCES', 'SPECIFICATION\_GR6\_PEDESTRIAN\_AND\_CYCLE\_TRACK', 'SPECIFICATION\_GR6\_TS', 'SPECIFICATION\_GR6\_UDL' }

- liveload\_less\_equal\_100 - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- liveload\_consider\_design\_situation\_seismic\_weight\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_seismic\_weight\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_seismic\_weight\_combo* - type *undefined* with restriction - enum { 'ESEISMIC\_WEIGHT\_FLOOR\_LOADS\_ABOVE\_3', 'ESEISMIC\_WEIGHT\_FLOOR\_LOADS\_UP\_TO\_3' }
- liveload\_consider\_design\_situation\_seismic\_mass\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_seismic\_mass\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_seismic\_mass\_combo* - type *undefined* with restriction - enum { 'ESEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_ACTUAL\_CONDITIONS', 'ESEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_EQUIVALENT\_UNIFORM\_LIBRARY\_ARCHIVES', 'ESEISMIC\_MASS\_LOAD\_CALCULATED\_ACCORDING\_EQUIVALENT\_UNIFORM\_OTHER\_CIVIL\_BUILDINGS' }
- consider\_design\_situation\_effective\_seismic\_weight - optional; type *boolean*
- liveload\_less\_equal\_48 - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_aci\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- liveload\_less\_equal\_100\_ibc - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- cranehookload\_nocombined\_roofliveload - optional; type *boolean*
- roof\_configurations\_that\_do\_not\_shed\_snow - optional; type *boolean*
- flat\_roof\_snow\_load - optional; type *boolean*
- flat\_roof\_snow\_load\_category - optional; type *load\_case\_flat\_roof\_snow\_load\_category* - type *undefined* with restriction - enum { 'ROOF\_CAT\_SNOW\_LOADS\_EXCEEDING\_30PSF', 'ROOF\_CAT\_SNOW\_LOADS\_OF\_30PSF\_OR\_LESS' }
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_check - optional; type *boolean*
- liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_combo - optional; type *load\_case\_liveload\_consider\_design\_situation\_effective\_seismic\_weight\_ibc\_s\_combo* - type *undefined* with restriction - enum { 'EFFECTIVE\_SEISMIC\_WEIGHT\_FLOOR\_LIVE\_LOAD', 'EFFECTIVE\_SEISMIC\_WEIGHT\_OPERATING\_WEIGHT\_OF\_PERMANENT\_EQUIPMENT', 'EFFECTIVE\_SEISMIC\_WEIGHT\_PARTITIONS' }
- deadload\_factor\_shall\_be\_increased - optional; type *boolean*
- deadload\_factor\_shall\_be\_increased1 - optional; type *boolean*
- deadload\_factor\_shall\_be\_increased2 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight\_NBC05 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight\_NBC15 - optional; type *boolean*
- liveload\_principal\_factor\_may\_be\_reduced1 - optional; type *boolean*
- liveload\_companion\_factor\_may\_be\_increased1 - optional; type *boolean*
- liveload\_principal\_factor\_may\_be\_reduced2 - optional; type *boolean*
- liveload\_companion\_factor\_may\_be\_increased2 - optional; type *boolean*
- consider\_design\_situation\_effective\_seismic\_weight\_NBC05 - optional; type *boolean*
- consider\_design\_situation\_effective\_seismic\_weight\_NBC15 - optional; type *boolean*
- dead\_no\_effective\_seismic\_weight - optional; type *boolean*
- longtermload\_tp\_considered\_in\_limit\_state\_of\_serviceability - optional; type *boolean*
- greater\_intensity\_than\_4 - optional; type *boolean*
- distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_check - optional; type *boolean*
- distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_combo - optional; type *load\_case\_distinguish\_different\_overstrength\_factor\_in\_direction\_xy\_combo* - type *undefined* with restriction - enum { 'OVERSTRENGTH\_FACTOR\_DIRECTION\_X', 'OVERSTRENGTH\_FACTOR\_DIRECTION\_Y' }
- consider\_design\_situation\_seismic\_mass\_combination - optional; type *boolean*
- wind\_simulation\_wind\_direction\_angle - optional; type *double*
- wind\_simulation\_terrain\_offset - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_minus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_plus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_minus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_plus\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_plus\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_depth\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_height\_coefficient - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_width\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_coefficient - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_coefficient - optional; type *double*
- wind\_simulation\_wind\_profile - optional; type *int*
- wind\_simulation\_analysis\_settings - optional; type *int*
- wind\_simulation\_wind\_tunnel\_depth\_minus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth\_plus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_minus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width\_plus\_length - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height\_plus\_length - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_depth - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_height - optional; type *double*
- wind\_simulation\_construction\_bounding\_box\_width - optional; type *double*
- wind\_simulation\_wind\_tunnel\_depth - optional; type *double*
- wind\_simulation\_wind\_tunnel\_width - optional; type *double*
- wind\_simulation\_wind\_tunnel\_height - optional; type *double*
- geotechnical\_analysis\_reset\_small\_strain\_state\_variables - optional; type *boolean*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_load\_case\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_load\_caseResponse*



## 169. set\_load\_cases\_and\_combinations

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_load\_cases\_and\_combinations

**Input:** set\_load\_cases\_and\_combinations\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_load_cases_and_combinations
 value type load_cases_and_combinations
 current_standard_for_combination_wizard - optional; type int
 activate_combination_wizard_and_classification - optional; type boolean
 activate_combination_wizard - optional; type boolean
 result_combinations_active - optional; type boolean
 result_combinations_parentheses_active - optional; type boolean
 result_combinations_consider_sub_results - optional; type boolean
 combination_name_according_to_action_category - optional; type boolean
```

**Output:** set\_load\_cases\_and\_combinations\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_load_cases_and_combinationsResponse
```

## 170. set\_load\_combination

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_load\_combination

**Input:** set\_load\_combination\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_load_combination
 value type load_combination
 no type int
 analysis_type - optional; type load_combination_analysis_type - type undefined with restriction - enum {
 'ANALYSIS_TYPE_HARMONIC_RESPONSE_ANALYSIS', 'ANALYSIS_TYPE_STATIC',
 'ANALYSIS_TYPE_STATIC_CREEP_AND_SHRINKAGE', 'ANALYSIS_TYPE_STATIC_TIME_DEPENDENCE' }
 design_situation - optional; type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 static_analysis_settings - optional; type int
 import_modal_analysis_load_case - optional; type int
 calculate_critical_load - optional; type boolean
 stability_analysis_settings - optional; type int
 consider_imperfection - optional; type boolean
 imperfection_case - optional; type int
 consider_initial_state - optional; type boolean
 initial_state_case - optional; type int
 consider_construction_stage - optional; type boolean
 construction_stage - optional; type int
 consider_creep_caused_by_permanent_loading - optional; type boolean
 creep_caused_by_permanent_loading_case - optional; type int
 sustained_load_enabled - optional; type boolean
 sustained_load - optional; type int
 sway_load_enabled - optional; type boolean
 sway_load - optional; type int
 structure_modification_enabled - optional; type boolean
 structure_modification - optional; type int
 to_solve - optional; type boolean
 comment - optional; type string
 load_duration - optional; type int
 items - optional; type array_of_load_combination_items
 load_combination_items - optional, unbounded; type load_combination_items
 no - optional; type int
 factor - optional; type double
 load_case - optional; type int
 action - optional; type int
 is_leading - optional; type boolean
 gamma - optional; type double
 psi - optional; type double
 xi - optional; type double
 k_fi - optional; type double
 c_esl - optional; type double
 k_def - optional; type double
 psi_0 - optional; type double
 psi_1 - optional; type double
 psi_2 - optional; type double
 fi - optional; type double
 gamma_0 - optional; type double
 alfa - optional; type double
 k_f - optional; type double
 phi - optional; type double
 rho - optional; type double
 omega_0 - optional; type double
 gamma_l_1 - optional; type double
 k_creep - optional; type double
 shift - optional; type double
 amplitude_function_type - optional; type amplitude_function_type - type undefined with restriction -
 enum { 'CONSTANT', 'LINEAR', 'QUADRATIC' }
 loading_start - optional; type double
 time_being_investigated - optional; type double
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 initial_state_definition_type - optional; type load_combination_initial_state_definition_type - type undefined with
 restriction - enum { 'DEFINITION_TYPE_FINAL_STATE', 'DEFINITION_TYPE_STIFFNESS',
 'DEFINITION_TYPE_STRAINS', 'DEFINITION_TYPE_STRAINS_WITH_USER_DEFINED_FACTORS' }
 individual_factors_of_selected_objects_table - optional; type
 array_of_load_combination_individual_factors_of_selected_objects_table
```

- **load\_combination\_individual\_factors\_of\_selected\_objects\_table** - optional, unbounded; type **load\_combination\_individual\_factors\_of\_selected\_objects\_table**
  - **no** - optional; type **int**
  - **object\_type** - optional; type **object\_type** - type **undefined** with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
  - **object\_list** - optional; type **array\_of\_int**
  - **strain\_type** - optional; type **strain\_type** - type **undefined** with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
  - **factor** - optional; type **double**
  - **comment** - optional; type **string**
- **geotechnical\_analysis\_reset\_small\_strain\_state\_variables** - optional; type **boolean**
- **id\_for\_export\_import** - optional; type **string**
- **metadata\_for\_export\_import** - optional; type **string**

**Output:** set\_load\_combination\_response (soap:body, use = literal) [Source code](#)

**parameters** type **set\_load\_combinationResponse**

## 171. set\_material

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_material

**Input:** set\_material\_request (soap:body, use = literal) [Source code](#)

**parameters** type **set\_material**

- value type **material**
  - **no** type **int**
  - **material\_type** - optional; type **material\_material\_type** - type **undefined** with restriction - enum { 'TYPE\_ALUMINUM', 'TYPE\_BASIC', 'TYPE\_CONCRETE', 'TYPE\_FABRIC', 'TYPE\_FOIL', 'TYPE\_GAS', 'TYPE\_GLASS', 'TYPE\_MASONRY', 'TYPE\_METAL', 'TYPE\_REINFORCING\_STEEL', 'TYPE\_SOIL', 'TYPE\_STEEL', 'TYPE\_TIMBER' }
  - **material\_model** - optional; type **material\_material\_model** - type **undefined** with restriction - enum { 'MODEL\_ISOTROPIC\_DAMAGE\_2D\_3D', 'MODEL\_ISOTROPIC\_LINEAR\_ELASTIC', 'MODEL\_ISOTROPIC\_MASONRY\_PLASTIC\_2D', 'MODEL\_ISOTROPIC\_NONLINEAR\_ELASTIC\_1D', 'MODEL\_ISOTROPIC\_NONLINEAR\_ELASTIC\_2D\_3D', 'MODEL\_ISOTROPIC\_PLASTIC\_1D', 'MODEL\_ISOTROPIC\_PLASTIC\_2D\_3D', 'MODEL\_ISOTROPIC\_SOIL\_NONLINEAR\_ELASTIC\_3D', 'MODEL\_ISOTROPIC\_SOIL\_PLASTIC\_3D', 'MODEL\_ISOTROPIC\_TIMBER\_LINEAR\_ELASTIC\_MEMBERS', 'MODEL\_ORTHOTROPIC\_2D', 'MODEL\_ORTHOTROPIC\_3D', 'MODEL\_ORTHOTROPIC\_MASONRY\_PLASTIC\_2D', 'MODEL\_ORTHOTROPIC\_PLASTIC\_2D', 'MODEL\_ORTHOTROPIC\_PLASTIC\_3D', 'MODEL\_ORTHOTROPIC\_TIMBER\_LINEAR\_ELASTIC\_SURFACES' }
  - **application\_context** - optional; type **material\_application\_context** - type **undefined** with restriction - enum { 'ALUMINUM\_DESIGN', 'COMBINATION\_WIZARD', 'CONCRETE\_DESIGN', 'CONCRETE\_FOUNDATION\_DESIGN', 'CRANEWAY\_DESIGN', 'DYNAMIC\_ANALYSIS', 'GEOTECHNICAL\_ANALYSIS', 'GLASS\_DESIGN', 'INVALID', 'INVALID\_DESIGN', 'LOAD\_WIZARD', 'MASONRY\_DESIGN', 'PIPING\_DESIGN', 'STEEL\_DESIGN', 'STEEL\_JOINT\_DESIGN', 'TIMBER\_DESIGN', 'TIMBER\_JOINT\_DESIGN', 'TOWER\_DESIGN' }
  - **diagram\_type** - optional; type **material\_diagram\_type** - type **undefined** with restriction - enum { 'DIAGRAM\_TYPE\_BASIC', 'DIAGRAM\_TYPE\_BILINEAR', 'DIAGRAM\_TYPE\_STRESS\_STRAIN\_DIAGRAM' }
  - **user\_defined\_name\_enabled** - optional; type **boolean**
  - **name** - optional; type **string**
  - **user\_defined** - optional; type **boolean**
  - **definition\_type** - optional; type **material\_definition\_type** - type **undefined** with restriction - enum { 'DERIVED\_G', 'DERIVED\_NU', 'E\_G\_NO\_NU', 'E\_G\_NU', 'NONE' }
  - **stress\_failure\_hypothesis** - optional; type **material\_stress\_failure\_hypothesis** - type **undefined** with restriction - enum { 'E\_STRESS\_FAILURE\_HYPOTHESIS\_OEDOMETRIC\_CONDITIONS\_WITH\_SMALL\_STRAIN\_STIFFNESS', 'STRESS\_FAILURE\_HYPOTHESIS\_DRUCKER\_PRAGER', 'STRESS\_FAILURE\_HYPOTHESIS\_MOHR\_COULOMB', 'STRESS\_FAILURE\_HYPOTHESIS\_OEDOMETRIC\_CONDITIONS', 'STRESS\_FAILURE\_HYPOTHESIS\_TRESCA', 'STRESS\_FAILURE\_HYPOTHESIS\_VON\_MISES' }
  - **is\_temperature\_dependent** - optional; type **boolean**
  - **is\_dynamic\_increase\_factor** - optional; type **boolean**
  - **has\_cost\_estimation** - optional; type **boolean**
  - **optimization** - optional; type **boolean**
  - **has\_emissions\_estimation** - optional; type **boolean**
  - **reference\_temperature** - optional; type **double**
  - **temperature\_properties\_sorted** - optional; type **boolean**
  - **stiffness\_matrix\_editable** - optional; type **boolean**
  - **stiffness\_modification** - optional; type **boolean**
  - **stiffness\_modification\_type** - optional; type **material\_stiffness\_modification\_type** - type **undefined** with restriction - enum { 'STIFFNESS\_MODIFICATION\_TYPE\_DIVISION', 'STIFFNESS\_MODIFICATION\_TYPE\_MULTIPLICATION' }
  - **has\_linear\_elastic\_with\_nonlinear\_criteria** - optional; type **boolean**
  - **comment** - optional; type **string**
  - **is\_generated** - optional; type **boolean**
  - **generating\_object\_info** - optional; type **string**
  - **members\_weight\_active** - optional; type **boolean**
  - **members\_weight\_unit\_cost** - optional; type **double**
  - **members\_weight\_unit** - optional; type **material\_members\_weight\_unit** - type **undefined** with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
  - **members\_weight\_quantity** - optional; type **double**
  - **members\_weight\_cost** - optional; type **double**
  - **members\_volume\_active** - optional; type **boolean**
  - **members\_volume\_unit\_cost** - optional; type **double**
  - **members\_volume\_unit** - optional; type **material\_members\_volume\_unit** - type **undefined** with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_US GAL', 'COST\_ESTIMATION\_YD3' }
  - **members\_volume\_quantity** - optional; type **double**
  - **members\_volume\_cost** - optional; type **double**
  - **members\_surface\_active** - optional; type **boolean**
  - **members\_surface\_unit\_cost** - optional; type **double**
  - **members\_surface\_unit** - optional; type **material\_members\_surface\_unit** - type **undefined** with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2',

'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }

- members\_surface\_quantity - optional; type *double*
- members\_surface\_cost - optional; type *double*
- surfaces\_weight\_active - optional; type *boolean*
- surfaces\_weight\_unit\_cost - optional; type *double*
- surfaces\_weight\_unit - optional; type *material\_surfaces\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- surfaces\_weight\_quantity - optional; type *double*
- surfaces\_weight\_cost - optional; type *double*
- surfaces\_volume\_active - optional; type *boolean*
- surfaces\_volume\_unit\_cost - optional; type *double*
- surfaces\_volume\_unit - optional; type *material\_surfaces\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- surfaces\_volume\_quantity - optional; type *double*
- surfaces\_volume\_cost - optional; type *double*
- surfaces\_top\_face\_active - optional; type *boolean*
- surfaces\_top\_face\_unit\_cost - optional; type *double*
- surfaces\_top\_face\_unit - optional; type *material\_surfaces\_top\_face\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_top\_face\_quantity - optional; type *double*
- surfaces\_top\_face\_cost - optional; type *double*
- surfaces\_area\_active - optional; type *boolean*
- surfaces\_area\_unit\_cost - optional; type *double*
- surfaces\_area\_unit - optional; type *material\_surfaces\_area\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_area\_quantity - optional; type *double*
- surfaces\_area\_cost - optional; type *double*
- surfaces\_bottom\_face\_active - optional; type *boolean*
- surfaces\_bottom\_face\_unit\_cost - optional; type *double*
- surfaces\_bottom\_face\_unit - optional; type *material\_surfaces\_bottom\_face\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- surfaces\_bottom\_face\_quantity - optional; type *double*
- surfaces\_bottom\_face\_cost - optional; type *double*
- solids\_weight\_active - optional; type *boolean*
- solids\_weight\_unit\_cost - optional; type *double*
- solids\_weight\_unit - optional; type *material\_solids\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- solids\_weight\_quantity - optional; type *double*
- solids\_weight\_cost - optional; type *double*
- solids\_volume\_active - optional; type *boolean*
- solids\_volume\_unit\_cost - optional; type *double*
- solids\_volume\_unit - optional; type *material\_solids\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMP GAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- solids\_volume\_quantity - optional; type *double*
- solids\_volume\_cost - optional; type *double*
- solids\_area\_active - optional; type *boolean*
- solids\_area\_unit\_cost - optional; type *double*
- solids\_area\_unit - optional; type *material\_solids\_area\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- solids\_area\_quantity - optional; type *double*
- solids\_area\_cost - optional; type *double*
- sum\_weight - optional; type *double*
- weight\_percentage - optional; type *double*
- cost\_percentage - optional; type *double*
- sum\_cost - optional; type *double*
- total\_cost - optional; type *double*
- emissions\_members\_weight\_active - optional; type *boolean*
- emissions\_members\_weight\_unit\_cost - optional; type *double*
- emissions\_members\_weight\_unit - optional; type *material\_emissions\_members\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG', 'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }
- emissions\_members\_weight\_quantity - optional; type *double*
- emissions\_members\_weight\_cost - optional; type *double*
- emissions\_members\_volume\_active - optional; type *boolean*
- emissions\_members\_volume\_unit\_cost - optional; type *double*
- emissions\_members\_volume\_unit - optional; type *material\_emissions\_members\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMP GAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_members\_volume\_quantity - optional; type *double*
- emissions\_members\_volume\_cost - optional; type *double*
- emissions\_members\_surface\_active - optional; type *boolean*
- emissions\_members\_surface\_unit\_cost - optional; type *double*
- emissions\_members\_surface\_unit - optional; type *material\_emissions\_members\_surface\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_members\_surface\_quantity - optional; type *double*
- emissions\_members\_surface\_cost - optional; type *double*
- emissions\_surfaces\_weight\_active - optional; type *boolean*
- emissions\_surfaces\_weight\_unit\_cost - optional; type *double*
- emissions\_surfaces\_weight\_unit - optional; type *material\_emissions\_surfaces\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG',

'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }

- emissions\_surfaces\_weight\_quantity - optional; type *double*
- emissions\_surfaces\_weight\_cost - optional; type *double*
- emissions\_surfaces\_volume\_active - optional; type *boolean*
- emissions\_surfaces\_volume\_unit\_cost - optional; type *double*
- emissions\_surfaces\_volume\_unit - optional; type *material\_emissions\_surfaces\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMPGAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_surfaces\_volume\_quantity - optional; type *double*
- emissions\_surfaces\_volume\_cost - optional; type *double*
- emissions\_surfaces\_top\_face\_active - optional; type *boolean*
- emissions\_surfaces\_top\_face\_unit\_cost - optional; type *double*
- emissions\_surfaces\_top\_face\_unit - optional; type *material\_emissions\_surfaces\_top\_face\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_top\_face\_quantity - optional; type *double*
- emissions\_surfaces\_top\_face\_cost - optional; type *double*
- emissions\_surfaces\_area\_active - optional; type *boolean*
- emissions\_surfaces\_area\_unit\_cost - optional; type *double*
- emissions\_surfaces\_area\_unit - optional; type *material\_emissions\_surfaces\_area\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_area\_quantity - optional; type *double*
- emissions\_surfaces\_area\_cost - optional; type *double*
- emissions\_surfaces\_bottom\_face\_active - optional; type *boolean*
- emissions\_surfaces\_bottom\_face\_unit\_cost - optional; type *double*
- emissions\_surfaces\_bottom\_face\_unit - optional; type *material\_emissions\_surfaces\_bottom\_face\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_surfaces\_bottom\_face\_quantity - optional; type *double*
- emissions\_surfaces\_bottom\_face\_cost - optional; type *double*
- emissions\_solids\_weight\_active - optional; type *boolean*
- emissions\_solids\_weight\_unit\_cost - optional; type *double*
- emissions\_solids\_weight\_unit - optional; type *material\_emissions\_solids\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CWT', 'EMISSION\_ESTIMATION\_G', 'EMISSION\_ESTIMATION\_KG', 'EMISSION\_ESTIMATION\_LB', 'EMISSION\_ESTIMATION\_OZ', 'EMISSION\_ESTIMATION\_SLUG', 'EMISSION\_ESTIMATION\_T', 'EMISSION\_ESTIMATION\_TON' }
- emissions\_solids\_weight\_quantity - optional; type *double*
- emissions\_solids\_weight\_cost - optional; type *double*
- emissions\_solids\_volume\_active - optional; type *boolean*
- emissions\_solids\_volume\_unit\_cost - optional; type *double*
- emissions\_solids\_volume\_unit - optional; type *material\_emissions\_solids\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM3', 'EMISSION\_ESTIMATION\_FLOZ', 'EMISSION\_ESTIMATION\_FT3', 'EMISSION\_ESTIMATION\_IMPGAL', 'EMISSION\_ESTIMATION\_IN3', 'EMISSION\_ESTIMATION\_L', 'EMISSION\_ESTIMATION\_M3', 'EMISSION\_ESTIMATION\_MM3', 'EMISSION\_ESTIMATION\_PT', 'EMISSION\_ESTIMATION\_QT', 'EMISSION\_ESTIMATION\_USGAL', 'EMISSION\_ESTIMATION\_YD3' }
- emissions\_solids\_volume\_quantity - optional; type *double*
- emissions\_solids\_volume\_cost - optional; type *double*
- emissions\_solids\_area\_active - optional; type *boolean*
- emissions\_solids\_area\_unit\_cost - optional; type *double*
- emissions\_solids\_area\_unit - optional; type *material\_emissions\_solids\_area\_unit* - type *undefined* with restriction - enum { 'EMISSION\_ESTIMATION\_CM2', 'EMISSION\_ESTIMATION\_FT2', 'EMISSION\_ESTIMATION\_IN2', 'EMISSION\_ESTIMATION\_M2', 'EMISSION\_ESTIMATION\_MM2', 'EMISSION\_ESTIMATION\_YD2' }
- emissions\_solids\_area\_quantity - optional; type *double*
- emissions\_solids\_area\_cost - optional; type *double*
- emissions\_percentage - optional; type *double*
- emissions\_sum - optional; type *double*
- emissions\_total - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_material\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_materialResponse*

## 172. set\_member

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member

**Input:** set\_member\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_member*

- value type *member*
  - no type *int*
  - type - optional; type *member\_type* - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE', 'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }
  - is\_deactivated\_for\_calculation - optional; type *boolean*
  - line - optional; type *int*
  - section\_distribution\_type - optional; type *member\_section\_distribution\_type* - type *undefined* with restriction - enum { ' ', 'SECTION\_DISTRIBUTION\_TYPE\_LINEAR', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
  - reference\_type - optional; type *member\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }

- nodes - optional; type *array\_of\_int*
- node\_start - optional; type *int*
- node\_end - optional; type *int*
- analytical\_length - optional; type *double*
- analytical\_volume - optional; type *double*
- analytical\_surface\_of\_coating - optional; type *double*
- analytical\_mass - optional; type *double*
- surface\_of\_coating - optional; type *double*
- analytical\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- analytical\_center\_of\_gravity\_x - optional; type *double*
- analytical\_center\_of\_gravity\_y - optional; type *double*
- analytical\_center\_of\_gravity\_z - optional; type *double*
- length - optional; type *double*
- volume - optional; type *double*
- mass - optional; type *double*
- center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- center\_of\_gravity\_x - optional; type *double*
- center\_of\_gravity\_y - optional; type *double*
- center\_of\_gravity\_z - optional; type *double*
- member\_representative - optional; type *int*
- design\_properties\_via\_member - optional; type *boolean*
- design\_properties\_via\_parent\_member\_set - optional; type *boolean*
- design\_properties\_parent\_member\_set - optional; type *int*
- comment - optional; type *string*
- member\_type\_rib\_alignment - optional; type *member\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
- member\_rib\_first\_surface - optional; type *int*
- member\_rib\_second\_surface - optional; type *int*
- member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
- align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
- flange\_dimensions - optional; type *array\_of\_member\_flange\_dimensions*
  - member\_flange\_dimensions - optional, unbounded; type *member\_flange\_dimensions*
    - no - optional; type *int*
    - end\_ordinate - optional; type *double*
    - length - optional; type *double*
    - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED' }
    - reference\_length - optional; type *double*
    - reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
    - width\_minus\_y\_integrative - optional; type *double*
    - width\_minus\_y\_effective - optional; type *double*
    - width\_minus\_y\_maximal - optional; type *double*
    - width\_plus\_y\_integrative - optional; type *double*
    - width\_plus\_y\_effective - optional; type *double*
    - width\_plus\_y\_maximal - optional; type *double*
    - distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- synchronize\_width\_mode - optional; type *boolean*
- relative\_ordinates\_mode - optional; type *boolean*
- member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_type\_definable\_stiffness - optional; type *int*
- result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- result\_beam\_y\_z - optional; type *double*
- result\_beam\_y\_plus - optional; type *double*
- result\_beam\_z\_plus - optional; type *double*
- result\_beam\_y\_minus - optional; type *double*
- result\_beam\_z\_minus - optional; type *double*
- result\_beam\_radius - optional; type *double*
- result\_beam\_include\_surfaces - optional; type *array\_of\_int*
- result\_beam\_include\_all\_surfaces - optional; type *boolean*
- result\_beam\_include\_solids - optional; type *array\_of\_int*
- result\_beam\_include\_all\_solids - optional; type *boolean*
- result\_beam\_include\_members - optional; type *array\_of\_int*
- result\_beam\_include\_all\_members - optional; type *boolean*
- result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- result\_beam\_exclude\_members - optional; type *array\_of\_int*
- projected\_length - optional; type *double*
- section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_start\_absolute - optional; type *double*
- section\_distance\_from\_start\_relative - optional; type *double*
- section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_end\_absolute - optional; type *double*
- section\_distance\_from\_end\_relative - optional; type *double*
- section\_alignment - optional; type *member\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- section\_start - optional; type *int*
- section\_end - optional; type *int*
- section\_internal - optional; type *int*
- section\_material - optional; type *int*
- curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- curved\_member\_is\_cantilevers - optional; type *boolean*

- *curved\_member\_cantilevers\_type* - optional; type *member\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- *rotation\_specification\_type* - optional; type *member\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- *rotation\_angle* - optional; type *double*
- *rotation\_help\_node* - optional; type *int*
- *rotation\_plane\_type* - optional; type *member\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- *rotation\_surface* - optional; type *int*
- *rotation\_surface\_plane\_type* - optional; type *member\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- *is\_rotated* - optional; type *boolean*
- *member\_hinge\_start* - optional; type *int*
- *member\_hinge\_end* - optional; type *int*
- *member\_eccentricity\_start* - optional; type *int*
- *member\_eccentricity\_end* - optional; type *int*
- *support* - optional; type *int*
- *member\_transverse\_stiffener* - optional; type *int*
- *member\_nonlinearity* - optional; type *int*
- *member\_result\_intermediate\_point* - optional; type *int*
- *stress\_analysis\_configuration* - optional; type *int*
- *aluminum\_effective\_lengths* - optional; type *int*
- *aluminum\_boundary\_conditions* - optional; type *int*
- *aluminum\_member\_local\_section\_reduction* - optional; type *int*
- *aluminum\_member\_transverse\_weld* - optional; type *int*
- *aluminum\_member\_shear\_panel* - optional; type *int*
- *aluminum\_member\_rotational\_restraint* - optional; type *int*
- *member\_aluminum\_design\_uls\_configuration* - optional; type *int*
- *member\_aluminum\_design\_sls\_configuration* - optional; type *int*
- *concrete\_effective\_lengths* - optional; type *int*
- *concrete\_cover\_user\_defined\_enabled* - optional; type *boolean*
- *concrete\_cover* - optional; type *double*
- *concrete\_cover\_top* - optional; type *double*
- *concrete\_cover\_left* - optional; type *double*
- *concrete\_cover\_right* - optional; type *double*
- *concrete\_cover\_bottom* - optional; type *double*
- *concrete\_cover\_min* - optional; type *array\_of\_int*
- *concrete\_cover\_min\_top* - optional; type *array\_of\_int*
- *concrete\_cover\_min\_left* - optional; type *array\_of\_int*
- *concrete\_cover\_min\_right* - optional; type *array\_of\_int*
- *concrete\_cover\_min\_bottom* - optional; type *array\_of\_int*
- *concrete\_cover\_different\_at\_section\_sides\_enabled* - optional; type *boolean*
- *concrete\_durability* - optional; type *int*
- *concrete\_durability\_top* - optional; type *int*
- *concrete\_durability\_left* - optional; type *int*
- *concrete\_durability\_right* - optional; type *int*
- *concrete\_durability\_bottom* - optional; type *int*
- *concrete\_shear\_reinforcement\_spans* - optional; type *array\_of\_member\_concrete\_shear\_reinforcement\_spans*
  - *member\_concrete\_shear\_reinforcement\_spans* - optional, unbounded; type *member\_concrete\_shear\_reinforcement\_spans*
    - *no* - optional; type *int*
    - *name* - optional; type *string*
    - *stirrup\_type* - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - *material* - optional; type *int*
    - *stirrup\_count* - optional; type *int*
    - *stirrup\_diameter* - optional; type *double*
    - *stirrup\_bar\_size\_type* - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - *stirrup\_distances* - optional; type *double*
    - *reinforcement\_area* - optional; type *double*
    - *span\_position\_reference\_type* - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - *span\_position\_reference\_internal\_node* - optional; type *int*
    - *span\_position\_reference\_x\_location\_relative* - optional; type *double*
    - *span\_position\_reference\_x\_location\_absolute* - optional; type *double*
    - *span\_position\_definition\_format\_type* - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - *span\_start\_relative* - optional; type *double*
    - *span\_start\_absolute* - optional; type *double*
    - *span\_end\_relative* - optional; type *double*
    - *span\_end\_absolute* - optional; type *double*
    - *span\_length* - optional; type *double*
    - *stirrup\_layout\_rule\_type* - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA' }
    - *stirrup\_start\_offset* - optional; type *double*
    - *stirrup\_end\_offset* - optional; type *double*
    - *one\_stirrup\_length* - optional; type *double*
    - *minimal\_and\_maximal\_stirrup\_length* - optional; type *string*

- length - optional; type *double*
- one\_stirrup\_weight - optional; type *double*
- minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_concrete\_longitudinal\_reinforcement\_items*
  - member\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_top\_side - optional; type *double*
    - additional\_offset\_bottom\_side - optional; type *double*
    - additional\_offset\_left\_side - optional; type *double*
    - additional\_offset\_right\_side - optional; type *double*
    - additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
    - additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }

'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }

- additional\_offset\_reference\_type\_at\_end\_type - optional; type *undefined* with restriction - enum {  
*additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum {  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',  
'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- steel\_effective\_lengths - optional; type *int*
- steel\_boundary\_conditions - optional; type *int*
- steel\_member\_local\_section\_reduction - optional; type *int*
- steel\_member\_transverse\_weld - optional; type *int*
- steel\_member\_shear\_panel - optional; type *int*
- steel\_member\_rotational\_restraint - optional; type *int*
- member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_steel\_design\_fr\_configuration - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- timber\_member\_local\_section\_reduction - optional; type *int*
- timber\_member\_shear\_panel - optional; type *int*
- timber\_member\_rotational\_restraint - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- end\_modifications\_member\_start\_extension - optional; type *double*
- end\_modifications\_member\_start\_slope\_y - optional; type *double*
- end\_modifications\_member\_start\_slope\_z - optional; type *double*
- end\_modifications\_member\_end\_extension - optional; type *double*
- end\_modifications\_member\_end\_slope\_y - optional; type *double*
- end\_modifications\_member\_end\_slope\_z - optional; type *double*
- has\_any\_end\_modifications - optional; type *boolean*
- deflection\_check\_direction - optional; type *member\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_start - optional; type *int*
- design\_support\_on\_member\_end - optional; type *int*
- design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_design\_supports\_on\_internal\_nodes*



- member\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_design\_supports\_on\_internal\_nodes*
  - no - optional; type *int*
  - node - optional; type *int*
  - design\_support - optional; type *int*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_y\_axis - optional; type *array\_of\_member\_deflection\_segments\_y\_axis*
  - member\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type *array\_of\_member\_deflection\_segments\_z\_axis*
  - member\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_memberResponse*

### 173. set\_member\_definable\_stiffness

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_definable\_stiffness

**Input:** set\_member\_definable\_stiffness\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_member\_definable\_stiffness*
- value type *member\_definable\_stiffness*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - assigned\_to - optional; type *array\_of\_int*
    - torsional\_stiffness - optional; type *double*
    - bending\_stiffness\_y - optional; type *double*
    - bending\_stiffness\_z - optional; type *double*
    - axial\_stiffness - optional; type *double*
    - shear\_stiffness\_y - optional; type *double*
    - shear\_stiffness\_z - optional; type *double*
    - specific\_weight - optional; type *double*
    - section\_area - optional; type *double*
    - rotation - optional; type *double*
    - thermal\_expansion\_alpha - optional; type *double*
    - thermal\_expansion\_width - optional; type *double*
    - thermal\_expansion\_height - optional; type *double*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_definable\_stiffness\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_definable\_stiffnessResponse*

### 174. set\_member\_eccentricity

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_eccentricity

**Input:** set\_member\_eccentricity\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_member\_eccentricity*
- value type *member\_eccentricity*
    - no type *int*
    - axial\_offset\_active - optional; type *boolean*
    - comment - optional; type *string*
    - coordinate\_system - optional; type *string*
    - generating\_object\_info - optional; type *string*
    - hinge\_location\_at\_node - optional; type *boolean*
    - horizontal\_section\_alignment - optional; type *member\_eccentricity\_horizontal\_section\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_RIGHT' }
    - is\_generated - optional; type *boolean*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - offset - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - offset\_x - optional; type *double*
    - offset\_y - optional; type *double*
    - offset\_z - optional; type *double*
    - specification\_type - optional; type *member\_eccentricity\_specification\_type* - type *undefined* with restriction - enum { 'TYPE\_ABSOLUTE', 'TYPE\_RELATIVE', 'TYPE\_RELATIVE\_AND\_ABSOLUTE' }
    - transverse\_offset\_reference\_type - optional; type *member\_eccentricity\_transverse\_offset\_reference\_type* - type *undefined* with restriction - enum { 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_MEMBER\_SECTION',

- `'TRANSVERSE_OFFSET_TYPE_FROM_SURFACE_THICKNESS'`, `'TRANSVERSE_OFFSET_TYPE_NONE'` }
- `transverse_offset_horizontal_alignment` - optional; type `member_eccentricity_transverse_offset_horizontal_alignment` - type *undefined* with restriction - enum { `'ALIGN_LEFT'`, `'ALIGN_MIDDLE'`, `'ALIGN_RIGHT'` }
- `transverse_offset_member_reference_node` - optional; type `int`
- `transverse_offset_reference_member` - optional; type `int`
- `transverse_offset_reference_surface` - optional; type `int`
- `transverse_offset_surface_reference_node` - optional; type `int`
- `transverse_offset_vertical_alignment` - optional; type `member_eccentricity_transverse_offset_vertical_alignment` - type *undefined* with restriction - enum { `'ALIGN_BOTTOM'`, `'ALIGN_MIDDLE'`, `'ALIGN_TOP'` }
- `vertical_section_alignment` - optional; type `member_eccentricity_vertical_section_alignment` - type *undefined* with restriction - enum { `'ALIGN_BOTTOM'`, `'ALIGN_MIDDLE'`, `'ALIGN_TOP'` }
- `id_for_export_import` - optional; type `string`
- `metadata_for_export_import` - optional; type `string`

**Output:** `set_member_eccentricity_response` (soap:body, use = literal) [Source code](#)

parameters type `set_member_eccentricityResponse`

## 175. `set_member_hinge`

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_member_hinge`

**Input:** `set_member_hinge_request` (soap:body, use = literal) [Source code](#)

parameters type `set_member_hinge`

- value type `member_hinge`
  - no type `int`
  - `user_defined_name_enabled` - optional; type `boolean`
  - `name` - optional; type `string`
  - `coordinate_system` - optional; type `string`
  - `axial_release_n` - optional; type `double`
  - `axial_release_n_nonlinearity` - optional; type `member_hinge_axial_release_n_nonlinearity` - type *undefined* with restriction - enum { `'NONLINEARITY_TYPE_DIAGRAM'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_2'`, `'NONLINEARITY_TYPE_NONE'`, `'NONLINEARITY_TYPE_PARTIAL_ACTIVITY'`, `'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM'` }
  - `axial_release_vy` - optional; type `double`
  - `axial_release_vy_nonlinearity` - optional; type `member_hinge_axial_release_vy_nonlinearity` - type *undefined* with restriction - enum { `'NONLINEARITY_TYPE_DIAGRAM'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_2'`, `'NONLINEARITY_TYPE_NONE'`, `'NONLINEARITY_TYPE_PARTIAL_ACTIVITY'`, `'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM'` }
  - `axial_release_vz` - optional; type `double`
  - `axial_release_vz_nonlinearity` - optional; type `member_hinge_axial_release_vz_nonlinearity` - type *undefined* with restriction - enum { `'NONLINEARITY_TYPE_DIAGRAM'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE'`, `'NONLINEARITY_TYPE_FAILURE_IF_POSITIVE'`, `'NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2'`, `'NONLINEARITY_TYPE_FRICTION_DIRECTION_2'`, `'NONLINEARITY_TYPE_NONE'`, `'NONLINEARITY_TYPE_PARTIAL_ACTIVITY'`, `'NONLINEARITY_TYPE_STIFFNESS_DIAGRAM'` }
  - `comment` - optional; type `string`
  - `diagram_along_x_end` - optional; type `member_hinge_diagram_along_x_end` - type *undefined* with restriction - enum { `'DIAGRAM_ENDING_TYPE_CONTINUOUS'`, `'DIAGRAM_ENDING_TYPE_STOP'`, `'DIAGRAM_ENDING_TYPE_TEARING'`, `'DIAGRAM_ENDING_TYPE_YIELDING'` }
  - `diagram_along_x_is_sorted` - optional; type `boolean`
  - `diagram_along_x_start` - optional; type `member_hinge_diagram_along_x_start` - type *undefined* with restriction - enum { `'DIAGRAM_ENDING_TYPE_CONTINUOUS'`, `'DIAGRAM_ENDING_TYPE_STOP'`, `'DIAGRAM_ENDING_TYPE_TEARING'`, `'DIAGRAM_ENDING_TYPE_YIELDING'` }
  - `diagram_along_x_symmetric` - optional; type `boolean`
  - `diagram_along_x_table` - optional; type `array_of_member_hinge_diagram_along_x_table`
    - `member_hinge_diagram_along_x_table` - optional, unbounded; type `member_hinge_diagram_along_x_table`
      - no - optional; type `int`
      - `displacement` - optional; type `double`
      - `force` - optional; type `double`
      - `spring` - optional; type `double`
      - `note` - optional; type `string`
  - `diagram_along_y_end` - optional; type `member_hinge_diagram_along_y_end` - type *undefined* with restriction - enum { `'DIAGRAM_ENDING_TYPE_CONTINUOUS'`, `'DIAGRAM_ENDING_TYPE_STOP'`, `'DIAGRAM_ENDING_TYPE_TEARING'`, `'DIAGRAM_ENDING_TYPE_YIELDING'` }
  - `diagram_along_y_is_sorted` - optional; type `boolean`
  - `diagram_along_y_start` - optional; type `member_hinge_diagram_along_y_start` - type *undefined* with restriction - enum { `'DIAGRAM_ENDING_TYPE_CONTINUOUS'`, `'DIAGRAM_ENDING_TYPE_STOP'`, `'DIAGRAM_ENDING_TYPE_TEARING'`, `'DIAGRAM_ENDING_TYPE_YIELDING'` }
  - `diagram_along_y_symmetric` - optional; type `boolean`
  - `diagram_along_y_table` - optional; type `array_of_member_hinge_diagram_along_y_table`
    - `member_hinge_diagram_along_y_table` - optional, unbounded; type `member_hinge_diagram_along_y_table`
      - no - optional; type `int`
      - `displacement` - optional; type `double`
      - `force` - optional; type `double`
      - `spring` - optional; type `double`
      - `note` - optional; type `string`
  - `diagram_along_z_end` - optional; type `member_hinge_diagram_along_z_end` - type *undefined* with restriction - enum { `'DIAGRAM_ENDING_TYPE_CONTINUOUS'`, `'DIAGRAM_ENDING_TYPE_STOP'`, `'DIAGRAM_ENDING_TYPE_TEARING'`, `'DIAGRAM_ENDING_TYPE_YIELDING'` }
  - `diagram_along_z_is_sorted` - optional; type `boolean`

- diagram\_along\_z\_start - optional; type *member\_hinge\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_z\_symmetric - optional; type *boolean*
- diagram\_along\_z\_table - optional; type *array\_of\_member\_hinge\_diagram\_along\_z\_table*
  - member\_hinge\_diagram\_along\_z\_table - optional, unbounded; type *member\_hinge\_diagram\_along\_z\_table*
    - no - optional; type *int*
    - displacement - optional; type *double*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_x\_end - optional; type *member\_hinge\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_is\_sorted - optional; type *boolean*
- diagram\_around\_x\_start - optional; type *member\_hinge\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_symmetric - optional; type *boolean*
- diagram\_around\_x\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_x\_table*
  - member\_hinge\_diagram\_around\_x\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_x\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_y\_end - optional; type *member\_hinge\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_is\_sorted - optional; type *boolean*
- diagram\_around\_y\_start - optional; type *member\_hinge\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_symmetric - optional; type *boolean*
- diagram\_around\_y\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_y\_table*
  - member\_hinge\_diagram\_around\_y\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_y\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_z\_end - optional; type *member\_hinge\_diagram\_around\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_is\_sorted - optional; type *boolean*
- diagram\_around\_z\_start - optional; type *member\_hinge\_diagram\_around\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_symmetric - optional; type *boolean*
- diagram\_around\_z\_table - optional; type *array\_of\_member\_hinge\_diagram\_around\_z\_table*
  - member\_hinge\_diagram\_around\_z\_table - optional, unbounded; type *member\_hinge\_diagram\_around\_z\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- friction\_coefficient\_x - optional; type *double*
- friction\_coefficient\_xy - optional; type *double*
- friction\_coefficient\_xz - optional; type *double*
- friction\_coefficient\_y - optional; type *double*
- friction\_coefficient\_yx - optional; type *double*
- friction\_coefficient\_yz - optional; type *double*
- friction\_coefficient\_z - optional; type *double*
- friction\_coefficient\_zx - optional; type *double*
- friction\_coefficient\_zy - optional; type *double*
- friction\_direction\_independent\_x - optional; type *boolean*
- friction\_direction\_independent\_y - optional; type *boolean*
- friction\_direction\_independent\_z - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- is\_generated - optional; type *boolean*
- members - optional; type *string*
- moment\_release\_mt - optional; type *double*
- moment\_release\_mt\_nonlinearity - optional; type *member\_hinge\_moment\_release\_mt\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- moment\_release\_my - optional; type *double*
- moment\_release\_my\_nonlinearity - optional; type *member\_hinge\_moment\_release\_my\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- moment\_release\_mz - optional; type *double*
- moment\_release\_mz\_nonlinearity - optional; type *member\_hinge\_moment\_release\_mz\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE',



- 'PARTIAL\_ACTIVITY\_TYPE\_FIXED', 'PARTIAL\_ACTIVITY\_TYPE\_INEFFECTIVENESS', 'PARTIAL\_ACTIVITY\_TYPE\_TEARING', 'PARTIAL\_ACTIVITY\_TYPE\_YIELDING' }
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_hinge\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_hingeResponse*

## 176. set\_member\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_imperfection

**Input:** set\_member\_imperfection\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_imperfection*

- imperfection\_case\_no type *int*
- value type *member\_imperfection*
  - no type *int*
  - imperfection\_type - optional; type *member\_imperfection\_imperfection\_type* - type *undefined* with restriction - enum { 'IMPERFECTION\_TYPE\_INITIAL\_BOW', 'IMPERFECTION\_TYPE\_INITIAL\_BOW\_AND\_CRITERION', 'IMPERFECTION\_TYPE\_INITIAL\_SWAY' }
  - members - optional; type *array\_of\_int*
  - imperfection\_case - optional; type *int*
  - definition\_type - optional; type *member\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_ABSOLUTE', 'DEFINITION\_TYPE\_ANSI\_CURRENT', 'DEFINITION\_TYPE\_ANSI\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_CSA\_CURRENT', 'DEFINITION\_TYPE\_CSA\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_EN\_1992\_1\_1993\_1', 'DEFINITION\_TYPE\_EN\_1993\_1\_1', 'DEFINITION\_TYPE\_EN\_1995\_1\_1', 'DEFINITION\_TYPE\_EN\_1999\_1\_1', 'DEFINITION\_TYPE\_GB\_50017\_2017', 'DEFINITION\_TYPE\_GB\_50017\_2017\_CURRENT', 'DEFINITION\_TYPE\_GB\_50017\_2017\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_NOTIONAL\_LOAD', 'DEFINITION\_TYPE\_RELATIVE' }
  - coordinate\_system - optional; type *string*
  - imperfection\_direction - optional; type *member\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V\_NEGATIVE' }
  - basic\_value\_absolute - optional; type *double*
  - basic\_value\_coefficient - optional; type *double*
  - basic\_value\_relative - optional; type *double*
  - basic\_value\_force - optional; type *double*
  - section\_design - optional; type *member\_imperfection\_section\_design* - type *undefined* with restriction - enum { 'SECTION\_DESIGN\_ELASTIC', 'SECTION\_DESIGN\_PLASTIC' }
  - active\_criterion - optional; type *member\_imperfection\_active\_criterion* - type *undefined* with restriction - enum { 'ACTIVITY\_CRITERION\_ALWAYS', 'ACTIVITY\_CRITERION\_DEFINE', 'ACTIVITY\_CRITERION\_DIN\_18800', 'ACTIVITY\_CRITERION\_EN\_1993', 'ACTIVITY\_CRITERION\_EN\_1999' }
  - active\_bow - optional; type *double*
  - standard\_factor\_enumeration - optional; type *member\_imperfection\_standard\_factor\_enumeration* - type *undefined* with restriction - enum { 'STANDARD\_FACTOR\_ASD', 'STANDARD\_FACTOR\_LRFD' }
  - standard\_factor\_number - optional; type *double*
  - height - optional; type *double*
  - column\_in\_row - optional; type *int*
  - number\_of\_floors - optional; type *int*
  - case\_object - optional; type *int*
  - reduction\_factor\_h - optional; type *double*
  - reduction\_factor\_m - optional; type *double*
  - initial\_sway - optional; type *double*
  - initial\_sway\_inverted - optional; type *double*
  - delta - optional; type *double*
  - parameters - optional; type *array\_of\_int*
  - reference\_to\_list\_of\_members - optional; type *boolean*
  - refer\_distance\_from\_objects\_to\_assign - optional; type *boolean*
  - imperfection\_over\_total\_length\_of\_objects\_to\_assign - optional; type *boolean*
  - distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_a\_relative - optional; type *double*
  - distance\_b\_relative - optional; type *double*
  - distance\_a\_absolute - optional; type *double*
  - distance\_b\_absolute - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_imperfection\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_imperfectionResponse*

## 177. set\_member\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_load

**Input:** set\_member\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_load*

- load\_case\_no type *int*
- value type *member\_load*
  - no type *int*

- `load_type` - optional; type *member\_load\_load\_type* - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_AXIAL\_DISPLACEMENT', 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_DISPLACEMENT', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_INITIAL\_PRESTRESS', 'LOAD\_TYPE\_MOMENT', 'LOAD\_TYPE\_PIPE\_CONTENT\_FULL', 'LOAD\_TYPE\_PIPE\_CONTENT\_PARTIAL', 'LOAD\_TYPE\_PIPE\_INTERNAL\_PRESSURE', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY\_MOTION', 'LOAD\_TYPE\_ROTATION', 'LOAD\_TYPE\_TEMPERATURE', 'LOAD\_TYPE\_TEMPERATURE\_CHANGE' }
- `members` - optional; type *array\_of\_int*
- `load_case` - optional; type *int*
- `coordinate_system` - optional; type *string*
- `load_distribution` - optional; type *member\_load\_load\_distribution* - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }
- `load_direction` - optional; type *member\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_PRINCIPAL\_U', 'LOAD\_DIRECTION\_PRINCIPAL\_V' }
- `load_direction_orientation` - optional; type *member\_load\_load\_direction\_orientation* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
- `form_finding_definition_type` - optional; type *member\_load\_form\_finding\_definition\_type* - type *undefined* with restriction - enum { 'FORM\_FINDING\_TYPE\_FORCE', 'FORM\_FINDING\_TYPE\_GEOMETRIC' }
- `magnitude` - optional; type *double*
- `magnitude_1` - optional; type *double*
- `magnitude_2` - optional; type *double*
- `magnitude_3` - optional; type *double*
- `magnitude_t_c` - optional; type *double*
- `magnitude_t_c_1` - optional; type *double*
- `magnitude_t_c_2` - optional; type *double*
- `magnitude_t_c_3` - optional; type *double*
- `magnitude_delta_t` - optional; type *double*
- `magnitude_delta_t_1` - optional; type *double*
- `magnitude_delta_t_2` - optional; type *double*
- `magnitude_delta_t_3` - optional; type *double*
- `magnitude_t_t` - optional; type *double*
- `magnitude_t_t_1` - optional; type *double*
- `magnitude_t_t_2` - optional; type *double*
- `magnitude_t_t_3` - optional; type *double*
- `magnitude_t_b` - optional; type *double*
- `magnitude_t_b_1` - optional; type *double*
- `magnitude_t_b_2` - optional; type *double*
- `magnitude_t_b_3` - optional; type *double*
- `individual_mass_components` - optional; type *boolean*
- `mass_global` - optional; type *double*
- `mass_x` - optional; type *double*
- `mass_y` - optional; type *double*
- `mass_z` - optional; type *double*
- `distance_a_is_defined_as_relative` - optional; type *boolean*
- `distance_a_absolute` - optional; type *double*
- `distance_a_relative` - optional; type *double*
- `distance_b_is_defined_as_relative` - optional; type *boolean*
- `distance_b_absolute` - optional; type *double*
- `distance_b_relative` - optional; type *double*
- `distance_c_is_defined_as_relative` - optional; type *boolean*
- `distance_c_absolute` - optional; type *double*
- `distance_c_relative` - optional; type *double*
- `count_n` - optional; type *int*
- `varying_load_parameters_are_defined_as_relative` - optional; type *boolean*
- `varying_load_parameters` - optional; type *array\_of\_member\_load\_varying\_load\_parameters*
  - *member\_load\_varying\_load\_parameters* - optional, unbounded; type *member\_load\_varying\_load\_parameters*
    - `no` - optional; type *int*
    - `distance` - optional; type *double*
    - `delta_distance` - optional; type *double*
    - `magnitude` - optional; type *double*
    - `note` - optional; type *string*
    - `magnitude_t_c` - optional; type *double*
    - `magnitude_delta_t` - optional; type *double*
    - `magnitude_t_t` - optional; type *double*
    - `magnitude_t_b` - optional; type *double*
- `varying_load_parameters_sorted` - optional; type *boolean*
- `angular_velocity` - optional; type *double*
- `angular_acceleration` - optional; type *double*
- `axis_definition_type` - optional; type *member\_load\_axis\_definition\_type* - type *undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- `axis_definition_p1` - optional; type *vector\_3d*
  - `x` type *double*
  - `y` type *double*
  - `z` type *double*
- `axis_definition_p1_x` - optional; type *double*
- `axis_definition_p1_y` - optional; type *double*
- `axis_definition_p1_z` - optional; type *double*
- `axis_definition_p2` - optional; type *vector\_3d*
  - `x` type *double*
  - `y` type *double*
  - `z` type *double*
- `axis_definition_p2_x` - optional; type *double*
- `axis_definition_p2_y` - optional; type *double*
- `axis_definition_p2_z` - optional; type *double*

- axis\_definition\_axis - optional; type *member\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *member\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- filling\_height - optional; type *double*
- reference\_to\_list\_of\_members - optional; type *boolean*
- distance\_from\_member\_end - optional; type *boolean*
- load\_is\_over\_total\_length - optional; type *boolean*
- has\_force\_eccentricity - optional; type *boolean*
- eccentricity\_horizontal\_alignment - optional; type *member\_load\_eccentricity\_horizontal\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_RIGHT' }
- eccentricity\_vertical\_alignment - optional; type *member\_load\_eccentricity\_vertical\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_TOP' }
- eccentricity\_section\_middle - optional; type *member\_load\_eccentricity\_section\_middle* - type *undefined* with restriction - enum { 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_CENTER\_OF\_GRAVITY', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_NONE', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_SHEAR\_CENTER' }
- is\_eccentricity\_at\_end\_different\_from\_start - optional; type *boolean*
- eccentricity\_y\_at\_start - optional; type *double*
- eccentricity\_z\_at\_start - optional; type *double*
- eccentricity\_y\_at\_end - optional; type *double*
- eccentricity\_z\_at\_end - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- form\_finding\_internal\_force - optional; type *member\_load\_form\_finding\_internal\_force* - type *undefined* with restriction - enum { 'FORM\_FINDING\_INTERNAL\_FORCE\_COMPRESSION', 'FORM\_FINDING\_INTERNAL\_FORCE\_TENSION' }
- form\_finding\_geometry\_definition - optional; type *member\_load\_form\_finding\_geometry\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LENGTH', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LOW\_POINT\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_MAXIMUM\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_UNSTRESSED\_LENGTH' }
- form\_finding\_force\_definition - optional; type *member\_load\_form\_finding\_force\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_AVERAGE', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_DENSITY', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_HORIZONTAL\_TENSION\_COMPONENT', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MAXIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_J\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_J\_END' }
- form\_finding\_magnitude\_is\_defined\_as\_relative - optional; type *boolean*
- form\_finding\_magnitude\_absolute - optional; type *double*
- form\_finding\_magnitude\_relative - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_loadResponse*

## 178. set\_member\_nonlinearity

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_nonlinearity

**Input:** set\_member\_nonlinearity\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_member\_nonlinearity*
- value type *member\_nonlinearity*
    - no type *int*
    - type - optional; type *member\_nonlinearity\_type* - type *undefined* with restriction - enum { 'TYPE\_FAILURE\_IF\_COMPRESSION', 'TYPE\_FAILURE\_IF\_COMPRESSION\_WITH\_SLIPPAGE', 'TYPE\_FAILURE\_IF\_TENSION', 'TYPE\_FAILURE\_IF\_TENSION\_WITH\_SLIPPAGE', 'TYPE\_SLIPPAGE', 'TYPE\_TEARING', 'TYPE\_TEARING\_IF\_COMPRESSION', 'TYPE\_TEARING\_IF\_TENSION', 'TYPE\_YIELDING', 'TYPE\_YIELDING\_IF\_COMPRESSION', 'TYPE\_YIELDING\_IF\_TENSION' }
    - assigned\_to - optional; type *array\_of\_int*
    - comment - optional; type *string*
    - compression\_force - optional; type *double*
    - generating\_object\_info - optional; type *string*
    - is\_generated - optional; type *boolean*
    - name - optional; type *string*
    - slippage - optional; type *double*
    - tension\_force - optional; type *double*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_nonlinearity\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_nonlinearityResponse*

## 179. set\_member\_representative

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_representative

**Input:** set\_member\_representative\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_member\_representative*
- value type *member\_representative*
    - no type *int*
    - type - optional; type *member\_representative\_type* - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE',

'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }

- user\_defined\_name\_enabled - optional; type *boolean*
- name - optional; type *string*
- comment - optional; type *string*
- number\_of\_members - optional; type *int*
- total\_length - optional; type *double*
- total\_volume - optional; type *double*
- total\_mass - optional; type *double*
- total\_surface\_of\_coating - optional; type *double*
- nodes\_on\_member\_from\_start - optional; type *array\_of\_int*
- is\_deactivated\_for\_calculation - optional; type *boolean*
- line - optional; type *int*
- section\_distribution\_type - optional; type *member\_representative\_section\_distribution\_type* - type *undefined* with restriction - enum { "SECTION\_DISTRIBUTION\_TYPE\_LINEAR", 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
- reference\_type - optional; type *member\_representative\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
- nodes - optional; type *array\_of\_int*
- node\_start - optional; type *int*
- node\_end - optional; type *int*
- analytical\_length - optional; type *double*
- analytical\_volume - optional; type *double*
- analytical\_mass - optional; type *double*
- analytical\_surface\_of\_coating - optional; type *double*
- analytical\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- analytical\_center\_of\_gravity\_x - optional; type *double*
- analytical\_center\_of\_gravity\_y - optional; type *double*
- analytical\_center\_of\_gravity\_z - optional; type *double*
- length - optional; type *double*
- volume - optional; type *double*
- mass - optional; type *double*
- surface\_of\_coating - optional; type *double*
- center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- center\_of\_gravity\_x - optional; type *double*
- center\_of\_gravity\_y - optional; type *double*
- center\_of\_gravity\_z - optional; type *double*
- member\_representative - optional; type *int*
- design\_properties\_via\_member - optional; type *boolean*
- design\_properties\_via\_parent\_member\_set - optional; type *boolean*
- design\_properties\_parent\_member\_set - optional; type *int*
- member\_type\_rib\_alignment - optional; type *member\_representative\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
- member\_rib\_first\_surface - optional; type *int*
- member\_rib\_second\_surface - optional; type *int*
- member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
- member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
- result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_representative\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- flange\_dimensions - optional; type *array\_of\_member\_representative\_flange\_dimensions*
  - member\_representative\_flange\_dimensions - optional, unbounded; type *member\_representative\_flange\_dimensions*
    - no - optional; type *int*
    - end\_ordinate - optional; type *double*
    - length - optional; type *double*
    - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED' }
    - reference\_length - optional; type *double*
    - reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
    - width\_minus\_y\_integrative - optional; type *double*
    - width\_minus\_y\_effictive - optional; type *double*
    - width\_minus\_y\_maximal - optional; type *double*
    - width\_plus\_y\_integrative - optional; type *double*
    - width\_plus\_y\_effictive - optional; type *double*
    - width\_plus\_y\_maximal - optional; type *double*
    - distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- synchronize\_width\_mode - optional; type *boolean*
- relative\_ordinates\_mode - optional; type *boolean*
- member\_type\_definable\_stiffness - optional; type *int*
- result\_beam\_y\_z - optional; type *double*
- result\_beam\_y\_plus - optional; type *double*
- result\_beam\_z\_plus - optional; type *double*
- result\_beam\_y\_minus - optional; type *double*
- result\_beam\_z\_minus - optional; type *double*
- result\_beam\_radius - optional; type *double*
- result\_beam\_include\_surfaces - optional; type *array\_of\_int*



- result\_beam\_include\_all\_surfaces - optional; type *boolean*
- result\_beam\_include\_solids - optional; type *array\_of\_int*
- result\_beam\_include\_all\_solids - optional; type *boolean*
- result\_beam\_include\_members - optional; type *array\_of\_int*
- result\_beam\_include\_all\_members - optional; type *boolean*
- result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- result\_beam\_exclude\_members - optional; type *array\_of\_int*
- projected\_length - optional; type *double*
- section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_start\_absolute - optional; type *double*
- section\_distance\_from\_start\_relative - optional; type *double*
- section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- section\_distance\_from\_end\_absolute - optional; type *double*
- section\_distance\_from\_end\_relative - optional; type *double*
- section\_alignment - optional; type *member\_representative\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- curved\_member\_is\_cantilevers - optional; type *boolean*
- curved\_member\_cantilevers\_type - optional; type *member\_representative\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- rotation\_specification\_type - optional; type *member\_representative\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- rotation\_angle - optional; type *double*
- rotation\_help\_node - optional; type *int*
- rotation\_plane\_type - optional; type *member\_representative\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- rotation\_surface - optional; type *int*
- rotation\_surface\_plane\_type - optional; type *member\_representative\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- is\_rotated - optional; type *boolean*
- section\_start - optional; type *int*
- section\_end - optional; type *int*
- section\_internal - optional; type *int*
- section\_material - optional; type *int*
- members - optional; type *array\_of\_int*
- member\_hinge\_start - optional; type *int*
- member\_hinge\_end - optional; type *int*
- member\_eccentricity\_start - optional; type *int*
- member\_eccentricity\_end - optional; type *int*
- support - optional; type *int*
- member\_transverse\_stiffener - optional; type *int*
- member\_nonlinearity - optional; type *int*
- member\_result\_intermediate\_point - optional; type *int*
- concrete\_effective\_lengths - optional; type *int*
- steel\_effective\_lengths - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- aluminum\_effective\_lengths - optional; type *int*
- concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_representative\_concrete\_shear\_reinforcement\_spans*
  - member\_representative\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_representative\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA' }
    - stirrup\_start\_offset - optional; type *double*

- stirrup\_end\_offset - optional; type *double*
- one\_stirrup\_length - optional; type *double*
- minimal\_and\_maximal\_stirrup\_length - optional; type *string*
- length - optional; type *double*
- one\_stirrup\_weight - optional; type *double*
- minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_representative\_concrete\_longitudinal\_reinforcement\_items*
  - member\_representative\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_representative\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_top\_side - optional; type *double*
    - additional\_offset\_bottom\_side - optional; type *double*
    - additional\_offset\_left\_side - optional; type *double*
    - additional\_offset\_right\_side - optional; type *double*
    - additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
    - additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER',

- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER',
- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_end\_type - optional; type *double*
- *additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- concrete\_cover - optional; type *double*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- concrete\_cover\_left - optional; type *double*
- concrete\_cover\_right - optional; type *double*
- concrete\_cover\_min - optional; type *array\_of\_int*
- concrete\_cover\_min\_top - optional; type *array\_of\_int*
- concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- concrete\_cover\_min\_left - optional; type *array\_of\_int*
- concrete\_cover\_min\_right - optional; type *array\_of\_int*
- concrete\_durability - optional; type *int*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- concrete\_durability\_left - optional; type *int*
- concrete\_durability\_right - optional; type *int*
- steel\_boundary\_conditions - optional; type *int*
- steel\_member\_local\_section\_reduction - optional; type *int*
- steel\_member\_transverse\_weld - optional; type *int*
- steel\_member\_shear\_panel - optional; type *int*
- steel\_member\_rotational\_restraint - optional; type *int*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- timber\_member\_local\_section\_reduction - optional; type *int*
- timber\_member\_shear\_panel - optional; type *int*
- timber\_member\_rotational\_restraint - optional; type *int*
- aluminum\_boundary\_conditions - optional; type *int*
- aluminum\_member\_local\_section\_reduction - optional; type *int*
- aluminum\_member\_transverse\_weld - optional; type *int*
- aluminum\_member\_shear\_panel - optional; type *int*
- aluminum\_member\_rotational\_restraint - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_uls\_configuration - optional; type *int*

- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- end\_modifications\_member\_start\_extension - optional; type *double*
- end\_modifications\_member\_start\_slope\_y - optional; type *double*
- end\_modifications\_member\_start\_slope\_z - optional; type *double*
- end\_modifications\_member\_end\_extension - optional; type *double*
- end\_modifications\_member\_end\_slope\_y - optional; type *double*
- end\_modifications\_member\_end\_slope\_z - optional; type *double*
- has\_any\_end\_modifications - optional; type *boolean*
- deflection\_check\_direction - optional; type *member\_representative\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_representative\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_start - optional; type *int*
- design\_support\_on\_member\_end - optional; type *int*
- design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_representative\_design\_supports\_on\_internal\_nodes*
  - member\_representative\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_representative\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_y\_axis - optional; type *array\_of\_member\_representative\_deflection\_segments\_y\_axis*
  - member\_representative\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_representative\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type *array\_of\_member\_representative\_deflection\_segments\_z\_axis*
  - member\_representative\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_representative\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_representative\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_representativeResponse*

#### 180. set\_member\_result\_intermediate\_point

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_result\_intermediate\_point

**Input:** set\_member\_result\_intermediate\_point\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_member\_result\_intermediate\_point*
- value type *member\_result\_intermediate\_point*
    - no type *int*
    - comment - optional; type *string*
    - distances - optional; type *array\_of\_member\_result\_intermediate\_point\_distances*
      - member\_result\_intermediate\_point\_distances - optional, unbounded; type *member\_result\_intermediate\_point\_distances*
        - no - optional; type *int*
        - value - optional; type *double*
        - note - optional; type *string*
    - distances\_are\_defined\_as\_absolute - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - is\_generated - optional; type *boolean*
    - members - optional; type *array\_of\_int*
    - name - optional; type *string*
    - point\_count - optional; type *int*
    - uniform\_distribution - optional; type *boolean*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_result\_intermediate\_point\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_result\_intermediate\_pointResponse*

#### 181. set\_member\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_set

```

parameters type set_member_set
 ■ value type member_set
 ■ no type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ set_type - optional; type member_set_set_type - type undefined with restriction - enum {
 'SET_TYPE_CONTINUOUS', 'SET_TYPE_GROUP' }
 ■ length - optional; type double
 ■ center_of_gravity - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ center_of_gravity_x - optional; type double
 ■ center_of_gravity_y - optional; type double
 ■ center_of_gravity_z - optional; type double
 ■ position - optional; type string
 ■ position_short - optional; type string
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ volume - optional; type double
 ■ mass - optional; type double
 ■ member_set_representative - optional; type int
 ■ discontinuous_torsional_warping - optional; type boolean
 ■ design_properties_activated - optional; type boolean
 ■ steel_effective_lengths - optional; type int
 ■ steel_boundary_conditions - optional; type int
 ■ steel_member_local_section_reductions - optional; type int
 ■ steel_member_shear_panels - optional; type int
 ■ steel_member_rotational_restraints - optional; type int
 ■ member_steel_design_uls_configuration - optional; type int
 ■ member_steel_design_sls_configuration - optional; type int
 ■ member_steel_design_fr_configuration - optional; type int
 ■ member_steel_design_seismic_configuration - optional; type int
 ■ aluminum_effective_lengths - optional; type int
 ■ aluminum_boundary_conditions - optional; type int
 ■ aluminum_member_local_section_reductions - optional; type int
 ■ aluminum_member_shear_panels - optional; type int
 ■ aluminum_member_rotational_restraints - optional; type int
 ■ aluminum_member_transverse_weld - optional; type int
 ■ member_aluminum_design_uls_configuration - optional; type int
 ■ member_aluminum_design_sls_configuration - optional; type int
 ■ member_set_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled - optional; type boolean
 ■ concrete_shear_reinforcement_spans - optional; type
 array_of_member_set_concrete_shear_reinforcement_spans
 ■ member_set_concrete_shear_reinforcement_spans - optional, unbounded; type
 member_set_concrete_shear_reinforcement_spans
 ■ no - optional; type int
 ■ name - optional; type string
 ■ stirrup_type - optional; type stirrup_type - type undefined with restriction - enum {
 'STIRRUP_TYPE_FOUR_LEGGED_CLOSED_HOOK_135',
 'STIRRUP_TYPE_FOUR_LEGGED_CLOSED_HOOK_90',
 'STIRRUP_TYPE_FOUR_LEGGED_OVERLAP_HOOK_180',
 'STIRRUP_TYPE_THREE_LEGGED_CLOSED_HOOK_135',
 'STIRRUP_TYPE_THREE_LEGGED_CLOSED_HOOK_90',
 'STIRRUP_TYPE_THREE_LEGGED_OVERLAP_HOOK_180',
 'STIRRUP_TYPE_TWO_LEGGED_CLOSED_HOOK_135',
 'STIRRUP_TYPE_TWO_LEGGED_CLOSED_HOOK_90', 'STIRRUP_TYPE_TWO_LEGGED_OPEN',
 'STIRRUP_TYPE_TWO_LEGGED_OVERLAP_HOOK_180' }
 ■ material - optional; type int
 ■ stirrup_count - optional; type int
 ■ stirrup_diameter - optional; type double
 ■ stirrup_bar_size_type - optional; type stirrup_bar_size_type - type undefined with restriction
 ■ stirrup_distances - optional; type double
 ■ reinforcement_area - optional; type double
 ■ span_position_reference_type - optional; type span_position_reference_type - type undefined with
 restriction - enum { 'SHEAR_REINFORCEMENT_SPAN_INTERNAL_NODE',
 'SHEAR_REINFORCEMENT_SPAN_REFERENCE_END',
 'SHEAR_REINFORCEMENT_SPAN_REFERENCE_START' }
 ■ span_position_reference_internal_node - optional; type int
 ■ span_position_reference_x_location_relative - optional; type double
 ■ span_position_reference_x_location_absolute - optional; type double
 ■ span_position_definition_format_type - optional; type span_position_definition_format_type - type
 undefined with restriction - enum {
 'SHEAR_REINFORCEMENT_SPAN_DEFINITION_FORMAT_ABSOLUTE',
 'SHEAR_REINFORCEMENT_SPAN_DEFINITION_FORMAT_RELATIVE' }
 ■ span_start_relative - optional; type double
 ■ span_start_absolute - optional; type double
 ■ span_end_relative - optional; type double
 ■ span_end_absolute - optional; type double
 ■ span_length - optional; type double
 ■ stirrup_layout_rule_type - optional; type stirrup_layout_rule_type - type undefined with restriction -
 enum { 'SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_END_DEFINED',
 'SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_END_EQUALS_REST_LENGTH_TO_STIRRUP_DISTA',
 'SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_DEFINED',
 'SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_EQUALS_END',
 'SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_EQUALS_REST_LENGTH_TO_STIRRUP_DIS'
 }
 ■ stirrup_start_offset - optional; type double
 ■ stirrup_end_offset - optional; type double
 ■ one_stirrup_length - optional; type double
 ■ minimal_and_maximal_stirrup_length - optional; type string
 ■ length - optional; type double
 ■ one_stirrup_weight - optional; type double
 ■ minimal_and_maximal_stirrup_weight - optional; type string

```

- weight - optional; type *double*
- concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_top\_side - optional; type *double*
    - additional\_offset\_bottom\_side - optional; type *double*
    - additional\_offset\_left\_side - optional; type *double*
    - additional\_offset\_right\_side - optional; type *double*
    - additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
    - additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }

- additional\_offset\_reference\_type\_at\_end\_type - optional; type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- concrete\_cover - optional; type *double*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- concrete\_cover\_left - optional; type *double*
- concrete\_cover\_right - optional; type *double*
- concrete\_cover\_min - optional; type *array\_of\_int*
- concrete\_cover\_min\_top - optional; type *array\_of\_int*
- concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- concrete\_cover\_min\_left - optional; type *array\_of\_int*
- concrete\_cover\_min\_right - optional; type *array\_of\_int*
- concrete\_durability - optional; type *int*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- concrete\_durability\_left - optional; type *int*
- concrete\_durability\_right - optional; type *int*
- concrete\_effective\_lengths - optional; type *int*
- member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_concrete\_design\_seismic\_configuration - optional; type *int*
- timber\_effective\_lengths - optional; type *int*
- service\_class\_timber\_design - optional; type *int*
- moisture\_class\_timber\_design - optional; type *int*
- service\_conditions\_timber\_design - optional; type *int*
- timber\_member\_shear\_panels - optional; type *int*
- timber\_member\_rotational\_restraints - optional; type *int*
- timber\_local\_section\_reductions - optional; type *int*
- member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_transverse\_stiffener - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- deflection\_check\_direction - optional; type *member\_set\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- deflection\_check\_displacement\_reference - optional; type *member\_set\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- design\_support\_on\_member\_set\_start - optional; type *int*
- design\_support\_on\_member\_set\_end - optional; type *int*

- design\_supports\_on\_internal\_nodes - optional; type [array\\_of\\_member\\_set\\_design\\_supports\\_on\\_internal\\_nodes](#)
  - member\_set\_design\_supports\_on\_internal\_nodes - optional, unbounded; type [member\\_set\\_design\\_supports\\_on\\_internal\\_nodes](#)
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- members - optional; type [array\\_of\\_int](#)
- deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- deflection\_segments\_y\_axis - optional; type [array\\_of\\_member\\_set\\_deflection\\_segments\\_y\\_axis](#)
  - member\_set\_deflection\_segments\_y\_axis - optional, unbounded; type [member\\_set\\_deflection\\_segments\\_y\\_axis](#)
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- deflection\_segments\_z\_axis - optional; type [array\\_of\\_member\\_set\\_deflection\\_segments\\_z\\_axis](#)
  - member\_set\_deflection\_segments\_z\_axis - optional, unbounded; type [member\\_set\\_deflection\\_segments\\_z\\_axis](#)
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_member\_set\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_member\\_setResponse](#)

## 182. set\_member\_set\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_set\_imperfection

**Input:** set\_member\_set\_imperfection\_request (soap:body, use = literal) [Source code](#)

parameters type [set\\_member\\_set\\_imperfection](#)

- imperfection\_case\_no type *int*
- value type [member\\_set\\_imperfection](#)
  - no type *int*
  - imperfection\_type - optional; type [member\\_set\\_imperfection\\_imperfection\\_type](#) - type *undefined* with restriction - enum { 'IMPERFECTION\_TYPE\_INITIAL\_BOW', 'IMPERFECTION\_TYPE\_INITIAL\_BOW\_AND\_CRITERION', 'IMPERFECTION\_TYPE\_INITIAL\_SWAY' }
  - member\_sets - optional; type [array\\_of\\_int](#)
  - imperfection\_case - optional; type *int*
  - definition\_type - optional; type [member\\_set\\_imperfection\\_definition\\_type](#) - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_ABSOLUTE', 'DEFINITION\_TYPE\_ANSI\_CURRENT', 'DEFINITION\_TYPE\_ANSI\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_CSA\_CURRENT', 'DEFINITION\_TYPE\_CSA\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_EN\_1992\_1\_1993\_1', 'DEFINITION\_TYPE\_EN\_1993\_1\_1', 'DEFINITION\_TYPE\_EN\_1995\_1\_1', 'DEFINITION\_TYPE\_EN\_1999\_1\_1', 'DEFINITION\_TYPE\_GB\_50017\_2017', 'DEFINITION\_TYPE\_GB\_50017\_2017\_CURRENT', 'DEFINITION\_TYPE\_GB\_50017\_2017\_GRAVITY\_LOAD', 'DEFINITION\_TYPE\_NOTIONAL\_LOAD', 'DEFINITION\_TYPE\_RELATIVE' }
  - coordinate\_system - optional; type *string*
  - imperfection\_direction - optional; type [member\\_set\\_imperfection\\_imperfection\\_direction](#) - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'IMPERFECTION\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y', 'IMPERFECTION\_DIRECTION\_LOCAL\_Y\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_U\_NEGATIVE', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V', 'IMPERFECTION\_DIRECTION\_PRINCIPAL\_V\_NEGATIVE' }
  - basic\_value\_absolute - optional; type *double*
  - basic\_value\_relative - optional; type *double*
  - basic\_value\_coefficient - optional; type *double*
  - basic\_value\_force - optional; type *double*
  - section\_design - optional; type [member\\_set\\_imperfection\\_section\\_design](#) - type *undefined* with restriction - enum { 'SECTION\_DESIGN\_ELASTIC', 'SECTION\_DESIGN\_PLASTIC' }
  - active\_criterion - optional; type [member\\_set\\_imperfection\\_active\\_criterion](#) - type *undefined* with restriction - enum { 'ACTIVITY\_CRITERION\_ALWAYS', 'ACTIVITY\_CRITERION\_DEFINE', 'ACTIVITY\_CRITERION\_DIN\_18800', 'ACTIVITY\_CRITERION\_EN\_1993', 'ACTIVITY\_CRITERION\_EN\_1999' }
  - active\_bow - optional; type *double*
  - column\_in\_row - optional; type *int*
  - number\_of\_floors - optional; type *int*
  - standard\_factor\_enumeration - optional; type [member\\_set\\_imperfection\\_standard\\_factor\\_enumeration](#) - type *undefined* with restriction - enum { 'STANDARD\_FACTOR\_ASD', 'STANDARD\_FACTOR\_LRFD' }
  - standard\_factor\_number - optional; type *double*
  - height - optional; type *double*
  - case\_object - optional; type *int*
  - reduction\_factor\_h - optional; type *double*
  - reduction\_factor\_m - optional; type *double*
  - initial\_sway - optional; type *double*
  - initial\_sway\_inverted - optional; type *double*
  - delta - optional; type *double*
  - parameters - optional; type [array\\_of\\_int](#)
  - refer\_distance\_from\_objects\_to\_assign - optional; type *boolean*
  - imperfection\_over\_total\_length\_of\_objects\_to\_assign - optional; type *boolean*
  - distance\_a\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_b\_is\_defined\_as\_relative - optional; type *boolean*
  - distance\_a\_relative - optional; type *double*
  - distance\_b\_relative - optional; type *double*
  - distance\_a\_absolute - optional; type *double*
  - distance\_b\_absolute - optional; type *double*
  - comment - optional; type *string*



- `is_generated` - optional; type *boolean*
- `generating_object_info` - optional; type *string*
- `id_for_export_import` - optional; type *string*
- `metadata_for_export_import` - optional; type *string*

Output: `set_member_set_imperfection_response` (soap:body, use = literal) [Source code](#)

parameters type `set_member_set_imperfectionResponse`

### 183. `set_member_set_load`

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: `http://localhost:8082/set_member_set_load`

Input: `set_member_set_load_request` (soap:body, use = literal) [Source code](#)

parameters type `set_member_set_load`

- `load_case_no` type *int*
- value type `member_set_load`
  - no type *int*
  - `load_type` - optional; type `member_set_load_load_type` - type *undefined* with restriction - enum { 'E\_TYPE\_MASS', 'LOAD\_TYPE\_AXIAL\_DISPLACEMENT', 'LOAD\_TYPE\_AXIAL\_STRAIN', 'LOAD\_TYPE\_DISPLACEMENT', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_FORM\_FINDING', 'LOAD\_TYPE\_INITIAL\_PRESTRESS', 'LOAD\_TYPE\_MOMENT', 'LOAD\_TYPE\_PIPE\_CONTENT\_FULL', 'LOAD\_TYPE\_PIPE\_CONTENT\_PARTIAL', 'LOAD\_TYPE\_PIPE\_INTERNAL\_PRESSURE', 'LOAD\_TYPE\_PRECAMBER', 'LOAD\_TYPE\_ROTARY\_MOTION', 'LOAD\_TYPE\_ROTATION', 'LOAD\_TYPE\_TEMPERATURE', 'LOAD\_TYPE\_TEMPERATURE\_CHANGE' }
  - `member_sets` - optional; type *array\_of\_int*
  - `load_case` - optional; type *int*
  - `coordinate_system` - optional; type *string*
  - `load_distribution` - optional; type `member_set_load_load_distribution` - type *undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_CONCENTRATED\_1', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_2x2', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_N', 'LOAD\_DISTRIBUTION\_CONCENTRATED\_VARYING', 'LOAD\_DISTRIBUTION\_PARABOLIC', 'LOAD\_DISTRIBUTION\_TAPERED', 'LOAD\_DISTRIBUTION\_TRAPEZOIDAL', 'LOAD\_DISTRIBUTION\_UNIFORM', 'LOAD\_DISTRIBUTION\_UNIFORM\_TOTAL', 'LOAD\_DISTRIBUTION\_VARYING', 'LOAD\_DISTRIBUTION\_VARYING\_IN\_Z' }
  - `load_direction` - optional; type `member_set_load_load_direction` - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z', 'LOAD\_DIRECTION\_PRINCIPAL\_U', 'LOAD\_DIRECTION\_PRINCIPAL\_V' }
  - `load_direction_orientation` - optional; type `member_set_load_load_direction_orientation` - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
  - `form_finding_definition_type` - optional; type `member_set_load_form_finding_definition_type` - type *undefined* with restriction - enum { 'FORM\_FINDING\_TYPE\_FORCE', 'FORM\_FINDING\_TYPE\_GEOMETRIC' }
  - `magnitude` - optional; type *double*
  - `magnitude_1` - optional; type *double*
  - `magnitude_2` - optional; type *double*
  - `magnitude_3` - optional; type *double*
  - `magnitude_t_c` - optional; type *double*
  - `magnitude_t_c_1` - optional; type *double*
  - `magnitude_t_c_2` - optional; type *double*
  - `magnitude_t_c_3` - optional; type *double*
  - `magnitude_delta_t` - optional; type *double*
  - `magnitude_delta_t_1` - optional; type *double*
  - `magnitude_delta_t_2` - optional; type *double*
  - `magnitude_delta_t_3` - optional; type *double*
  - `magnitude_t_t` - optional; type *double*
  - `magnitude_t_t_1` - optional; type *double*
  - `magnitude_t_t_2` - optional; type *double*
  - `magnitude_t_t_3` - optional; type *double*
  - `magnitude_t_b` - optional; type *double*
  - `magnitude_t_b_1` - optional; type *double*
  - `magnitude_t_b_2` - optional; type *double*
  - `magnitude_t_b_3` - optional; type *double*
  - `mass_global` - optional; type *double*
  - `mass_x` - optional; type *double*
  - `mass_y` - optional; type *double*
  - `mass_z` - optional; type *double*
  - `distance_a_is_defined_as_relative` - optional; type *boolean*
  - `distance_a_absolute` - optional; type *double*
  - `distance_a_relative` - optional; type *double*
  - `distance_b_is_defined_as_relative` - optional; type *boolean*
  - `distance_b_absolute` - optional; type *double*
  - `distance_b_relative` - optional; type *double*
  - `distance_c_is_defined_as_relative` - optional; type *boolean*
  - `distance_c_absolute` - optional; type *double*
  - `distance_c_relative` - optional; type *double*
  - `count_n` - optional; type *int*
  - `varying_load_parameters_are_defined_as_relative` - optional; type *boolean*
  - `varying_load_parameters` - optional; type *array\_of\_member\_set\_load\_varying\_load\_parameters*
    - `member_set_load_varying_load_parameters` - optional, unbounded; type `member_set_load_varying_load_parameters`
      - no - optional; type *int*
      - `distance` - optional; type *double*
      - `delta_distance` - optional; type *double*
      - `magnitude` - optional; type *double*
      - `note` - optional; type *string*
      - `magnitude_t_c` - optional; type *double*
      - `magnitude_delta_t` - optional; type *double*
      - `magnitude_t_t` - optional; type *double*
      - `magnitude_t_b` - optional; type *double*

- `varying_load_parameters_sorted` - optional; type `boolean`
- `angular_velocity` - optional; type `double`
- `angular_acceleration` - optional; type `double`
- `axis_definition_type` - optional; type `member_set_load_axis_definition_type` - type `undefined` with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- `axis_definition_p1` - optional; type `vector_3d`
  - `x` type `double`
  - `y` type `double`
  - `z` type `double`
- `axis_definition_p1_x` - optional; type `double`
- `axis_definition_p1_y` - optional; type `double`
- `axis_definition_p1_z` - optional; type `double`
- `axis_definition_p2` - optional; type `vector_3d`
  - `x` type `double`
  - `y` type `double`
  - `z` type `double`
- `axis_definition_p2_x` - optional; type `double`
- `axis_definition_p2_y` - optional; type `double`
- `axis_definition_p2_z` - optional; type `double`
- `axis_definition_axis` - optional; type `member_set_load_axis_definition_axis` - type `undefined` with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- `axis_definition_axis_orientation` - optional; type `member_set_load_axis_definition_axis_orientation` - type `undefined` with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- `filling_height` - optional; type `double`
- `distance_from_member_set_end` - optional; type `boolean`
- `load_is_over_total_length` - optional; type `boolean`
- `has_force_eccentricity` - optional; type `boolean`
- `eccentricity_horizontal_alignment` - optional; type `member_set_load_eccentricity_horizontal_alignment` - type `undefined` with restriction - enum { 'ALIGN\_LEFT', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_RIGHT' }
- `eccentricity_vertical_alignment` - optional; type `member_set_load_eccentricity_vertical_alignment` - type `undefined` with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_NONE', 'ALIGN\_TOP' }
- `eccentricity_section_middle` - optional; type `member_set_load_eccentricity_section_middle` - type `undefined` with restriction - enum { 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_CENTER\_OF\_GRAVITY', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_NONE', 'LOAD\_ECCENTRICITY\_SECTION\_MIDDLE\_SHEAR\_CENTER' }
- `is_eccentricity_at_end_different_from_start` - optional; type `boolean`
- `eccentricity_y_at_start` - optional; type `double`
- `eccentricity_z_at_start` - optional; type `double`
- `eccentricity_y_at_end` - optional; type `double`
- `eccentricity_z_at_end` - optional; type `double`
- `form_finding_internal_force` - optional; type `member_set_load_form_finding_internal_force` - type `undefined` with restriction - enum { 'FORM\_FINDING\_INTERNAL\_FORCE\_COMPRESSION', 'FORM\_FINDING\_INTERNAL\_FORCE\_TENSION' }
- `form_finding_geometry_definition` - optional; type `member_set_load_form_finding_geometry_definition` - type `undefined` with restriction - enum { 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LENGTH', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_LOW\_POINT\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_MAXIMUM\_VERTICAL\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_SAG', 'FORM\_FINDING\_GEOMETRIC\_INPUT\_PARAMETER\_UNSTRESSED\_LENGTH' }
- `form_finding_force_definition` - optional; type `member_set_load_form_finding_force_definition` - type `undefined` with restriction - enum { 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_AVERAGE', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_DENSITY', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_HORIZONTAL\_TENSION\_COMPONENT', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MAXIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMAL\_TENSION\_AT\_J\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_MINIMUM\_FORCE\_IN\_MEMBER', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_I\_END', 'FORM\_FINDING\_FORCE\_INPUT\_PARAMETER\_TENSION\_AT\_J\_END' }
- `form_finding_magnitude_is_defined_as_relative` - optional; type `boolean`
- `form_finding_magnitude_absolute` - optional; type `double`
- `form_finding_magnitude_relative` - optional; type `double`
- `individual_mass_components` - optional; type `boolean`
- `comment` - optional; type `string`
- `is_generated` - optional; type `boolean`
- `generating_object_info` - optional; type `string`
- `id_for_export_import` - optional; type `string`
- `metadata_for_export_import` - optional; type `string`

**Output:** `set_member_set_load_response` (soap:body, use = literal) [Source code](#)

parameters type `set_member_set_loadResponse`

#### 184. `set_member_set_representative`

[Source code](#)

**Operation type:** `Request-response`. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_member_set_representative`

**Input:** `set_member_set_representative_request` (soap:body, use = literal) [Source code](#)

- parameters type `set_member_set_representative`
- value type `member_set_representative`
    - no type `int`
    - `user_defined_name_enabled` - optional; type `boolean`
    - `name` - optional; type `string`
    - `member_sets` - optional; type `array_of_int`
    - `comment` - optional; type `string`
    - `number_of_member_sets` - optional; type `int`
    - `number_of_members` - optional; type `int`
    - `section_ids` - optional; type `string`
    - `total_length` - optional; type `double`
    - `total_volume` - optional; type `double`
    - `total_mass` - optional; type `double`
    - `total_surface_of_coating` - optional; type `double`
    - `total_center_of_gravity_x` - optional; type `double`
    - `total_center_of_gravity_y` - optional; type `double`
    - `total_center_of_gravity_z` - optional; type `double`

- member\_model\_no - optional; type *int*
- member\_model\_type - optional; type *member\_set\_representative\_member\_model\_type* - type *undefined* with restriction - enum { 'TYPE\_BEAM', 'TYPE\_BUCKLING', 'TYPE\_CABLE', 'TYPE\_COMPRESSION', 'TYPE\_COUPLING\_HINGE\_HINGE', 'TYPE\_COUPLING\_HINGE\_RIGID', 'TYPE\_COUPLING\_RIGID\_HINGE', 'TYPE\_COUPLING\_RIGID\_RIGID', 'TYPE\_DEFINABLE\_STIFFNESS', 'TYPE\_RESULT\_BEAM', 'TYPE\_RIB', 'TYPE\_RIGID', 'TYPE\_TENSION', 'TYPE\_TRUSS', 'TYPE\_TRUSS\_ONLY\_N' }
- member\_model\_is\_deactivated\_for\_calculation - optional; type *boolean*
- member\_model\_line - optional; type *int*
- member\_model\_section\_distribution\_type - optional; type *member\_set\_representative\_member\_model\_section\_distribution\_type* - type *undefined* with restriction - enum { 'SECTION\_DISTRIBUTION\_TYPE\_LINEAR', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_OFFSET\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_SADDLE', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_BOTH\_SIDES', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_END\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_TAPERED\_AT\_START\_OF\_MEMBER', 'SECTION\_DISTRIBUTION\_TYPE\_UNIFORM' }
- member\_model\_reference\_type - optional; type *member\_set\_representative\_member\_model\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
- member\_model\_nodes - optional; type *array\_of\_int*
- member\_model\_node\_start - optional; type *int*
- member\_model\_node\_end - optional; type *int*
- member\_model\_analytical\_length - optional; type *double*
- member\_model\_analytical\_volume - optional; type *double*
- member\_model\_analytical\_mass - optional; type *double*
- member\_model\_analytical\_surface\_of\_coating - optional; type *double*
- member\_model\_analytical\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- member\_model\_analytical\_center\_of\_gravity\_x - optional; type *double*
- member\_model\_analytical\_center\_of\_gravity\_y - optional; type *double*
- member\_model\_analytical\_center\_of\_gravity\_z - optional; type *double*
- member\_model\_length - optional; type *double*
- member\_model\_volume - optional; type *double*
- member\_model\_mass - optional; type *double*
- member\_model\_surface\_of\_coating - optional; type *double*
- member\_model\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- member\_model\_center\_of\_gravity\_x - optional; type *double*
- member\_model\_center\_of\_gravity\_y - optional; type *double*
- member\_model\_center\_of\_gravity\_z - optional; type *double*
- member\_set\_model\_position - optional; type *string*
- member\_set\_model\_position\_short - optional; type *string*
- member\_model\_member\_representative - optional; type *int*
- member\_model\_design\_properties\_via\_member - optional; type *boolean*
- member\_model\_design\_properties\_via\_parent\_member\_set - optional; type *boolean*
- member\_model\_design\_properties\_parent\_member\_set - optional; type *int*
- member\_model\_comment - optional; type *string*
- member\_model\_member\_type\_rib\_alignment - optional; type *member\_set\_representative\_member\_model\_member\_type\_rib\_alignment* - type *undefined* with restriction - enum { 'ALIGNMENT\_CENTRIC', 'ALIGNMENT\_ON\_Z\_SIDE\_NEGATIVE', 'ALIGNMENT\_ON\_Z\_SIDE\_POSITIVE', 'ALIGNMENT\_USER\_DEFINED\_VIA\_MEMBER\_ECCENTRICITY' }
- member\_model\_member\_rib\_first\_surface - optional; type *int*
- member\_model\_member\_rib\_second\_surface - optional; type *int*
- member\_model\_member\_rib\_surface\_assignment\_autodetect - optional; type *boolean*
- member\_model\_member\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_model\_align\_local\_z\_axis\_to\_local\_z\_axis\_of\_surface - optional; type *boolean*
- member\_model\_flange\_dimensions - optional; type *array\_of\_member\_set\_representative\_member\_model\_flange\_dimensions*
  - member\_set\_representative\_member\_model\_flange\_dimensions - optional, unbounded; type *member\_set\_representative\_member\_model\_flange\_dimensions*
    - no - optional; type *int*
    - end\_ordinate - optional; type *double*
    - length - optional; type *double*
    - reference\_length\_definition\_type - optional; type *reference\_length\_definition\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_TYPE\_MEMBER\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_SEGMENT\_LENGTH', 'REFERENCE\_LENGTH\_TYPE\_USER\_DEFINED' }
    - reference\_length - optional; type *double*
    - reference\_length\_width\_type - optional; type *reference\_length\_width\_type* - type *undefined* with restriction - enum { 'REFERENCE\_LENGTH\_WIDTH\_EC2', 'REFERENCE\_LENGTH\_WIDTH\_EIGHTH', 'REFERENCE\_LENGTH\_WIDTH\_NONE', 'REFERENCE\_LENGTH\_WIDTH\_SIXTH' }
    - width\_minus\_y\_integrative - optional; type *double*
    - width\_minus\_y\_effective - optional; type *double*
    - width\_minus\_y\_maximal - optional; type *double*
    - width\_plus\_y\_integrative - optional; type *double*
    - width\_plus\_y\_effective - optional; type *double*
    - width\_plus\_y\_maximal - optional; type *double*
    - distribution\_linearity\_type - optional; type *distribution\_linearity\_type* - type *undefined* with restriction
- member\_model\_synchronize\_width\_mode - optional; type *boolean*
- member\_model\_relative\_ordinates\_mode - optional; type *boolean*
- member\_model\_member\_type\_definable\_stiffness - optional; type *int*
- member\_model\_result\_beam\_integrate\_stresses\_and\_forces - optional; type *member\_set\_representative\_member\_model\_result\_beam\_integrate\_stresses\_and\_forces* - type *undefined* with restriction - enum { 'INTEGRATE\_FROM\_LISTED\_OBJECT', 'INTEGRATE\_WITHIN\_CUBOID\_GENERAL', 'INTEGRATE\_WITHIN\_CUBOID\_QUADRATIC', 'INTEGRATE\_WITHIN\_CYLINDER' }
- member\_model\_result\_beam\_y\_z - optional; type *double*
- member\_model\_result\_beam\_y\_plus - optional; type *double*
- member\_model\_result\_beam\_z\_plus - optional; type *double*
- member\_model\_result\_beam\_y\_minus - optional; type *double*
- member\_model\_result\_beam\_z\_minus - optional; type *double*

- member\_model\_result\_beam\_radius - optional; type *double*
- member\_model\_result\_beam\_include\_surfaces - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_surfaces - optional; type *boolean*
- member\_model\_result\_beam\_include\_solids - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_solids - optional; type *boolean*
- member\_model\_result\_beam\_include\_members - optional; type *array\_of\_int*
- member\_model\_result\_beam\_include\_all\_members - optional; type *boolean*
- member\_model\_result\_beam\_exclude\_surfaces - optional; type *array\_of\_int*
- member\_model\_result\_beam\_exclude\_solids - optional; type *array\_of\_int*
- member\_model\_result\_beam\_exclude\_members - optional; type *array\_of\_int*
- member\_model\_projected\_length - optional; type *double*
- member\_model\_section\_distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- member\_model\_section\_distance\_from\_start\_absolute - optional; type *double*
- member\_model\_section\_distance\_from\_end\_relative - optional; type *double*
- member\_model\_section\_distance\_from\_end\_is\_defined\_as\_relative - optional; type *boolean*
- member\_model\_section\_distance\_from\_end\_absolute - optional; type *double*
- member\_model\_section\_distance\_from\_start\_relative - optional; type *double*
- member\_model\_section\_alignment - optional; type *member\_set\_representative\_member\_model\_section\_alignment* - type *undefined* with restriction - enum { 'SECTION\_ALIGNMENT\_BOTTOM', 'SECTION\_ALIGNMENT\_CENTRIC', 'SECTION\_ALIGNMENT\_TOP' }
- member\_model\_curved\_member\_is\_asymmetric\_layout - optional; type *boolean*
- member\_model\_curved\_member\_is\_cantilevers - optional; type *boolean*
- member\_model\_curved\_member\_cantilevers\_type - optional; type *member\_set\_representative\_member\_model\_curved\_member\_cantilevers\_type* - type *undefined* with restriction - enum { 'CANTILEVERS\_TYPE\_HORIZONTAL', 'CANTILEVERS\_TYPE\_OFFSET', 'CANTILEVERS\_TYPE\_PARALLEL', 'CANTILEVERS\_TYPE\_TAPER' }
- member\_model\_rotation\_specification\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_ANGLE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_HELP\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_INSIDE\_NODE', 'COORDINATE\_SYSTEM\_ROTATION\_VIA\_SURFACE' }
- member\_model\_rotation\_angle - optional; type *double*
- member\_model\_rotation\_help\_node - optional; type *int*
- member\_model\_rotation\_plane\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- member\_model\_rotation\_surface - optional; type *int*
- member\_model\_rotation\_surface\_plane\_type - optional; type *member\_set\_representative\_member\_model\_rotation\_surface\_plane\_type* - type *undefined* with restriction - enum { 'ROTATION\_PLANE\_XY', 'ROTATION\_PLANE\_XZ' }
- member\_model\_is\_rotated - optional; type *boolean*
- member\_model\_section\_start - optional; type *int*
- member\_model\_section\_end - optional; type *int*
- member\_model\_section\_internal - optional; type *int*
- member\_model\_section\_material - optional; type *int*
- member\_model\_member\_hinge\_start - optional; type *int*
- member\_model\_member\_hinge\_end - optional; type *int*
- member\_model\_member\_eccentricity\_start - optional; type *int*
- member\_model\_member\_eccentricity\_end - optional; type *int*
- member\_model\_support - optional; type *int*
- member\_model\_member\_transverse\_stiffener - optional; type *int*
- member\_model\_member\_nonlinearity - optional; type *int*
- member\_model\_member\_result\_intermediate\_point - optional; type *int*
- member\_model\_concrete\_effective\_lengths - optional; type *int*
- member\_model\_steel\_effective\_lengths - optional; type *int*
- member\_model\_timber\_effective\_lengths - optional; type *int*
- member\_model\_aluminum\_effective\_lengths - optional; type *int*
- member\_model\_concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans*
  - member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_set\_representative\_member\_model\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*
    - stirrup\_diameter - optional; type *double*
    - stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
    - stirrup\_distances - optional; type *double*
    - reinforcement\_area - optional; type *double*
    - span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTA

'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED',  
'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END',  
'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DIS'  
}

- stirrup\_start\_offset - optional; type *double*
- stirrup\_end\_offset - optional; type *double*
- one\_stirrup\_length - optional; type *double*
- minimal\_and\_maximal\_stirrup\_length - optional; type *string*
- length - optional; type *double*
- one\_stirrup\_weight - optional; type *double*
- minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
- weight - optional; type *double*
- member\_model\_concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_representative\_member\_model\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*
    - additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
    - additional\_offset\_top\_side - optional; type *double*
    - additional\_offset\_bottom\_side - optional; type *double*
    - additional\_offset\_left\_side - optional; type *double*
    - additional\_offset\_right\_side - optional; type *double*
    - additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM',

- 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_end\_type - optional; type *additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- member\_model\_concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- member\_model\_concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- member\_model\_concrete\_cover - optional; type *double*
- member\_model\_concrete\_cover\_top - optional; type *double*
- member\_model\_concrete\_cover\_bottom - optional; type *double*
- member\_model\_concrete\_cover\_left - optional; type *double*
- member\_model\_concrete\_cover\_right - optional; type *double*
- member\_model\_concrete\_cover\_min - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_top - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_left - optional; type *array\_of\_int*
- member\_model\_concrete\_cover\_min\_right - optional; type *array\_of\_int*
- member\_model\_concrete\_durability - optional; type *int*
- member\_model\_concrete\_durability\_top - optional; type *int*
- member\_model\_concrete\_durability\_bottom - optional; type *int*
- member\_model\_concrete\_durability\_left - optional; type *int*
- member\_model\_concrete\_durability\_right - optional; type *int*
- member\_model\_steel\_boundary\_conditions - optional; type *int*
- member\_model\_steel\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_steel\_member\_transverse\_weld - optional; type *int*
- member\_model\_steel\_member\_shear\_panel - optional; type *int*
- member\_model\_steel\_member\_rotational\_restraint - optional; type *int*
- member\_model\_timber\_service\_class - optional; type *int*
- member\_model\_timber\_moisture\_class - optional; type *int*
- member\_model\_timber\_service\_conditions - optional; type *int*
- member\_model\_timber\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_timber\_member\_shear\_panel - optional; type *int*
- member\_model\_timber\_member\_rotational\_restraint - optional; type *int*
- member\_model\_aluminum\_boundary\_conditions - optional; type *int*
- member\_model\_aluminum\_member\_local\_section\_reduction - optional; type *int*
- member\_model\_aluminum\_member\_transverse\_weld - optional; type *int*

- member\_model\_aluminum\_member\_shear\_panel - optional; type *int*
- member\_model\_aluminum\_member\_rotational\_restraint - optional; type *int*
- member\_model\_stress\_analysis\_configuration - optional; type *int*
- member\_model\_member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_model\_member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_model\_member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_model\_end\_modifications\_member\_start\_extension - optional; type *double*
- member\_model\_end\_modifications\_member\_start\_slope\_y - optional; type *double*
- member\_model\_end\_modifications\_member\_start\_slope\_z - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_extension - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_slope\_y - optional; type *double*
- member\_model\_end\_modifications\_member\_end\_slope\_z - optional; type *double*
- member\_model\_has\_any\_end\_modifications - optional; type *boolean*
- member\_model\_deflection\_check\_direction - optional; type *member\_set\_representative\_member\_model\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- member\_model\_deflection\_check\_displacement\_reference - optional; type *member\_set\_representative\_member\_model\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- member\_model\_design\_support\_on\_member\_start - optional; type *int*
- member\_model\_design\_support\_on\_member\_end - optional; type *int*
- member\_model\_design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes*
  - member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_set\_representative\_member\_model\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- member\_model\_is\_generated - optional; type *boolean*
- member\_model\_generating\_object\_info - optional; type *string*
- member\_set\_model\_no - optional; type *int*
- member\_set\_model\_user\_defined\_name\_enabled - optional; type *boolean*
- member\_set\_model\_name - optional; type *string*
- member\_set\_model\_set\_type - optional; type *member\_set\_representative\_member\_set\_model\_set\_type* - type *undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
- member\_set\_model\_members - optional; type *array\_of\_int*
- member\_set\_model\_length - optional; type *double*
- member\_set\_model\_center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- member\_set\_model\_center\_of\_gravity\_x - optional; type *double*
- member\_set\_model\_center\_of\_gravity\_y - optional; type *double*
- member\_set\_model\_center\_of\_gravity\_z - optional; type *double*
- member\_set\_model\_volume - optional; type *double*
- member\_set\_model\_mass - optional; type *double*
- member\_set\_model\_member\_set\_representative - optional; type *int*
- member\_set\_model\_discontinuous\_torsional\_warping - optional; type *boolean*
- member\_set\_model\_design\_properties\_activated - optional; type *boolean*
- member\_set\_model\_steel\_effective\_lengths - optional; type *int*
- member\_set\_model\_steel\_boundary\_conditions - optional; type *int*
- member\_set\_model\_steel\_member\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_steel\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_steel\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_member\_steel\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_steel\_design\_seismic\_configuration - optional; type *int*
- member\_set\_model\_aluminum\_effective\_lengths - optional; type *int*
- member\_set\_model\_aluminum\_boundary\_conditions - optional; type *int*
- member\_set\_model\_aluminum\_member\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_aluminum\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_aluminum\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_aluminum\_member\_transverse\_weld - optional; type *int*
- member\_set\_model\_member\_aluminum\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_aluminum\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_concrete\_shear\_reinforcement\_spans - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans*
  - member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_concrete\_shear\_reinforcement\_spans*
    - no - optional; type *int*
    - name - optional; type *string*
    - stirrup\_type - optional; type *stirrup\_type* - type *undefined* with restriction - enum { 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_FOUR\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_THREE\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_THREE\_LEGGED\_OVERLAP\_HOOK\_180', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_135', 'STIRRUP\_TYPE\_TWO\_LEGGED\_CLOSED\_HOOK\_90', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OPEN', 'STIRRUP\_TYPE\_TWO\_LEGGED\_OVERLAP\_HOOK\_180' }
    - material - optional; type *int*
    - stirrup\_count - optional; type *int*

- stirrup\_diameter - optional; type *double*
- stirrup\_bar\_size\_type - optional; type *stirrup\_bar\_size\_type* - type *undefined* with restriction
- stirrup\_distances - optional; type *double*
- reinforcement\_area - optional; type *double*
- span\_position\_reference\_type - optional; type *span\_position\_reference\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_INTERNAL\_NODE', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_END', 'SHEAR\_REINFORCEMENT\_SPAN\_REFERENCE\_START' }
- span\_position\_reference\_internal\_node - optional; type *int*
- span\_position\_reference\_x\_location\_relative - optional; type *double*
- span\_position\_reference\_x\_location\_absolute - optional; type *double*
- span\_position\_definition\_format\_type - optional; type *span\_position\_definition\_format\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_ABSOLUTE', 'SHEAR\_REINFORCEMENT\_SPAN\_DEFINITION\_FORMAT\_RELATIVE' }
- span\_start\_relative - optional; type *double*
- span\_start\_absolute - optional; type *double*
- span\_end\_relative - optional; type *double*
- span\_end\_absolute - optional; type *double*
- span\_length - optional; type *double*
- stirrup\_layout\_rule\_type - optional; type *stirrup\_layout\_rule\_type* - type *undefined* with restriction - enum { 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_END\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTANCES', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_DEFINED', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_END', 'SHEAR\_REINFORCEMENT\_STIRRUP\_LAYOUT\_RULE\_START\_EQUALS\_REST\_LENGTH\_TO\_STIRRUP\_DISTANCES' }
- stirrup\_start\_offset - optional; type *double*
- stirrup\_end\_offset - optional; type *double*
- one\_stirrup\_length - optional; type *double*
- minimal\_and\_maximal\_stirrup\_length - optional; type *string*
- length - optional; type *double*
- one\_stirrup\_weight - optional; type *double*
- minimal\_and\_maximal\_stirrup\_weight - optional; type *string*
- weight - optional; type *double*
- member\_set\_model\_concrete\_longitudinal\_reinforcement\_items - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items*
  - member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_concrete\_longitudinal\_reinforcement\_items*
    - no - optional; type *int*
    - name - optional; type *string*
    - rebar\_type - optional; type *rebar\_type* - type *undefined* with restriction - enum { 'REBAR\_TYPE\_LINE', 'REBAR\_TYPE\_SINGLE', 'REBAR\_TYPE\_SYMMETRICAL', 'REBAR\_TYPE\_UNIFORMLY\_SURROUNDING', 'REBAR\_TYPE\_UNSYMMETRICAL' }
    - material - optional; type *int*
    - reinforcement\_placed\_in\_bending\_corner\_enabled - optional; type *boolean*
    - bar\_count\_symmetrical - optional; type *int*
    - bar\_count\_unsymmetrical\_at\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_top\_side - optional; type *int*
    - bar\_count\_unsymmetrical\_bottom\_side - optional; type *int*
    - bar\_count\_uniformly\_surrounding - optional; type *int*
    - bar\_count\_line - optional; type *int*
    - bar\_count\_corner - optional; type *int*
    - bar\_diameter\_symmetrical - optional; type *double*
    - bar\_diameter\_unsymmetrical\_at\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_top\_side - optional; type *double*
    - bar\_diameter\_unsymmetrical\_bottom\_side - optional; type *double*
    - bar\_diameter\_uniformly\_surrounding - optional; type *double*
    - bar\_diameter\_line - optional; type *double*
    - bar\_diameter\_single - optional; type *double*
    - bar\_diameter\_corner - optional; type *double*
    - bar\_size\_designation\_symmetrical\_type - optional; type *bar\_size\_designation\_symmetrical\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_at\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_at\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_top\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_top\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_unsymmetrical\_bottom\_side\_type - optional; type *bar\_size\_designation\_unsymmetrical\_bottom\_side\_type* - type *undefined* with restriction
    - bar\_size\_designation\_uniformly\_surrounding\_type - optional; type *bar\_size\_designation\_uniformly\_surrounding\_type* - type *undefined* with restriction
    - bar\_size\_designation\_line\_type - optional; type *bar\_size\_designation\_line\_type* - type *undefined* with restriction
    - bar\_size\_designation\_single\_type - optional; type *bar\_size\_designation\_single\_type* - type *undefined* with restriction
    - bar\_size\_designation\_corner\_type - optional; type *bar\_size\_designation\_corner\_type* - type *undefined* with restriction
    - corner\_reinforcement\_enabled - optional; type *boolean*
    - reinforcement\_area\_symmetrical - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_at\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_top\_side - optional; type *double*
    - reinforcement\_area\_unsymmetrical\_bottom\_side - optional; type *double*
    - reinforcement\_area\_uniformly\_surrounding - optional; type *double*
    - reinforcement\_area\_line - optional; type *double*
    - reinforcement\_area\_single - optional; type *double*
    - reinforcement\_area\_corner - optional; type *double*
    - reinforcement\_area\_total - optional; type *double*
    - span\_position\_reference\_internal\_node - optional; type *int*
    - span\_position\_reference\_x\_location\_relative - optional; type *double*
    - span\_position\_reference\_x\_location\_absolute - optional; type *double*
    - span\_start\_relative - optional; type *double*
    - span\_start\_absolute - optional; type *double*
    - span\_end\_relative - optional; type *double*
    - span\_end\_absolute - optional; type *double*
    - span\_length - optional; type *double*



- additional\_offset\_type - optional; type *additional\_offset\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- additional\_offset\_type\_single\_line\_type - optional; type *additional\_offset\_type\_single\_line\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_CONCRETE\_COVER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_SECTION\_SURFACE', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_FROM\_STIRRUP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_TYPE\_NONE' }
- additional\_offset\_top\_side - optional; type *double*
- additional\_offset\_bottom\_side - optional; type *double*
- additional\_offset\_left\_side - optional; type *double*
- additional\_offset\_right\_side - optional; type *double*
- additional\_offset\_reference\_type - optional; type *additional\_offset\_reference\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_start\_type - optional; type *additional\_offset\_reference\_type\_at\_start\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_offset\_reference\_type\_at\_end\_type - optional; type *additional\_offset\_reference\_type\_at\_end\_type* - type *undefined* with restriction - enum { 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_CENTER\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_LEFT\_TOP', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_BOTTOM', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_CENTER', 'LONGITUDINAL\_REINFORCEMENT\_ADDITIONAL\_OFFSET\_REFERENCE\_TYPE\_RIGHT\_TOP' }
- additional\_horizontal\_offset - optional; type *double*
- additional\_horizontal\_offset\_at\_start - optional; type *double*
- additional\_horizontal\_offset\_at\_end - optional; type *double*
- additional\_vertical\_offset - optional; type *double*
- additional\_vertical\_offset\_at\_start - optional; type *double*
- additional\_vertical\_offset\_at\_end - optional; type *double*
- anchorage\_start\_anchor\_type - optional; type *anchorage\_start\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_start\_anchor\_length - optional; type *double*
- anchorage\_start\_bending\_diameter - optional; type *double*
- anchorage\_end\_anchor\_type - optional; type *anchorage\_end\_anchor\_type* - type *undefined* with restriction - enum { 'ANCHORAGE\_TYPE\_BEND', 'ANCHORAGE\_TYPE\_HOOK', 'ANCHORAGE\_TYPE\_HOOK\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_NONE', 'ANCHORAGE\_TYPE\_STRAIGHT', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TRANSVERSE\_BAR', 'ANCHORAGE\_TYPE\_STRAIGHT\_WITH\_TWO\_TRANSVERSE\_BARS' }
- anchorage\_end\_anchor\_length - optional; type *double*
- anchorage\_end\_bending\_diameter - optional; type *double*
- one\_rebar\_length - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_length - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length - optional; type *string*
- one\_rebar\_corner\_length - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_length - optional; type *string*
- length - optional; type *double*
- one\_rebar\_weight - optional; type *double*
- one\_rebar\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_at\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_top\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_unsymmetrical\_bottom\_side\_weight - optional; type *double*
- one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight - optional; type *string*
- one\_rebar\_corner\_weight - optional; type *double*
- one\_rebar\_corner\_minimal\_and\_maximal\_weight - optional; type *string*
- weight - optional; type *double*
- member\_set\_model\_concrete\_cover\_user\_defined\_enabled - optional; type *boolean*
- member\_set\_model\_concrete\_cover\_different\_at\_section\_sides\_enabled - optional; type *boolean*
- member\_set\_model\_concrete\_cover - optional; type *double*
- member\_set\_model\_concrete\_cover\_top - optional; type *double*
- member\_set\_model\_concrete\_cover\_bottom - optional; type *double*
- member\_set\_model\_concrete\_cover\_left - optional; type *double*
- member\_set\_model\_concrete\_cover\_right - optional; type *double*
- member\_set\_model\_concrete\_cover\_min - optional; type *array\_of\_int*

- member\_set\_model\_concrete\_cover\_min\_top - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_bottom - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_left - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_cover\_min\_right - optional; type *array\_of\_int*
- member\_set\_model\_concrete\_durability - optional; type *int*
- member\_set\_model\_concrete\_durability\_top - optional; type *int*
- member\_set\_model\_concrete\_durability\_bottom - optional; type *int*
- member\_set\_model\_concrete\_durability\_left - optional; type *int*
- member\_set\_model\_concrete\_durability\_right - optional; type *int*
- member\_set\_model\_concrete\_effective\_lengths - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_concrete\_design\_seismic\_configuration - optional; type *int*
- member\_set\_model\_service\_class\_timber\_design - optional; type *int*
- member\_set\_model\_moisture\_class\_timber\_design - optional; type *int*
- member\_set\_model\_timber\_effective\_lengths - optional; type *int*
- member\_set\_model\_service\_conditions\_timber\_design - optional; type *int*
- member\_set\_model\_timber\_member\_shear\_panels - optional; type *int*
- member\_set\_model\_timber\_member\_rotational\_restraints - optional; type *int*
- member\_set\_model\_timber\_local\_section\_reductions - optional; type *int*
- member\_set\_model\_member\_timber\_design\_uls\_configuration - optional; type *int*
- member\_set\_model\_member\_timber\_design\_sls\_configuration - optional; type *int*
- member\_set\_model\_member\_timber\_design\_fr\_configuration - optional; type *int*
- member\_set\_model\_member\_transverse\_stiffener - optional; type *int*
- member\_set\_model\_stress\_analysis\_configuration - optional; type *int*
- member\_set\_model\_deflection\_check\_direction - optional; type *member\_set\_representative\_member\_set\_model\_deflection\_check\_direction* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z', 'DEFLECTION\_CHECK\_DIRECTION\_LOCAL\_AXIS\_Z\_AND\_Y', 'DEFLECTION\_CHECK\_DIRECTION\_RESULTING\_AXIS' }
- member\_set\_model\_deflection\_check\_displacement\_reference - optional; type *member\_set\_representative\_member\_set\_model\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_SEGMENT\_ENDS', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_UNDEFORMED\_SYSTEM' }
- member\_set\_model\_design\_support\_on\_member\_set\_start - optional; type *int*
- member\_set\_model\_design\_support\_on\_member\_set\_end - optional; type *int*
- member\_set\_model\_design\_supports\_on\_internal\_nodes - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes*
  - member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_design\_supports\_on\_internal\_nodes*
    - no - optional; type *int*
    - node - optional; type *int*
    - design\_support - optional; type *int*
- member\_set\_model\_comment - optional; type *string*
- member\_set\_model\_is\_generated - optional; type *boolean*
- member\_set\_model\_generating\_object\_info - optional; type *string*
- member\_set\_model\_member\_set\_rib\_generating\_longitudinal\_reinforcement\_items\_from\_surfaces\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- member\_model\_deflection\_segments\_y\_axis - optional; type *array\_of\_member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis*
  - member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_set\_representative\_member\_model\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_model\_deflection\_segments\_z\_axis - optional; type *array\_of\_member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis*
  - member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_set\_representative\_member\_model\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_set\_model\_deflection\_segments\_defined\_length\_y\_axis\_enabled - optional; type *boolean*
- member\_set\_model\_deflection\_segments\_defined\_length\_z\_axis\_enabled - optional; type *boolean*
- member\_set\_model\_deflection\_segments\_y\_axis - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis*
  - member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_deflection\_segments\_y\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- member\_set\_model\_deflection\_segments\_z\_axis - optional; type *array\_of\_member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis*
  - member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis - optional, unbounded; type *member\_set\_representative\_member\_set\_model\_deflection\_segments\_z\_axis*
    - no - optional; type *int*
    - active - optional; type *boolean*
    - length - optional; type *double*
    - precamber - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_member\_set\_representative\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_member\_set\_representativeResponse*

## 185. set\_member\_stiffness\_modification

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_stiffness\_modification

**Input:** set\_member\_stiffness\_modification\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_member_stiffness_modification
 value type member_stiffness_modification
 no type int
 type - optional; type member_stiffness_modification_type - type undefined with restriction - enum {
 'TYPE_CONCRETE_STRUCTURES_ACI', 'TYPE_CONCRETE_STRUCTURES_CSA',
 'TYPE_PARTIAL_STIFFNESSES_FACTORS', 'TYPE_STEEL_STRUCTURES', 'TYPE_STEEL_STRUCTURES_CSA',
 'TYPE_TOTAL_STIFFNESSES_FACTORS' }
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 factor_of_axial_stiffness - optional; type double
 factor_of_bending_y_stiffness - optional; type double
 factor_of_bending_z_stiffness - optional; type double
 partial_stiffness_factor_of_shear_y_stiffness - optional; type double
 partial_stiffness_factor_of_shear_z_stiffness - optional; type double
 partial_stiffness_factor_of_torsion_stiffness - optional; type double
 partial_stiffness_factor_of_weight - optional; type double
 total_stiffness_factor_of_total_stiffness - optional; type double
 steel_structure_csa_stiffness_factor_of_shear_y_stiffness - optional; type double
 steel_structure_csa_stiffness_factor_of_shear_z_stiffness - optional; type double
 steel_structure_csa_stiffness_factor_of_torsion_stiffness - optional; type double
 steel_structure_csa_factor_of_axial_stiffness_enable - optional; type boolean
 steel_structure_csa_factor_of_bending_y_stiffness_enable - optional; type boolean
 steel_structure_csa_factor_of_bending_z_stiffness_enable - optional; type boolean
 steel_structure_csa_factor_of_shear_y_stiffness_enable - optional; type boolean
 steel_structure_csa_factor_of_shear_z_stiffness_enable - optional; type boolean
 steel_structure_csa_stiffness_factor_of_torsion_stiffness_enable - optional; type boolean
 steel_structure_csa_determine_tau_b - optional; type
 member_stiffness_modification_steel_structure_csa_determine_tau_b - type undefined with restriction -
 enum { 'ITERATIVE', 'SET_TO_1' }
 steel_structure_gb_direct_method_enabled - optional; type boolean
 steel_structure_determine_tau_b - optional; type
 member_stiffness_modification_steel_structure_determine_tau_b - type undefined with restriction - enum {
 'ITERATIVE', 'SET_TO_1' }
 steel_structure_design_method - optional; type
 member_stiffness_modification_steel_structure_design_method - type undefined with restriction - enum {
 'ASD', 'LRFD' }
 concrete_structure_component_type - optional; type
 member_stiffness_modification_concrete_structure_component_type - type undefined with restriction -
 enum { 'COMPONENT_TYPE_BEAMS', 'COMPONENT_TYPE_COLUMNS',
 'COMPONENT_TYPE_FLAT_PLATES_AND_FLAT_SLABS', 'COMPONENT_TYPE_WALLS_CRACKED',
 'COMPONENT_TYPE_WALLS_UNCRACKED' }
 assigned_to_structure_modification - optional; type array_of_int
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_member\_stiffness\_modification\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_member_stiffness_modificationResponse
```

## 186. set\_member\_support

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_support

**Input:** set\_member\_support\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_member_support
 value type member_support
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 members - optional; type array_of_int
 spring_translation - optional; type vector_3d
 x type double
 y type double
 z type double
 spring_rotation - optional; type double
 spring_translation_x - optional; type double
 spring_translation_y - optional; type double
 spring_translation_z - optional; type double
 spring_shear - optional; type vector_3d
 x type double
 y type double
 z type double
 spring_shear_x - optional; type double
 spring_shear_y - optional; type double
 spring_shear_z - optional; type double
 nonlinearity - optional; type member_support_nonlinearity - type undefined with restriction - enum {
 'NONLINEARITY_FAILURE_IF_NEGATIVE_CONTACT_STRESS_Z',
 'NONLINEARITY_FAILURE_IF_POSITIVE_CONTACT_STRESS_Z', 'NONLINEARITY_NONE' }
 support_dimensions_enabled - optional; type boolean
 support_width_y - optional; type double
 support_width_z - optional; type double
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_member\_support\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_member\\_supportResponse](#)

### 187. set\_member\_transverse\_stiffener

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_member\_transverse\_stiffener

**Input:** set\_member\_transverse\_stiffener\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_member_transverse_stiffener
 ■ value type member_transverse_stiffener
 ■ no type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ members - optional; type array_of_int
 ■ member_sets - optional; type array_of_int
 ■ components - optional; type array_of_member_transverse_stiffener_components
 ■ member_transverse_stiffener_components - optional, unbounded; type member_transverse_stiffener_components
 ■ no - optional; type int
 ■ stiffener_type - optional; type stiffener_type - type undefined with restriction - enum {
 'STIFFENER_COMPONENT_TYPE_ANGLE', 'STIFFENER_COMPONENT_TYPE_CHANNEL_SECTION',
 'STIFFENER_COMPONENT_TYPE_CONNECTING_COLUMN_END',
 'STIFFENER_COMPONENT_TYPE_CONNECTING_COLUMN_START',
 'STIFFENER_COMPONENT_TYPE_END_PLATE_END',
 'STIFFENER_COMPONENT_TYPE_END_PLATE_START', 'STIFFENER_COMPONENT_TYPE_FLAT',
 'STIFFENER_COMPONENT_TYPE_WARPING_RESTRAINT' }
 ■ position - optional; type double
 ■ position_type - optional; type position_type - type undefined with restriction - enum {
 'STIFFENER_COMPONENT_POSITION_DOUBLE_SIDED',
 'STIFFENER_COMPONENT_POSITION_SINGLE_SIDED_LEFT',
 'STIFFENER_COMPONENT_POSITION_SINGLE_SIDED_RIGHT' }
 ■ multiple - optional; type boolean
 ■ note - optional; type string
 ■ multiple_number - optional; type int
 ■ multiple_offset_definition_type - optional; type multiple_offset_definition_type - type undefined with
 restriction - enum { 'OFFSET_DEFINITION_TYPE_ABSOLUTE', 'OFFSET_DEFINITION_TYPE_RELATIVE'
 }
 ■ multiple_offset - optional; type double
 ■ material - optional; type int
 ■ consider_stiffener - optional; type boolean
 ■ thickness - optional; type double
 ■ width - optional; type double
 ■ height - optional; type double
 ■ non_rigid - optional; type boolean
 ■ rigid - optional; type boolean
 ■ width_b_u - optional; type double
 ■ height_h_u - optional; type double
 ■ thickness_t_u - optional; type double
 ■ thickness_s_u - optional; type double
 ■ width_b - optional; type double
 ■ thickness_t - optional; type double
 ■ column_section - optional; type int
 ■ height_h_m - optional; type double
 ■ section - optional; type int
 ■ cantilever_l_k - optional; type double
 ■ full_warping_restraint - optional; type boolean
 ■ user_defined_restraint - optional; type boolean
 ■ user_defined_restraint_value - optional; type double
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

**Output:** set\_member\_transverse\_stiffener\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_member\\_transverse\\_stiffenerResponse](#)

### 188. set\_mesh\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_mesh\_settings

**Input:** set\_mesh\_settings\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_mesh_settings
 ■ value type meshConfig_type
 ■ general_target_length_of_fe type double
 ■ general_maximum_distance_between_node_and_line type double
 ■ general_maximum_number_of_mesh_nodes type int
 ■ members_number_of_divisions_for_special_types type int
 ■ members_activate_division_for_large_deformation_and_post_critical_analysis type boolean
 ■ members_number_of_divisions_for_result_diagram type int
 ■ members_number_of_divisions_for_min_max_values type int
 ■ members_use_division_for_concrete_members type boolean
 ■ members_number_of_divisions_for_concrete_members type int
 ■ members_use_division_for_straight_members type boolean
 ■ members_division_for_straight_members_type type boolean
 ■ members_length_of_fe type double
 ■ members_minimum_number_of_divisions type int
 ■ members_use_division_for_members_with_nodes type boolean
 ■ surfaces_maximum_ratio_of_fe type double
 ■ surfaces_maximum_out_of_plane_inclination type double
```

- surfaces\_mesh\_refinement *type boolean*
- surfaces\_relationship *type double*
- surfaces\_integrate\_also\_unutilized\_objects *type boolean*
- surfaces\_shape\_of\_finite\_elements *type meshConfig\_surfaces\_shape\_of\_finite\_elements\_type* - *type undefined* with restriction - enum { 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_QUADRANGLES\_ONLY', 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_TRIANGLES\_AND\_QUADRANGLES', 'E\_SHAPE\_OF\_FINITE\_ELEMENTS\_FOR\_SURFACES\_TRIANGLES\_ONLY' }
- surfaces\_same\_squares *type boolean*
- surfaces\_triangles\_for\_membranes *type boolean*
- surfaces\_mapped\_mesh\_preferred *type boolean*
- solids\_use\_refinement\_if\_containing\_close\_nodes *type boolean*
- solids\_maximum\_number\_of\_elements *type int*
- solids\_use\_target\_length\_of\_fe\_for\_type\_soil *type boolean*
- solids\_target\_length\_of\_fe\_for\_type\_soil *type double*
- SurfacesMeshQualityConfig *type meshConfig\_SurfacesMeshQualityConfig\_type*
  - mesh\_quality\_color\_indicator\_ok\_color *type color* - *type undefined* with restriction
  - mesh\_quality\_color\_indicator\_warning\_color *type color* - *type undefined* with restriction
  - mesh\_quality\_color\_indicator\_failure\_color *type color* - *type undefined* with restriction
  - QualityCriteriaConfig *type meshConfig\_SurfacesMeshQualityConfig\_QualityCriteriaConfig\_type*
    - quality\_criterion\_check\_aspect\_ratio *type boolean*
    - quality\_criterion\_check\_aspect\_ratio\_warning *type double*
    - quality\_criterion\_check\_aspect\_ratio\_failure *type double*
    - quality\_criterion\_parallel\_deviations *type boolean*
    - quality\_criterion\_parallel\_deviations\_warning *type double*
    - quality\_criterion\_parallel\_deviations\_failure *type double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements *type boolean*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_warning *type double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_failure *type double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements *type boolean*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_warning *type double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_failure *type double*
    - quality\_criterion\_warping\_of\_membrane\_elements *type boolean*
    - quality\_criterion\_warping\_of\_membrane\_elements\_warning *type double*
    - quality\_criterion\_warping\_of\_membrane\_elements\_failure *type double*
    - quality\_criterion\_warping\_of\_non\_membrane\_elements *type boolean*
    - quality\_criterion\_warping\_of\_non\_membrane\_elements\_warning *type double*
    - quality\_criterion\_warping\_of\_non\_membrane\_elements\_failure *type double*
    - quality\_criterion\_jacobian\_ratio *type boolean*
    - quality\_criterion\_jacobian\_ratio\_warning *type double*
    - quality\_criterion\_jacobian\_ratio\_failure *type double*
- SolidsMeshQualityConfig *type meshConfig\_SolidsMeshQualityConfig\_type*
  - mesh\_quality\_color\_indicator\_ok\_color *type color* - *type undefined* with restriction
  - mesh\_quality\_color\_indicator\_warning\_color *type color* - *type undefined* with restriction
  - mesh\_quality\_color\_indicator\_failure\_color *type color* - *type undefined* with restriction
  - QualityCriteriaConfig *type meshConfig\_SolidsMeshQualityConfig\_QualityCriteriaConfig\_type*
    - quality\_criterion\_check\_aspect\_ratio *type boolean*
    - quality\_criterion\_check\_aspect\_ratio\_warning *type double*
    - quality\_criterion\_check\_aspect\_ratio\_failure *type double*
    - quality\_criterion\_parallel\_deviations *type boolean*
    - quality\_criterion\_parallel\_deviations\_warning *type double*
    - quality\_criterion\_parallel\_deviations\_failure *type double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements *type boolean*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_warning *type double*
    - quality\_criterion\_corner\_angles\_of\_triangle\_elements\_failure *type double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements *type boolean*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_warning *type double*
    - quality\_criterion\_corner\_angles\_of\_quadrangle\_elements\_failure *type double*
    - quality\_criterion\_warping *type boolean*
    - quality\_criterion\_warping\_warning *type double*
    - quality\_criterion\_warping\_failure *type double*
    - quality\_criterion\_jacobian\_ratio *type boolean*
    - quality\_criterion\_jacobian\_ratio\_warning *type double*
    - quality\_criterion\_jacobian\_ratio\_failure *type double*
- windSimulationMeshConfig *type meshConfig\_windSimulationMeshConfig\_type*
  - windsimulation\_mesh\_config\_value\_simplify\_model *type boolean*
  - windsimulation\_mesh\_config\_value\_determine\_details\_by *type meshConfig\_windSimulationMeshConfig\_windsimulation\_mesh\_config\_value\_determine\_details\_by\_type* - *type undefined* with restriction - enum { 'E\_WINDSIMULATION\_DETERMINE\_DETAILS\_BY\_DETAIL\_SIZE', 'E\_WINDSIMULATION\_DETERMINE\_DETAILS\_BY\_LEVEL\_OF\_DETAILS' }
  - windsimulation\_mesh\_config\_value\_level\_of\_details *type int*
  - windsimulation\_mesh\_config\_value\_detail\_size *type double*
  - windsimulation\_mesh\_config\_value\_small\_openings\_closure\_type *type meshConfig\_windSimulationMeshConfig\_windsimulation\_mesh\_config\_value\_small\_openings\_closure\_type* - *type undefined* with restriction - enum { 'E\_WINDSIMULATION\_OPENINGS\_CLOSURE\_TYPE\_PERCENT\_OF\_MODEL\_DIAMETER', 'E\_WINDSIMULATION\_OPENINGS\_CLOSURE\_TYPE\_REAL\_SIZE' }
  - windsimulation\_mesh\_config\_value\_small\_openings\_closure\_value *type double*
  - windsimulation\_mesh\_config\_value\_optimized\_member\_topology *type boolean*
  - windsimulation\_mesh\_config\_value\_optimized\_member\_topo\_value *type int*
  - windsimulation\_mesh\_config\_value\_active\_objects\_only *type boolean*
  - windsimulation\_mesh\_config\_value\_terrain *type boolean*
  - windsimulation\_mesh\_config\_value\_terrain\_from\_model *type boolean*
  - windsimulation\_mesh\_config\_value\_terrain\_objects\_id *type string*
  - windsimulation\_mesh\_config\_value\_terrain\_objects\_all *type boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model *type boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_ifc\_objects\_id *type string*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_ifc\_objects\_all *type boolean*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_visual\_objects\_id *type string*
  - windsimulation\_mesh\_config\_value\_surrounding\_model\_visual\_objects\_all *type boolean*
  - windsimulation\_mesh\_config\_value\_keep\_results\_if\_mesh\_deleted *type boolean*
  - windsimulation\_mesh\_config\_value\_consider\_surface\_thickness *type boolean*
  - windsimulation\_mesh\_config\_value\_run\_rwind\_silent *type boolean*

Output: set\_mesh\_settings\_response (soap:body, use = literal) [Source code](#)

parameters *type set\_mesh\_settingsResponse*

## 189. set\_model\_type

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_model\_type

**Input:** set\_model\_type\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_model\_type*

- model\_type type *model\_type* - type *undefined* with restriction - enum { 'E\_MODEL\_TYPE\_1D\_X\_3D', 'E\_MODEL\_TYPE\_1D\_X\_AXIAL', 'E\_MODEL\_TYPE\_2D\_XY\_3D', 'E\_MODEL\_TYPE\_2D\_XY\_PLATE', 'E\_MODEL\_TYPE\_2D\_XZ\_3D', 'E\_MODEL\_TYPE\_2D\_XZ\_PLANE\_STRAIN', 'E\_MODEL\_TYPE\_2D\_XZ\_PLANE\_STRESS', 'E\_MODEL\_TYPE\_3D' }

**Output:** set\_model\_type\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_model\_typeResponse*

## 190. set\_nodal\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_nodal\_load

**Input:** set\_nodal\_load\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_nodal\_load*

- load\_case\_no type *int*
- value type *nodal\_load*
  - no type *int*
  - load\_type - optional; type *nodal\_load\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_COMPONENTS', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_MASS', 'LOAD\_TYPE\_MOMENT' }
  - nodes - optional; type *array\_of\_int*
  - force\_eccentricity - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - force\_eccentricity\_x - optional; type *double*
  - force\_eccentricity\_y - optional; type *double*
  - force\_eccentricity\_z - optional; type *double*
  - has\_force\_eccentricity - optional; type *boolean*
  - coordinate\_system - optional; type *int*
  - has\_specific\_direction - optional; type *boolean*
  - specific\_direction\_type - optional; type *nodal\_load\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
  - axes\_sequence - optional; type *nodal\_load\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - rotated\_about\_angle\_x - optional; type *double*
  - rotated\_about\_angle\_y - optional; type *double*
  - rotated\_about\_angle\_z - optional; type *double*
  - rotated\_about\_angle\_1 - optional; type *double*
  - rotated\_about\_angle\_2 - optional; type *double*
  - rotated\_about\_angle\_3 - optional; type *double*
  - directed\_to\_node\_direction\_node - optional; type *int*
  - parallel\_to\_two\_nodes\_first\_node - optional; type *int*
  - parallel\_to\_two\_nodes\_second\_node - optional; type *int*
  - parallel\_to\_line - optional; type *int*
  - parallel\_to\_member - optional; type *int*
  - components\_force - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - components\_force\_x - optional; type *double*
  - components\_force\_y - optional; type *double*
  - components\_force\_z - optional; type *double*
  - components\_moment - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - components\_moment\_x - optional; type *double*
  - components\_moment\_y - optional; type *double*
  - components\_moment\_z - optional; type *double*
  - force\_magnitude - optional; type *double*
  - load\_direction - optional; type *nodal\_load\_load\_direction* - type *undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W', 'LOAD\_DIRECTION\_LOCAL\_X', 'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }
  - moment\_magnitude - optional; type *double*
  - mass\_global - optional; type *double*
  - mass - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - mass\_x - optional; type *double*
  - mass\_y - optional; type *double*
  - mass\_z - optional; type *double*
  - individual\_mass\_components - optional; type *boolean*
  - mass\_moment\_of\_inertia - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - mass\_moment\_of\_inertia\_x - optional; type *double*
  - mass\_moment\_of\_inertia\_y - optional; type *double*

- mass\_moment\_of\_inertia\_z - optional; type *double*
- mass\_has\_rotational\_mass - optional; type *boolean*
- mass\_rotational\_mass - optional; type *double*
- mass\_angular\_velocity - optional; type *double*
- mass\_angular\_acceleration - optional; type *double*
- mass\_radius - optional; type *double*
- mass\_axis\_of\_rotation - optional; type *nodal\_load\_mass\_axis\_of\_rotation* - type *undefined* with restriction - enum { 'AXIS\_OF\_ROTATION\_X\_NEGATIVE', 'AXIS\_OF\_ROTATION\_X\_POSITIVE', 'AXIS\_OF\_ROTATION\_Y\_NEGATIVE', 'AXIS\_OF\_ROTATION\_Y\_POSITIVE', 'AXIS\_OF\_ROTATION\_Z\_NEGATIVE', 'AXIS\_OF\_ROTATION\_Z\_POSITIVE' }
- mass\_angle - optional; type *double*
- has\_shifted\_display - optional; type *boolean*
- offset - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- offset\_x - optional; type *double*
- offset\_y - optional; type *double*
- offset\_z - optional; type *double*
- size\_or\_distance - optional; type *double*
- comment - optional; type *string*
- load\_case - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_nodal\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_nodal\_loadResponse*

### 191. set\_nodal\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_nodal\_mesh\_refinement

**Input:** set\_nodal\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_nodal\_mesh\_refinement*
- value type *nodal\_mesh\_refinement*
    - no type *int*
    - type - optional; type *nodal\_mesh\_refinement\_type* - type *undefined* with restriction - enum { 'TYPE\_CIRCULAR', 'TYPE\_RECTANGULAR' }
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - comment - optional; type *string*
    - nodes - optional; type *array\_of\_int*
    - circular\_radius - optional; type *double*
    - circular\_target\_inner\_length - optional; type *double*
    - circular\_target\_outer\_length - optional; type *double*
    - circular\_length\_arrangement - optional; type *nodal\_mesh\_refinement\_circular\_length\_arrangement* - type *undefined* with restriction - enum { 'LENGTH\_ARRANGEMENT\_COMBINED', 'LENGTH\_ARRANGEMENT\_GRADUALLY', 'LENGTH\_ARRANGEMENT\_RADIAL' }
    - rectangular\_side - optional; type *double*
    - rectangular\_target\_inner\_length - optional; type *double*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - apply\_only\_on\_selected\_surfaces - optional; type *boolean*
    - selected\_surfaces - optional; type *array\_of\_int*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_nodal\_mesh\_refinement\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_nodal\_mesh\_refinementResponse*

### 192. set\_nodal\_support

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_nodal\_support

**Input:** set\_nodal\_support\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_nodal\_support*
- value type *nodal\_support*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - nodes - optional; type *array\_of\_int*
    - spring - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - spring\_x - optional; type *double*
    - spring\_y - optional; type *double*
    - spring\_z - optional; type *double*
    - rotational\_restraint - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - rotational\_restraint\_x - optional; type *double*
    - rotational\_restraint\_y - optional; type *double*
    - rotational\_restraint\_z - optional; type *double*
    - spring\_x\_nonlinearity - optional; type *nodal\_support\_spring\_x\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE',

'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }

- `spring_y_nonlinearity` - optional; type `nodal_support_spring_y_nonlinearity` - type `undefined` with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `spring_z_nonlinearity` - optional; type `nodal_support_spring_z_nonlinearity` - type `undefined` with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `rotational_restraint_x_nonlinearity` - optional; type `nodal_support_rotational_restraint_x_nonlinearity` - type `undefined` with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `rotational_restraint_y_nonlinearity` - optional; type `nodal_support_rotational_restraint_y_nonlinearity` - type `undefined` with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `rotational_restraint_z_nonlinearity` - optional; type `nodal_support_rotational_restraint_z_nonlinearity` - type `undefined` with restriction - enum { 'NONLINEARITY\_TYPE\_DIAGRAM', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_ALL\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_NEGATIVE', 'NONLINEARITY\_TYPE\_FAILURE\_IF\_POSITIVE', 'NONLINEARITY\_TYPE\_FORCE\_MOMENT\_DIAGRAM', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_1\_PLUS\_2', 'NONLINEARITY\_TYPE\_FRICTION\_DIRECTION\_2', 'NONLINEARITY\_TYPE\_NONE', 'NONLINEARITY\_TYPE\_PARTIAL\_ACTIVITY', 'NONLINEARITY\_TYPE\_STIFFNESS\_DIAGRAM' }
- `axes_sequence` - optional; type `nodal_support_axes_sequence` - type `undefined` with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
- `column_base_semi_rigid` - optional; type `double`
- `column_base_support_type` - optional; type `nodal_support_column_base_support_type` - type `undefined` with restriction - enum { 'BASE\_SUPPORT\_TYPE\_HINGED', 'BASE\_SUPPORT\_TYPE\_RIGID', 'BASE\_SUPPORT\_TYPE\_SEMI\_RIGID' }
- `column_head_support_type` - optional; type `nodal_support_column_head_support_type` - type `undefined` with restriction - enum { 'HEAD\_SUPPORT\_TYPE\_HINGED', 'HEAD\_SUPPORT\_TYPE\_SEMI\_RIGID' }
- `column_head_type` - optional; type `nodal_support_column_head_type` - type `undefined` with restriction - enum { 'HEAD\_TYPE\_CIRCULAR', 'HEAD\_TYPE\_RECTANGULAR' }
- `column_height` - optional; type `double`
- `column_material` - optional; type `int`
- `column_rotation` - optional; type `double`
- `column_rotational_restraint_x` - optional; type `double`
- `column_rotational_restraint_y` - optional; type `double`
- `column_section` - optional; type `int`
- `column_section_same_as_head` - optional; type `boolean`
- `column_shear_stiffness` - optional; type `boolean`
- `column_spring_x` - optional; type `double`
- `column_spring_y` - optional; type `double`
- `column_spring_z` - optional; type `double`
- `column_support_type` - optional; type `nodal_support_column_support_type` - type `undefined` with restriction - enum { 'SUPPORT\_TYPE\_ELASTIC\_NODAL\_SUPPORT', 'SUPPORT\_TYPE\_ELASTIC\_SURFACE\_FOUNDATIONS', 'SUPPORT\_TYPE\_WITH\_ADAPTED\_FE\_MESH' }
- `column_width_x` - optional; type `double`
- `column_width_y` - optional; type `double`
- `comment` - optional; type `string`
- `coordinate_system` - optional; type `int`
- `diagram_along_x_end` - optional; type `nodal_support_diagram_along_x_end` - type `undefined` with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_x_is_sorted` - optional; type `boolean`
- `diagram_along_x_start` - optional; type `nodal_support_diagram_along_x_start` - type `undefined` with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_x_symmetric` - optional; type `boolean`
- `diagram_along_x_table` - optional; type `array_of_nodal_support_diagram_along_x_table`
  - `nodal_support_diagram_along_x_table` - optional, unbounded; type `nodal_support_diagram_along_x_table`
    - `no` - optional; type `int`
    - `displacement` - optional; type `double`
    - `force` - optional; type `double`
    - `spring` - optional; type `double`
    - `note` - optional; type `string`
- `diagram_along_y_end` - optional; type `nodal_support_diagram_along_y_end` - type `undefined` with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_y_is_sorted` - optional; type `boolean`
- `diagram_along_y_start` - optional; type `nodal_support_diagram_along_y_start` - type `undefined` with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `diagram_along_y_symmetric` - optional; type `boolean`
- `diagram_along_y_table` - optional; type `array_of_nodal_support_diagram_along_y_table`



- nodal\_support\_diagram\_along\_y\_table - optional, unbounded; type *nodal\_support\_diagram\_along\_y\_table*
  - no - optional; type *int*
  - displacement - optional; type *double*
  - force - optional; type *double*
  - spring - optional; type *double*
  - note - optional; type *string*
- diagram\_along\_z\_end - optional; type *nodal\_support\_diagram\_along\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_z\_is\_sorted - optional; type *boolean*
- diagram\_along\_z\_start - optional; type *nodal\_support\_diagram\_along\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_along\_z\_symmetric - optional; type *boolean*
- diagram\_along\_z\_table - optional; type *array\_of\_nodal\_support\_diagram\_along\_z\_table*
  - nodal\_support\_diagram\_along\_z\_table - optional, unbounded; type *nodal\_support\_diagram\_along\_z\_table*
    - no - optional; type *int*
    - displacement - optional; type *double*
    - force - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_x\_end - optional; type *nodal\_support\_diagram\_around\_x\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_is\_sorted - optional; type *boolean*
- diagram\_around\_x\_start - optional; type *nodal\_support\_diagram\_around\_x\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_x\_symmetric - optional; type *boolean*
- diagram\_around\_x\_table - optional; type *array\_of\_nodal\_support\_diagram\_around\_x\_table*
  - nodal\_support\_diagram\_around\_x\_table - optional, unbounded; type *nodal\_support\_diagram\_around\_x\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_y\_end - optional; type *nodal\_support\_diagram\_around\_y\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_is\_sorted - optional; type *boolean*
- diagram\_around\_y\_start - optional; type *nodal\_support\_diagram\_around\_y\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_y\_symmetric - optional; type *boolean*
- diagram\_around\_y\_table - optional; type *array\_of\_nodal\_support\_diagram\_around\_y\_table*
  - nodal\_support\_diagram\_around\_y\_table - optional, unbounded; type *nodal\_support\_diagram\_around\_y\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- diagram\_around\_z\_end - optional; type *nodal\_support\_diagram\_around\_z\_end* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_is\_sorted - optional; type *boolean*
- diagram\_around\_z\_start - optional; type *nodal\_support\_diagram\_around\_z\_start* - type *undefined* with restriction - enum { 'DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'DIAGRAM\_ENDING\_TYPE\_STOP', 'DIAGRAM\_ENDING\_TYPE\_TEARING', 'DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- diagram\_around\_z\_symmetric - optional; type *boolean*
- diagram\_around\_z\_table - optional; type *array\_of\_nodal\_support\_diagram\_around\_z\_table*
  - nodal\_support\_diagram\_around\_z\_table - optional, unbounded; type *nodal\_support\_diagram\_around\_z\_table*
    - no - optional; type *int*
    - rotation - optional; type *double*
    - moment - optional; type *double*
    - spring - optional; type *double*
    - note - optional; type *string*
- directed\_to\_node\_direction\_node - optional; type *int*
- directed\_to\_node\_first\_axis - optional; type *nodal\_support\_directed\_to\_node\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- directed\_to\_node\_plane\_node - optional; type *int*
- directed\_to\_node\_second\_axis - optional; type *nodal\_support\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- fictitious\_column\_enabled - optional; type *boolean*
- friction\_coefficient\_x - optional; type *double*
- friction\_coefficient\_xy - optional; type *double*
- friction\_coefficient\_xz - optional; type *double*
- friction\_coefficient\_y - optional; type *double*
- friction\_coefficient\_yx - optional; type *double*
- friction\_coefficient\_yz - optional; type *double*
- friction\_coefficient\_z - optional; type *double*
- friction\_coefficient\_zx - optional; type *double*
- friction\_coefficient\_zy - optional; type *double*
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *nodal\_support\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_axis - optional; type *nodal\_support\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- partial\_activity\_along\_x\_negative\_displacement - optional; type *double*
- partial\_activity\_along\_x\_negative\_force - optional; type *double*
- partial\_activity\_along\_x\_negative\_slippage - optional; type *double*



- `stiffness_diagram_around_x_end` - optional; type `nodal_support_stiffness_diagram_around_x_end` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_x_is_sorted` - optional; type `boolean`
- `stiffness_diagram_around_x_start` - optional; type `nodal_support_stiffness_diagram_around_x_start` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_x_symmetric` - optional; type `boolean`
- `stiffness_diagram_around_x_table` - optional; type `array_of_nodal_support_stiffness_diagram_around_x_table`
  - `nodal_support_stiffness_diagram_around_x_table` - optional, unbounded; type `nodal_support_stiffness_diagram_around_x_table`
    - `no` - optional; type `int`
    - `force` - optional; type `double`
    - `spring` - optional; type `double`
    - `note` - optional; type `string`
- `stiffness_diagram_around_y_depends_on` - optional; type `nodal_support_stiffness_diagram_around_y_depends_on` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_P', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PX', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PY', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PZ' }
- `stiffness_diagram_around_y_end` - optional; type `nodal_support_stiffness_diagram_around_y_end` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_y_is_sorted` - optional; type `boolean`
- `stiffness_diagram_around_y_start` - optional; type `nodal_support_stiffness_diagram_around_y_start` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_y_symmetric` - optional; type `boolean`
- `stiffness_diagram_around_y_table` - optional; type `array_of_nodal_support_stiffness_diagram_around_y_table`
  - `nodal_support_stiffness_diagram_around_y_table` - optional, unbounded; type `nodal_support_stiffness_diagram_around_y_table`
    - `no` - optional; type `int`
    - `force` - optional; type `double`
    - `spring` - optional; type `double`
    - `note` - optional; type `string`
- `stiffness_diagram_around_z_depends_on` - optional; type `nodal_support_stiffness_diagram_around_z_depends_on` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_P', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PX', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PY', 'STIFFNESS\_DIAGRAM\_DEPENDS\_ON\_PZ' }
- `stiffness_diagram_around_z_end` - optional; type `nodal_support_stiffness_diagram_around_z_end` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_z_is_sorted` - optional; type `boolean`
- `stiffness_diagram_around_z_start` - optional; type `nodal_support_stiffness_diagram_around_z_start` - type `undefined` with restriction - enum { 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_CONTINUOUS', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_TEARING', 'STIFFNESS\_DIAGRAM\_ENDING\_TYPE\_YIELDING' }
- `stiffness_diagram_around_z_symmetric` - optional; type `boolean`
- `stiffness_diagram_around_z_table` - optional; type `array_of_nodal_support_stiffness_diagram_around_z_table`
  - `nodal_support_stiffness_diagram_around_z_table` - optional, unbounded; type `nodal_support_stiffness_diagram_around_z_table`
    - `no` - optional; type `int`
    - `force` - optional; type `double`
    - `spring` - optional; type `double`
    - `note` - optional; type `string`
- `support_dimension_diameter_x` - optional; type `double`
- `support_dimension_diameter_y` - optional; type `double`
- `support_dimension_diameter_z` - optional; type `double`
- `support_dimension_height_x` - optional; type `double`
- `support_dimension_height_y` - optional; type `double`
- `support_dimension_height_z` - optional; type `double`
- `support_dimension_type_on_x` - optional; type `nodal_support_support_dimension_type_on_x` - type `undefined` with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- `support_dimension_type_on_y` - optional; type `nodal_support_support_dimension_type_on_y` - type `undefined` with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- `support_dimension_type_on_z` - optional; type `nodal_support_support_dimension_type_on_z` - type `undefined` with restriction - enum { 'SUPPORT\_DIMENSION\_TYPE\_CIRCULAR', 'SUPPORT\_DIMENSION\_TYPE\_NONE', 'SUPPORT\_DIMENSION\_TYPE\_RECTANGULAR' }
- `support_dimension_width_x` - optional; type `double`
- `support_dimension_width_y` - optional; type `double`
- `support_dimension_width_z` - optional; type `double`
- `support_dimensions_enabled` - optional; type `boolean`
- `is_generated` - optional; type `boolean`
- `generating_object_info` - optional; type `string`
- `id_for_export_import` - optional; type `string`
- `metadata_for_export_import` - optional; type `string`

Output: `set_nodal_support_response` (soap:body, use = literal) [Source code](#)

parameters type `set_nodal_supportResponse`

### 193. `set_node`

[Source code](#)

**Operation type:** `Request-response`. The endpoint receives a message, and sends a correlated message.

**SOAP action:** `http://localhost:8082/set_node`

**Input:** `set_node_request` (soap:body, use = literal) [Source code](#)

parameters type `set_node`

- value type `node`
  - `no` type `int`
  - `type` - optional; type `node_type` - type `undefined` with restriction - enum { 'TYPE\_BETWEEN\_TWO\_NODES', 'TYPE\_BETWEEN\_TWO\_POINTS', 'TYPE\_ON\_LINE', 'TYPE\_ON\_MEMBER', 'TYPE\_STANDARD' }
  - `reference_node` - optional; type `int`
  - `coordinate_system` - optional; type `int`

- coordinate\_system\_type - optional; type *node\_coordinate\_system\_type* - type *undefined* with restriction - enum { 'COORDINATE\_SYSTEM\_CARTESIAN', 'COORDINATE\_SYSTEM\_POLAR', 'COORDINATE\_SYSTEM\_X\_CYLINDRICAL', 'COORDINATE\_SYSTEM\_Y\_CYLINDRICAL', 'COORDINATE\_SYSTEM\_Z\_CYLINDRICAL' }
- coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- coordinate\_1 - optional; type *double*
- coordinate\_2 - optional; type *double*
- coordinate\_3 - optional; type *double*
- global\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- global\_coordinate\_1 - optional; type *double*
- global\_coordinate\_2 - optional; type *double*
- global\_coordinate\_3 - optional; type *double*
- between\_two\_nodes\_start\_node - optional; type *int*
- between\_two\_points\_start\_point\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- between\_two\_points\_start\_point\_coordinate\_1 - optional; type *double*
- between\_two\_points\_start\_point\_coordinate\_2 - optional; type *double*
- between\_two\_points\_start\_point\_coordinate\_3 - optional; type *double*
- between\_two\_nodes\_end\_node - optional; type *int*
- between\_two\_points\_end\_point\_coordinates - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- between\_two\_points\_end\_point\_coordinate\_1 - optional; type *double*
- between\_two\_points\_end\_point\_coordinate\_2 - optional; type *double*
- between\_two\_points\_end\_point\_coordinate\_3 - optional; type *double*
- on\_line\_reference\_line - optional; type *int*
- on\_member\_reference\_member - optional; type *int*
- reference\_type - optional; type *node\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
- reference\_object\_projected\_length - optional; type *double*
- distance\_from\_start\_is\_defined\_as\_relative - optional; type *boolean*
- distance\_from\_start\_relative - optional; type *double*
- distance\_from\_start\_absolute - optional; type *double*
- offset\_in\_local\_direction\_y - optional; type *double*
- offset\_in\_local\_direction\_z - optional; type *double*
- offset\_in\_global\_direction\_x - optional; type *double*
- offset\_in\_global\_direction\_y - optional; type *double*
- offset\_in\_global\_direction\_z - optional; type *double*
- support - optional; type *int*
- mesh\_refinement - optional; type *int*
- concrete\_design\_ultimate\_configuration - optional; type *int*
- concrete\_design\_fire\_configuration - optional; type *int*
- punching\_design - optional; type *boolean*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_node\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_nodeResponse*

#### 194. set\_note

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_note

**Input:** set\_note\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_note*

- value type *note*
  - no type *int*
  - type - optional; type *note\_type* - type *undefined* with restriction - enum { 'NOTE\_TYPE\_LINE', 'NOTE\_TYPE\_MEMBER', 'NOTE\_TYPE\_NODE', 'NOTE\_TYPE\_POINT', 'NOTE\_TYPE\_SURFACE' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - text - optional; type *string*
  - point\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - point\_coordinate\_x - optional; type *double*
  - point\_coordinate\_y - optional; type *double*
  - point\_coordinate\_z - optional; type *double*
  - node - optional; type *int*
  - member - optional; type *int*
  - member\_reference\_type - optional; type *note\_member\_reference\_type* - type *undefined* with restriction - enum { 'REFERENCE\_TYPE\_L', 'REFERENCE\_TYPE\_XY', 'REFERENCE\_TYPE\_XZ', 'REFERENCE\_TYPE\_YZ' }
  - member\_length - optional; type *double*
  - member\_distance\_is\_defined\_as\_relative - optional; type *boolean*
  - member\_distance\_relative - optional; type *double*
  - member\_distance\_absolute - optional; type *double*
  - line - optional; type *int*
  - line\_length - optional; type *double*
  - surface - optional; type *int*

- surface\_reference\_type - optional; type *note\_surface\_reference\_type* - type *undefined* with restriction - enum { 'OFFSET\_TYPE\_XY', 'OFFSET\_TYPE\_XZ', 'OFFSET\_TYPE\_YZ' }
- surface\_first\_coordinate - optional; type *double*
- surface\_second\_coordinate - optional; type *double*
- offset - optional; type *boolean*
- offset\_type - optional; type *note\_offset\_type* - type *undefined* with restriction - enum { 'OFFSET\_TYPE\_XY', 'OFFSET\_TYPE\_XZ', 'OFFSET\_TYPE\_YZ' }
- offset\_coordinate - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- offset\_coordinate\_x - optional; type *double*
- offset\_coordinate\_y - optional; type *double*
- offset\_coordinate\_z - optional; type *double*
- rotation - optional; type *double*
- show\_comment - optional; type *boolean*
- display\_properties\_index - optional; type *int*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_note\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_noteResponse*

## 195. set\_object\_snap

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_object\_snap

**Input:** set\_object\_snap\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_object\_snap*
- value type *object\_snap*
    - no type *int*
    - type - optional; type *object\_snap\_type* - type *undefined* with restriction - enum { 'TYPE\_STANDARD' }
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - snap\_nodes - optional; type *boolean*
    - snap\_centers\_and\_focuses - optional; type *boolean*
    - snap\_intersections - optional; type *boolean*
    - snap\_perpendicular - optional; type *boolean*
    - snap\_extend - optional; type *boolean*
    - snap\_parallel - optional; type *boolean*
    - snap\_tangent - optional; type *boolean*
    - snap\_circle\_quadrants - optional; type *boolean*
    - snap\_shapes - optional; type *boolean*
    - snap\_parts - optional; type *boolean*
    - snap\_absolute\_distance - optional; type *boolean*
    - snap\_relative\_distance - optional; type *boolean*
    - snap\_snappable\_points\_only - optional; type *boolean*
    - snap\_guidelines - optional; type *boolean*
    - snap\_background\_layers - optional; type *boolean*
    - snap\_line\_grids - optional; type *boolean*
    - comment - optional; type *string*
    - parts\_count - optional; type *int*
    - absolute\_distance - optional; type *double*
    - relative\_distance - optional; type *double*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_object\_snap\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_object\_snapResponse*

## 196. set\_opening

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_opening

**Input:** set\_opening\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_opening*
- value type *opening*
    - no type *int*
    - area - optional; type *double*
    - boundary\_lines - optional; type *array\_of\_int*
    - center\_of\_opening - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - center\_of\_opening\_x - optional; type *double*
    - center\_of\_opening\_y - optional; type *double*
    - center\_of\_opening\_z - optional; type *double*
    - position\_full\_description - optional; type *string*
    - position\_Short\_description - optional; type *string*
    - surfaces - optional; type *array\_of\_int*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_opening\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_openingResponse](#)

#### 197. set\_opening\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_opening\_load

**Input:** set\_opening\_load\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_opening_load
 ■ load_case_no type int
 ■ value type opening_load
 ■ no type int
 ■ load_type - optional; type opening_load_load_type - type undefined with restriction - enum { 'LOAD_TYPE_FORCE' }
 ■ openings - optional; type array_of_int
 ■ load_case - optional; type int
 ■ coordinate_system - optional; type string
 ■ load_direction - optional; type opening_load_load_direction - type undefined with restriction - enum { 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_PROJECTED', 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE', 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE', 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_PROJECTED', 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE', 'LOAD_DIRECTION_LOCAL_Z' }
 ■ load_distribution - optional; type opening_load_load_distribution - type undefined with restriction - enum { 'LOAD_DISTRIBUTION_LINEAR_TRAPEZOIDAL', 'LOAD_DISTRIBUTION_UNIFORM_TRAPEZOIDAL' }
 ■ magnitude - optional; type double
 ■ magnitude_1 - optional; type double
 ■ magnitude_2 - optional; type double
 ■ magnitude_3 - optional; type double
 ■ node_1 - optional; type int
 ■ node_2 - optional; type int
 ■ node_3 - optional; type int
 ■ smooth_punctual_load_enabled - optional; type boolean
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

**Output:** set\_opening\_load\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_opening\\_loadResponse](#)

#### 198. set\_optimization\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_optimization\_settings

**Input:** set\_optimization\_settings\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_optimization_settings
 ■ value type optimizationSettingsConfig_type
 ■ general_optimization_active type boolean
 ■ general_keep_best_number_model_mutations type int
 ■ general_optimize_on type optimizationSettingsConfig_general_optimize_on_type - type undefined with restriction - enum { 'E_OPTIMIZE_ON_TYPE_MAX_GLOBAL_PARAMETER', 'E_OPTIMIZE_ON_TYPE_MIN_CO2_EMISSIONS', 'E_OPTIMIZE_ON_TYPE_MIN_COST', 'E_OPTIMIZE_ON_TYPE_MIN_GLOBAL_PARAMETER', 'E_OPTIMIZE_ON_TYPE_MIN_MEMBER_DEFORMATION', 'E_OPTIMIZE_ON_TYPE_MIN_NODAL_DEFORMATION', 'E_OPTIMIZE_ON_TYPE_MIN_SURFACE_DEFORMATION', 'E_OPTIMIZE_ON_TYPE_MIN_VECTORIAL_DISPLACEMENT', 'E_OPTIMIZE_ON_TYPE_MIN_WHOLE_WEIGHT', 'E_OPTIMIZE_ON_TYPE_BEGIN', 'E_OPTIMIZE_ON_TYPE_END' }
 ■ general_optimizer type optimizationSettingsConfig_general_optimizer_type - type undefined with restriction - enum { 'E_OPTIMIZER_TYPE_ALL_MUTATIONS', 'E_OPTIMIZER_TYPE_PARTICLE_SWARM', 'E_OPTIMIZER_TYPE_PERCENTS_OF_RANDOM_MUTATIONS', 'E_OPTIMIZER_TYPE_BEGIN', 'E_OPTIMIZER_TYPE_END' }
 ■ general_number_random_mutations type double
```

**Output:** set\_optimization\_settings\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_optimization\\_settingsResponse](#)

#### 199. set\_result\_combination

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_result\_combination

**Input:** set\_result\_combination\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_result_combination
 ■ value type result_combination
 ■ no type int
 ■ design_situation - optional; type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ to_solve - optional; type boolean
 ■ comment - optional; type string
 ■ combination_type - optional; type result_combination_combination_type - type undefined with restriction - enum { 'COMBINATION_TYPE_ENVELOPE_PERMANENT', 'COMBINATION_TYPE_ENVELOPE_TRANSIENT', 'COMBINATION_TYPE_GENERAL', 'COMBINATION_TYPE_SUPERPOSITION' }
 ■ srss_combination - optional; type boolean
 ■ srss_extreme_value_sign - optional; type result_combination_srss_extreme_value_sign - type undefined with restriction - enum { 'EXTREME_VALUE_SIGN_ACCORDING_TO_LC_CO', 'EXTREME_VALUE_SIGN_NEGATIVE',
```

- 'EXTREME\_VALUE\_SIGN\_POSITIVE', 'EXTREME\_VALUE\_SIGN\_POSITIVE\_OR\_NEGATIVE' }
- srss\_use\_equivalent\_linear\_combination - optional; type *boolean*
- srss\_according\_load\_case\_or\_combination - optional; type *int*
- items - optional; type *array\_of\_result\_combination\_items*
  - result\_combination\_items - optional, unbounded; type *result\_combination\_items*
    - no - optional; type *int*
    - case\_object\_item - optional; type *int*
    - operator\_type - optional; type *operator\_type* - type *undefined* with restriction - enum { 'OPERATOR\_AND', 'OPERATOR\_NONE', 'OPERATOR\_OR' }
    - left\_parenthesis - optional; type *boolean*
    - right\_parenthesis - optional; type *boolean*
    - group\_factor - optional; type *double*
    - case\_object\_factor - optional; type *double*
    - case\_object\_sub\_result\_type - optional; type *case\_object\_sub\_result\_type* - type *undefined* with restriction - enum { 'SUB\_RESULT\_INCREMENTAL\_ALL', 'SUB\_RESULT\_INCREMENTAL\_FINAL\_STATE', 'SUB\_RESULT\_INCREMENTAL\_SUB\_RESULT\_ID', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_ABSOLUTE\_SUM', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_X', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_X\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Y', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Y\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Z', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_DIRECTION\_Z\_WITH\_MODE\_SHAPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUMS\_ENVELOPE', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_X', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_Y', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SCALED\_SUM\_FULL\_Z', 'SUB\_RESULT\_SPECTRAL\_ANALYSIS\_SRSS' }
    - case\_object\_sub\_result\_id - optional; type *int*
    - case\_object\_load\_type - optional; type *case\_object\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
    - group\_load\_type - optional; type *group\_load\_type* - type *undefined* with restriction - enum { 'LOAD\_TYPE\_PERMANENT', 'LOAD\_TYPE\_TRANSIENT' }
    - action - optional; type *int*
    - is\_leading - optional; type *boolean*
    - gamma - optional; type *double*
    - psi - optional; type *double*
    - xi - optional; type *double*
    - k\_fi - optional; type *double*
    - c\_esl - optional; type *double*
    - k\_def - optional; type *double*
    - psi\_0 - optional; type *double*
    - psi\_1 - optional; type *double*
    - psi\_2 - optional; type *double*
    - fi - optional; type *double*
    - gamma\_0 - optional; type *double*
    - alfa - optional; type *double*
    - k\_f - optional; type *double*
    - phi - optional; type *double*
    - rho - optional; type *double*
    - omega\_0 - optional; type *double*
    - gamma\_1\_1 - optional; type *double*
    - k\_creep - optional; type *double*
  - generate\_subcombinations - optional; type *boolean*
  - load\_duration - optional; type *int*
  - is\_generated - optional; type *boolean*
  - consider\_construction\_stage - optional; type *int*
  - consider\_construction\_stage\_active - optional; type *boolean*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_result\_combination\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_result\_combinationResponse*

## 200. set\_result\_section

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_result\_section

**Input:** set\_result\_section\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_result\_section*

- value type *result\_section*
  - no type *int*
  - type - optional; type *result\_section\_type* - type *undefined* with restriction - enum { 'TYPE\_2\_POINTS\_AND\_VECTOR', 'TYPE\_LINE' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to\_surfaces - optional; type *array\_of\_int*
  - assigned\_to\_solids - optional; type *array\_of\_int*
  - show\_results\_in\_direction - optional; type *result\_section\_show\_results\_in\_direction* - type *undefined* with restriction - enum { 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_X', 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_Y', 'SHOW\_RESULTS\_IN\_GLOBAL\_MINUS\_Z', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_X', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_Y', 'SHOW\_RESULTS\_IN\_GLOBAL\_PLUS\_Z', 'SHOW\_RESULTS\_IN\_LOCAL\_MINUS\_Z', 'SHOW\_RESULTS\_IN\_LOCAL\_PLUS\_Y', 'SHOW\_RESULTS\_IN\_LOCAL\_PLUS\_Z' }
  - coordinate\_system - optional; type *int*
  - show\_values\_on\_isolines\_enabled - optional; type *boolean*
  - lines - optional; type *array\_of\_int*
  - first\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - first\_point\_coordinate\_1 - optional; type *double*
  - first\_point\_coordinate\_2 - optional; type *double*

- first\_point\_coordinate\_3 - optional; type *double*
- second\_point - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- second\_point\_coordinate\_1 - optional; type *double*
- second\_point\_coordinate\_2 - optional; type *double*
- second\_point\_coordinate\_3 - optional; type *double*
- projection\_in\_direction - optional; type *result\_section\_projection\_in\_direction* - type *undefined* with restriction - enum { 'PROJECTION\_IN\_GLOBAL\_X', 'PROJECTION\_IN\_GLOBAL\_Y', 'PROJECTION\_IN\_GLOBAL\_Z', 'PROJECTION\_IN\_USER\_DEFINED\_U', 'PROJECTION\_IN\_USER\_DEFINED\_V', 'PROJECTION\_IN\_USER\_DEFINED\_W', 'PROJECTION\_IN\_VECTOR' }
- vector - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- vector\_coordinate\_1 - optional; type *double*
- vector\_coordinate\_2 - optional; type *double*
- vector\_coordinate\_3 - optional; type *double*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_result\_section\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_result\_sectionResponse*

## 201. set\_rigid\_link

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_rigid\_link

**Input:** set\_rigid\_link\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_rigid\_link*

- value type *rigid\_link*
  - no type *int*
  - type - optional; type *rigid\_link\_type* - type *undefined* with restriction - enum { 'TYPE\_DIAPHRAGM', 'TYPE\_LINE\_TO\_LINE', 'TYPE\_LINE\_TO\_SURFACE' }
  - line1 - optional; type *int*
  - line2 - optional; type *int*
  - surface - optional; type *int*
  - rigid\_link\_type - optional; type *rigid\_link\_rigid\_link\_type* - type *undefined* with restriction - enum { 'RESILIENT', 'RIGID' }
  - comment - optional; type *string*
  - ignore\_relative\_position - optional; type *boolean*
  - user\_defined\_distribution - optional; type *boolean*
  - line1\_start\_is\_relative - optional; type *boolean*
  - line1\_start\_relative - optional; type *double*
  - line1\_start\_absolute - optional; type *double*
  - line1\_end\_is\_relative - optional; type *boolean*
  - line1\_end\_relative - optional; type *double*
  - line1\_end\_absolute - optional; type *double*
  - line2\_start\_is\_relative - optional; type *boolean*
  - line2\_start\_relative - optional; type *double*
  - line2\_start\_absolute - optional; type *double*
  - line2\_end\_is\_relative - optional; type *boolean*
  - line2\_end\_relative - optional; type *double*
  - line2\_end\_absolute - optional; type *double*
  - nodes - optional; type *array\_of\_int*
  - lines - optional; type *array\_of\_int*
  - center\_node\_no - optional; type *int*
  - center\_user\_defined - optional; type *boolean*
  - center\_point - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - center\_point\_x - optional; type *double*
  - center\_point\_y - optional; type *double*
  - center\_point\_z - optional; type *double*
  - link\_plane\_user\_defined - optional; type *boolean*
  - link\_plane\_node1\_no - optional; type *int*
  - link\_plane\_node2\_no - optional; type *int*
  - link\_plane\_node3\_no - optional; type *int*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_rigid\_link\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_rigid\_linkResponse*

## 202. set\_section

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_section

**Input:** set\_section\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_section*

- value type *section*
  - no type *int*



- type - optional; type *section\_type* - type *undefined* with restriction - enum { 'TYPE\_BASIC', 'TYPE\_BUILT\_UP\_STEEL', 'TYPE\_BUILT\_UP\_TIMBER', 'TYPE\_GENERAL\_BY\_RSECTION', 'TYPE\_PARAMETRIC\_BARS', 'TYPE\_PARAMETRIC\_BRIDGES', 'TYPE\_PARAMETRIC\_MASSIVE\_I', 'TYPE\_PARAMETRIC\_MASSIVE\_II', 'TYPE\_PARAMETRIC\_THIN\_WALLED', 'TYPE\_PHASE', 'TYPE\_STANDARDIZED\_STEEL', 'TYPE\_STANDARDIZED\_TIMBER' }
- manufacturing\_type - optional; type *section\_manufacturing\_type* - type *undefined* with restriction - enum { 'MANUFACTURING\_TYPE\_COLD\_FORMED', 'MANUFACTURING\_TYPE\_GLULAM', 'MANUFACTURING\_TYPE\_HOT\_ROLLED', 'MANUFACTURING\_TYPE\_NONE', 'MANUFACTURING\_TYPE\_SAWN', 'MANUFACTURING\_TYPE\_WELDED' }
- name - optional; type *string*
- shear\_stiffness\_deactivated - optional; type *boolean*
- warping\_stiffness\_deactivated - optional; type *boolean*
- thin\_walled\_model - optional; type *boolean*
- us\_spelling\_of\_properites - optional; type *boolean*
- has\_cost\_estimation - optional; type *boolean*
- has\_emissions\_estimation - optional; type *boolean*
- optimization - optional; type *boolean*
- stress\_smoothing\_to\_avoid\_singularities - optional; type *boolean*
- area\_axial - optional; type *double*
- area\_shear\_y - optional; type *double*
- area\_shear\_z - optional; type *double*
- inclination\_principal\_axes - optional; type *double*
- rotation\_angle - optional; type *double*
- mirrored\_axis\_y - optional; type *boolean*
- mirrored\_axis\_z - optional; type *boolean*
- moment\_of\_inertia\_bending\_y - optional; type *double*
- moment\_of\_inertia\_bending\_z - optional; type *double*
- moment\_of\_inertia\_torsion - optional; type *double*
- warping - optional; type *double*
- depth\_temperature\_load - optional; type *double*
- width\_temperature\_load - optional; type *double*
- material - optional; type *int*
- reference\_material - optional; type *int*
- material\_part\_1 - optional; type *int*
- material\_part\_2 - optional; type *int*
- material\_part\_3 - optional; type *int*
- material\_part\_4 - optional; type *int*
- material\_part\_5 - optional; type *int*
- material\_part\_6 - optional; type *int*
- material\_part\_7 - optional; type *int*
- material\_part\_8 - optional; type *int*
- material\_part\_9 - optional; type *int*
- material\_part\_10 - optional; type *int*
- hybrid\_active - optional; type *boolean*
- parametrization\_type - optional; type *section\_parametrization\_type* - type *undefined* with restriction - enum { 'BUILT\_UP\_STEEL\_CHANNELS\_U\_BU', 'BUILT\_UP\_STEEL\_CUT\_I\_SECTIONS\_WITH\_FLAT\_BAR\_CIF\_BU', 'BUILT\_UP\_STEEL\_CUT\_I\_SECTIONS\_WITH\_MODIFIED\_DEPTH\_CIMD\_BU', 'BUILT\_UP\_STEEL\_EQUAL\_LEG\_ANGLES\_LE\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_OR\_CHANNELS\_WITH\_SIDE\_FLAT\_BARS\_SF\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_WITH\_CHANNELS\_IU\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_WITH\_OTHER\_SECTION\_ON\_FLANGE\_I\_OF\_BU', 'BUILT\_UP\_STEEL\_I\_SECTIONS\_I\_BU', 'BUILT\_UP\_STEEL\_UNEQUAL\_LEG\_ANGLES\_LU\_BU', 'BUILT\_UP\_TIMBER\_RECTANGLES\_R\_BU', 'PARAMETRIC\_BARS\_FLAT\_BAR\_FLAT', 'PARAMETRIC\_BARS\_HALF\_OVAL\_BAR\_HALFOVAL', 'PARAMETRIC\_BARS\_HALF\_ROUND\_BAR\_HALFRROUND', 'PARAMETRIC\_BARS\_HEXAGON\_BAR\_HEXAGON', 'PARAMETRIC\_BARS\_ISOSCELES\_RIGHT\_TRIANGLE\_BAR\_TRIANGLEIR', 'PARAMETRIC\_BARS\_OCTAGON\_BAR\_OCTAGON', 'PARAMETRIC\_BARS\_ROUND\_BAR\_ROUND', 'PARAMETRIC\_BARS\_ROUND\_CORNER\_SQUARE\_BAR\_SQUARER', 'PARAMETRIC\_BARS\_SHARP\_CORNER\_SQUARE\_BAR\_SQUARES', 'PARAMETRIC\_BRIDGES\_MULTI\_PIECE\_TAPERED\_FLANGE\_T\_SECTION\_TTF\_B', 'PARAMETRIC\_MASSIVE\_II\_2X2\_MASSIVE\_RECTANGLES\_WITH\_FULLY\_RIGID\_CONNECTION\_2X2R\_M2', 'PARAMETRIC\_MASSIVE\_II\_2X3\_MASSIVE\_RECTANGLES\_WITH\_FULLY\_RIGID\_CONNECTION\_2X3R\_M2', 'PARAMETRIC\_MASSIVE\_II\_2\_MASSIVE\_RECTANGLES\_WITH\_A\_CONTINUOUS\_PACK\_AND\_2\_SPLICES\_2R2S\_M2', 'PARAMETRIC\_MASSIVE\_II\_2\_MASSIVE\_RECTANGLES\_2R\_M2', 'PARAMETRIC\_MASSIVE\_II\_3\_MASSIVE\_RECTANGLES\_3R\_M2', 'PARAMETRIC\_MASSIVE\_II\_4\_MASSIVE\_RECTANGLES\_4R\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_BOXH\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_OVERHANGING\_FLANGES\_AND\_HORIZONTAL\_CONNECTION\_LINES\_BOXOV\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_OVERHANGING\_FLANGES\_AND\_VERTICAL\_CONNECTION\_LINES\_BOXV\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_BOX\_WITH\_VERTICAL\_CONNECTION\_LINES\_BOXV\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_FIVE\_LAYER\_RECTANGLE\_WITH\_3\_DIFFERENT\_MATERIALS\_5LR\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_OVAL\_WITH\_VERTICAL\_CONNECTION\_LINES\_OVAL\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_RECTANGLE\_WITH\_2\_CONTINUOUS\_SPLICES\_R2S\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_I\_SECTION\_H\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_THIN\_FLANGES\_AND\_4\_REINFORCING\_I\_SECTION\_TFL\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_I\_SECTION\_V\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_THREE\_LAYER\_RECTANGLE\_WITH\_2\_DIFFERENT\_MATERIALS\_3LR\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_BOX\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_2BOXH\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_BOX\_WITH\_VERTICAL\_CONNECTION\_LINES\_2BOXV\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_TWO\_LAYER\_RECTANGLE\_WITH\_2\_DIFFERENT\_MATERIALS\_2LR\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_T\_SECTION\_WITH\_A\_HORIZONTAL\_CONNECTION\_LINE\_TH\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_T\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_TV\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_U\_SECTION\_WITH\_HORIZONTAL\_CONNECTION\_LINES\_UH\_M2', 'PARAMETRIC\_MASSIVE\_II\_MASSIVE\_U\_SECTION\_WITH\_VERTICAL\_CONNECTION\_LINES\_UV\_M2', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_CIRCLE\_CIRCLE\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLE\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_DTC\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLE\_T\_SECTION\_DT\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_DOUBLY\_SYMMETRIC\_I\_SECTION\_ID\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_OVAL\_HALFOVAL\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_ROUND\_HALFRROUND\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HEXAGON\_HEXAGON\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HOLLOW\_CIRCLE\_HCIRCLE\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IC\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_I\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_L\_SECTION\_L\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OCTAGON\_OCTAGON\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OVAL\_OVAL\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_CHAMFERED\_RECTANGULAR\_OPENING\_RROC\_M1', 'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_RECTANGULAR\_OPENING\_RRO\_M1',

'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_R\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_RECTANGLE\_RR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_SQUARE\_SQR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_SQUARE\_SQ\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_DOUBLE\_T\_SECTION\_DTT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_I\_SECTION\_ITF\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_T\_SECTION\_TTF\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_U\_SECTION\_UT\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TTWC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_TTW\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TRAPEZOID\_TR\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_T\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IUC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_IU\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_T\_SECTION\_TU\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_UC\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_U\_M1',  
'PARAMETRIC\_MASSIVE\_I\_MASSIVE\_Z\_SECTION\_Z\_M1',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_AX\_BOX\_AX',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_A\_BOX\_A',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_BX\_BOX\_BX',  
'PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_B\_BOX\_B', 'PARAMETRIC\_THIN\_WALLED\_CHANNEL\_U',  
'PARAMETRIC\_THIN\_WALLED\_CIRCULAR\_HOLLOW\_SECTION\_CHS',  
'PARAMETRIC\_THIN\_WALLED\_CROSS\_SHAPED\_SECTION\_CROSS',  
'PARAMETRIC\_THIN\_WALLED\_ELLIPTICAL\_HOLLOW\_SECTION\_EHS',  
'PARAMETRIC\_THIN\_WALLED\_EQUAL\_LEG\_ANGLE\_LE',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_2\_SIDE\_FLAT\_BARS\_I2SFB',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_T\_SECTION\_IT',  
'PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_I', 'PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_A\_PI\_A',  
'PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_B\_PI\_B',  
'PARAMETRIC\_THIN\_WALLED\_POLYGONAL\_HOLLOW\_SECTION\_PHS',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAK\_OUTWARD\_AND\_INWARD\_RHSPO',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAK\_OUTWARD\_RHSPO',  
'PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_RHS',  
'PARAMETRIC\_THIN\_WALLED\_RIB\_HOLLOW\_SECTION\_RIBHS',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_FLAT\_BARS\_I2SFB',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_SLOPING\_STIFFENERS\_I2SS',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_WELDED\_FLAT\_BARS\_I2FBW',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_DOUBLE\_WEB\_THICKNESS\_I2WT',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_EDGE\_STIFFENERS\_ON\_1\_FLANGE\_IS1FES',  
'PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_IS',  
'PARAMETRIC\_THIN\_WALLED\_SQUARE\_HOLLOW\_SECTION\_SHS',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_CHANNEL\_UTF',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_I\_SECTION\_ITF',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_WITH\_TAPERED\_WEB\_TFTTW',  
'PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_TTF',  
'PARAMETRIC\_THIN\_WALLED\_TRAPEZOIDAL\_HOLLOW\_SECTION\_THS',  
'PARAMETRIC\_THIN\_WALLED\_T\_SECTION\_T', 'PARAMETRIC\_THIN\_WALLED\_UNEQUAL\_LEG\_ANGLE\_LU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_INWARD\_UUESI',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_AND\_INWARD\_UUE',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_UUESO',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_UU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_I\_SECTION\_IU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_RECTANGULAR\_HOLLOW\_SECTION\_RHSU',  
'PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_T\_SECTION\_TU',  
'PARAMETRIC\_THIN\_WALLED\_Z\_SECTION\_Z', 'STANDARDIZED\_STEEL\_60\_DEGREE\_ANGLES\_L60\_S',  
'STANDARDIZED\_STEEL\_BAR\_SECTIONS\_BAR\_S',  
'STANDARDIZED\_STEEL\_CIRCULAR\_HOLLOW\_SECTIONS\_PIPES\_CHS\_S',  
'STANDARDIZED\_STEEL\_ELLIPTICAL\_AND\_SEMI\_ELLIPTICAL\_HOLLOW\_SECTIONS\_EHS\_S',  
'STANDARDIZED\_STEEL\_I\_BEAMS\_H\_BEAMS\_I\_S',  
'STANDARDIZED\_STEEL\_L\_SECTIONS\_STEEL\_ANGLES\_L\_S',  
'STANDARDIZED\_STEEL\_RAIL\_SECTIONS\_RAIL\_S', 'STANDARDIZED\_STEEL\_SHEETS\_SHEET\_S',  
'STANDARDIZED\_STEEL\_SQUARE\_AND\_RECTANGULAR\_HOLLOW\_SECTIONS\_RHS\_S',  
'STANDARDIZED\_STEEL\_T\_SECTIONS\_STEEL\_TEES\_T\_S',  
'STANDARDIZED\_STEEL\_U\_SECTIONS\_STEEL\_CHANNELS\_U\_S',  
'STANDARDIZED\_STEEL\_Z\_SECTIONS\_STEEL\_ZEES\_Z\_S',  
'STANDARDIZED\_TIMBER\_TIMBER\_COMPOSED\_SECTIONS\_COMP\_S',  
'STANDARDIZED\_TIMBER\_TIMBER\_RECTANGLES\_RECT\_S'}

- combination\_type - optional; type *section\_combination\_type* - type *undefined* with restriction - enum {  
'BUILT\_UP\_NG\_2I', 'BUILT\_UP\_NG\_2I\_A', 'BUILT\_UP\_NG\_2LCI', 'BUILT\_UP\_NG\_2LCIIR',  
'BUILT\_UP\_NG\_2LCIIR\_A', 'BUILT\_UP\_NG\_2LCI\_A', 'BUILT\_UP\_NG\_2LCLO', 'BUILT\_UP\_NG\_2LCLOC',  
'BUILT\_UP\_NG\_2LCLOC\_A', 'BUILT\_UP\_NG\_2LCLO', 'BUILT\_UP\_NG\_2LCLO\_A', 'BUILT\_UP\_NG\_2LCLO\_A',  
'BUILT\_UP\_NG\_2LHLI', 'BUILT\_UP\_NG\_2LHLI\_A', 'BUILT\_UP\_NG\_2LHLO', 'BUILT\_UP\_NG\_2LHLO\_A',  
'BUILT\_UP\_NG\_2LLHLI', 'BUILT\_UP\_NG\_2LLHLI\_A', 'BUILT\_UP\_NG\_2LLHLO', 'BUILT\_UP\_NG\_2LLHLO\_A',  
'BUILT\_UP\_NG\_2LSHLI', 'BUILT\_UP\_NG\_2LSHLI\_A', 'BUILT\_UP\_NG\_2LSHLO', 'BUILT\_UP\_NG\_2LSHLO\_A',  
'BUILT\_UP\_NG\_2UF', 'BUILT\_UP\_NG\_2UI', 'BUILT\_UP\_NG\_2UI\_A', 'BUILT\_UP\_NG\_2UO', 'BUILT\_UP\_NG\_2UO\_A',  
'BUILT\_UP\_NG\_3I', 'BUILT\_UP\_NG\_4LHLO', 'BUILT\_UP\_NG\_4LHLO\_A', 'BUILT\_UP\_NG\_4LLI',  
'BUILT\_UP\_NG\_4LLI\_A', 'BUILT\_UP\_NG\_4LLO', 'BUILT\_UP\_NG\_4LLO\_A', 'BUILT\_UP\_NG\_CIHf',  
'BUILT\_UP\_NG\_CIMD', 'BUILT\_UP\_NG\_HNBH', 'BUILT\_UP\_NG\_I1F', 'BUILT\_UP\_NG\_I1HIC', 'BUILT\_UP\_NG\_I1IC',  
'BUILT\_UP\_NG\_I2F', 'BUILT\_UP\_NG\_I2HIC', 'BUILT\_UP\_NG\_I2I', 'BUILT\_UP\_NG\_ICI-MAX', 'BUILT\_UP\_NG\_ICI-MIN',  
'BUILT\_UP\_NG\_IHF', 'BUILT\_UP\_NG\_IUD', 'BUILT\_UP\_NG\_IUU', 'BUILT\_UP\_NG\_IVF', 'BUILT\_UP\_NG\_IWT',  
'BUILT\_UP\_NG\_UF' }
- corrugated\_sheet\_width - optional; type *double*
- rail\_worn\_out - optional; type *double*
- phase\_parent - optional; type *int*
- phase\_parts - optional; type *string*
- comment - optional; type *string*
- advanced\_time\_dependent\_properties\_of\_concrete\_enabled - optional; type *boolean*
- creep\_enabled - optional; type *boolean*
- shrinkage\_enabled - optional; type *boolean*
- relative\_humidity - optional; type *double*
- function\_data\_function\_type - optional; type *section\_function\_data\_function\_type* - type *undefined* with restriction - enum { 'FUNCTION\_TYPE\_CREEP\_COEFFICIENT', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CA', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CD', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CS' }
- function\_data\_part\_id - optional; type *int*
- function\_data\_age\_of\_concrete\_at\_the\_considered\_moment - optional; type *double*
- function\_data\_number\_of\_steps - optional; type *int*
- function\_data\_coefficients - optional; type *array\_of\_section\_function\_data\_coefficients*
  - section\_function\_data\_coefficients - optional, unbounded; type *section\_function\_data\_coefficients*

- no - optional; type *int*
- time - optional; type *double*
- coefficient - optional; type *double*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- cost\_estimation\_apply\_from\_material - optional; type *boolean*
- members\_weight\_active - optional; type *boolean*
- members\_weight\_unit\_cost - optional; type *double*
- members\_weight\_unit - optional; type *section\_members\_weight\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CWT', 'COST\_ESTIMATION\_G', 'COST\_ESTIMATION\_KG', 'COST\_ESTIMATION\_LB', 'COST\_ESTIMATION\_OZ', 'COST\_ESTIMATION\_SLUG', 'COST\_ESTIMATION\_T', 'COST\_ESTIMATION\_TON' }
- members\_weight\_quantity - optional; type *double*
- members\_weight\_cost - optional; type *double*
- members\_volume\_active - optional; type *boolean*
- members\_volume\_unit\_cost - optional; type *double*
- members\_volume\_unit - optional; type *section\_members\_volume\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM3', 'COST\_ESTIMATION\_FLOZ', 'COST\_ESTIMATION\_FT3', 'COST\_ESTIMATION\_IMPGAL', 'COST\_ESTIMATION\_IN3', 'COST\_ESTIMATION\_L', 'COST\_ESTIMATION\_M3', 'COST\_ESTIMATION\_MM3', 'COST\_ESTIMATION\_PT', 'COST\_ESTIMATION\_QT', 'COST\_ESTIMATION\_USGAL', 'COST\_ESTIMATION\_YD3' }
- members\_volume\_quantity - optional; type *double*
- members\_volume\_cost - optional; type *double*
- members\_surface\_active - optional; type *boolean*
- members\_surface\_unit\_cost - optional; type *double*
- members\_surface\_unit - optional; type *section\_members\_surface\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM2', 'COST\_ESTIMATION\_FT2', 'COST\_ESTIMATION\_IN2', 'COST\_ESTIMATION\_M2', 'COST\_ESTIMATION\_MM2', 'COST\_ESTIMATION\_YD2' }
- members\_surface\_quantity - optional; type *double*
- members\_surface\_cost - optional; type *double*
- members\_length\_active - optional; type *boolean*
- members\_length\_unit\_cost - optional; type *double*
- members\_length\_unit - optional; type *section\_members\_length\_unit* - type *undefined* with restriction - enum { 'COST\_ESTIMATION\_CM', 'COST\_ESTIMATION\_FT', 'COST\_ESTIMATION\_IN', 'COST\_ESTIMATION\_KM', 'COST\_ESTIMATION\_M', 'COST\_ESTIMATION\_MM', 'COST\_ESTIMATION\_YD' }
- members\_length\_quantity - optional; type *double*
- members\_length\_cost - optional; type *double*
- sum\_cost - optional; type *double*
- cost\_percentage - optional; type *double*
- total\_cost - optional; type *double*
- emissions\_estimation\_apply\_from\_material - optional; type *boolean*
- emissions\_members\_weight\_active - optional; type *boolean*
- emissions\_members\_weight\_unit\_cost - optional; type *double*
- emissions\_members\_weight\_unit - optional; type *section\_emissions\_members\_weight\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CWT', 'EMISSIONS\_ESTIMATION\_G', 'EMISSIONS\_ESTIMATION\_KG', 'EMISSIONS\_ESTIMATION\_LB', 'EMISSIONS\_ESTIMATION\_OZ', 'EMISSIONS\_ESTIMATION\_SLUG', 'EMISSIONS\_ESTIMATION\_T', 'EMISSIONS\_ESTIMATION\_TON' }
- emissions\_members\_weight\_quantity - optional; type *double*
- emissions\_members\_weight\_cost - optional; type *double*
- emissions\_members\_volume\_active - optional; type *boolean*
- emissions\_members\_volume\_unit\_cost - optional; type *double*
- emissions\_members\_volume\_unit - optional; type *section\_emissions\_members\_volume\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM3', 'EMISSIONS\_ESTIMATION\_FLOZ', 'EMISSIONS\_ESTIMATION\_FT3', 'EMISSIONS\_ESTIMATION\_IMPGAL', 'EMISSIONS\_ESTIMATION\_IN3', 'EMISSIONS\_ESTIMATION\_L', 'EMISSIONS\_ESTIMATION\_M3', 'EMISSIONS\_ESTIMATION\_MM3', 'EMISSIONS\_ESTIMATION\_PT', 'EMISSIONS\_ESTIMATION\_QT', 'EMISSIONS\_ESTIMATION\_USGAL', 'EMISSIONS\_ESTIMATION\_YD3' }
- emissions\_members\_volume\_quantity - optional; type *double*
- emissions\_members\_volume\_cost - optional; type *double*
- emissions\_members\_surface\_active - optional; type *boolean*
- emissions\_members\_surface\_unit\_cost - optional; type *double*
- emissions\_members\_surface\_unit - optional; type *section\_emissions\_members\_surface\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM2', 'EMISSIONS\_ESTIMATION\_FT2', 'EMISSIONS\_ESTIMATION\_IN2', 'EMISSIONS\_ESTIMATION\_M2', 'EMISSIONS\_ESTIMATION\_MM2', 'EMISSIONS\_ESTIMATION\_YD2' }
- emissions\_members\_surface\_quantity - optional; type *double*
- emissions\_members\_surface\_cost - optional; type *double*
- emissions\_members\_length\_active - optional; type *boolean*
- emissions\_members\_length\_unit\_cost - optional; type *double*
- emissions\_members\_length\_unit - optional; type *section\_emissions\_members\_length\_unit* - type *undefined* with restriction - enum { 'EMISSIONS\_ESTIMATION\_CM', 'EMISSIONS\_ESTIMATION\_FT', 'EMISSIONS\_ESTIMATION\_IN', 'EMISSIONS\_ESTIMATION\_KM', 'EMISSIONS\_ESTIMATION\_M', 'EMISSIONS\_ESTIMATION\_MM', 'EMISSIONS\_ESTIMATION\_YD' }
- emissions\_members\_length\_quantity - optional; type *double*
- emissions\_members\_length\_cost - optional; type *double*
- emissions\_sum\_cost - optional; type *double*
- emissions\_cost\_percentage - optional; type *double*
- emissions\_total\_cost - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_section\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_sectionResponse*

### 203. set\_soil\_massif

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_soil\_massif

Input: set\_soil\_massif\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_soil\_massif*

- value type *soil\_massif*
  - no type *int*
  - type - optional; type *soil\_massif\_type* - type *undefined* with restriction - enum { 'TYPE\_PHANTOM', 'TYPE\_STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*

- name - optional; type *string*
- assigned\_to\_type - optional; type *soil\_massif\_assigned\_to\_type* - type *undefined* with restriction - enum { 'ASSIGNED\_TO\_TYPE\_SOIL\_SAMPLES', 'ASSIGNED\_TO\_TYPE\_SOIL\_SOLIDS' }
- assigned\_to\_soil\_samples - optional; type *array\_of\_int*
- assigned\_to\_solids - optional; type *array\_of\_int*
- assigned\_to\_solid\_sets - optional; type *array\_of\_int*
- assigned\_to\_solids\_and\_solid\_sets - optional; type *string*
- topology\_type - optional; type *soil\_massif\_topology\_type* - type *undefined* with restriction - enum { 'TOPOLOGY\_TYPE\_RECTANGLE' }
- depth\_according\_to\_soil\_samples - optional; type *boolean*
- diameter\_for\_circle\_topology - optional; type *double*
- polyline\_for\_polygon\_topology - optional; type *int*
- center\_x - optional; type *double*
- center\_y - optional; type *double*
- size - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- size\_x - optional; type *double*
- size\_y - optional; type *double*
- size\_z - optional; type *double*
- rotation\_about\_z - optional; type *double*
- groundwater - optional; type *boolean*
- groundwater\_surface - optional; type *int*
- model\_soil\_block\_via - optional; type *soil\_massif\_model\_soil\_block\_via* - type *undefined* with restriction - enum { 'MODEL\_SOIL\_BLOCK\_VIA\_FE\_SOLIDS', 'MODEL\_SOIL\_BLOCK\_VIA\_SUPPORT\_SPRINGS' }
- relevant\_permanent\_loading - optional; type *int*
- mesh\_nodes\_in\_position\_of\_precise\_layer\_surfaces - optional; type *boolean*
- rock\_beneath\_last\_layer - optional; type *boolean*
- change\_mesh\_refinement - optional; type *boolean*
- size\_increasing\_fe\_during\_growing\_depth - optional; type *double*
- comment - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_soil\_massif\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_soil\_massifResponse*

#### 204. set\_soil\_sample

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_soil\_sample

**Input:** set\_soil\_sample\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_soil\_sample*
- value type *soil\_sample*
    - no type *int*
    - type - optional; type *soil\_sample\_type* - type *undefined* with restriction - enum { 'TYPE\_PHANTOM', 'TYPE\_STANDARD' }
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - coordinate\_0 - optional; type *double*
    - coordinate\_1 - optional; type *double*
    - coordinate\_2 - optional; type *double*
    - import\_coordinate\_z\_from\_terrain - optional; type *boolean*
    - groundwater - optional; type *boolean*
    - groundwater\_ordinate - optional; type *double*
    - layers\_table - optional; type *array\_of\_soil\_sample\_layers\_table*
      - soil\_sample\_layers\_table - optional, unbounded; type *soil\_sample\_layers\_table*
        - no - optional; type *int*
        - layer\_no - optional; type *int*
        - soil\_material - optional; type *int*
        - thickness - optional; type *double*
        - bottom\_ordinate - optional; type *double*
    - layers\_thickness\_sum - optional; type *double*
    - comment - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_soil\_sample\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_soil\_sampleResponse*

#### 205. set\_solid

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid

**Input:** set\_solid\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_solid*
- value type *solid*
    - no type *int*
    - type - optional; type *solid\_type* - type *undefined* with restriction - enum { 'TYPE\_CONTACT', 'TYPE\_GAS', 'TYPE\_HOLE', 'TYPE\_INTERSECTION', 'TYPE\_SOIL', 'TYPE\_STANDARD' }
    - analytical\_center\_of\_gravity - optional; type *vector\_3d*
      - x type *double*
      - y type *double*

- z type *double*
- analytical\_center\_of\_gravity\_x - optional; type *double*
- analytical\_center\_of\_gravity\_y - optional; type *double*
- analytical\_center\_of\_gravity\_z - optional; type *double*
- analytical\_mass - optional; type *double*
- analytical\_surface\_area - optional; type *double*
- analytical\_volume - optional; type *double*
- boundary\_surfaces - optional; type *array\_of\_int*
- center\_of\_gravity - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- center\_of\_gravity\_x - optional; type *double*
- center\_of\_gravity\_y - optional; type *double*
- center\_of\_gravity\_z - optional; type *double*
- gas - optional; type *int*
- is\_deactivated\_for\_calculation - optional; type *boolean*
- mass - optional; type *double*
- material - optional; type *int*
- mesh\_refinement - optional; type *int*
- solid\_contact - optional; type *int*
- solid\_contact\_first\_surface - optional; type *int*
- solid\_contact\_second\_surface - optional; type *int*
- stress\_analysis\_configuration - optional; type *int*
- surface\_area - optional; type *double*
- volume - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- grid\_enabled - optional; type *boolean*
- grid\_inner\_points - optional; type *boolean*
- grid\_boundary\_points - optional; type *boolean*
- specific\_direction\_type - optional; type *solid\_specific\_direction\_type* - type *undefined* with restriction - enum { 'DIRECTION\_TYPE\_DIRECTED\_TO\_NODE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_LINE', 'DIRECTION\_TYPE\_PARALLEL\_TO\_CS\_OF\_MEMBER', 'DIRECTION\_TYPE\_PARALLEL\_TO\_TWO\_NODES', 'DIRECTION\_TYPE\_ROTATED\_VIA\_3\_ANGLES' }
- grid\_distance\_x - optional; type *double*
- grid\_distance\_y - optional; type *double*
- grid\_distance\_z - optional; type *double*
- use\_center\_of\_gravity\_as\_grid\_origin - optional; type *boolean*
- grid\_origin - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- grid\_origin\_x - optional; type *double*
- grid\_origin\_y - optional; type *double*
- grid\_origin\_z - optional; type *double*
- grid\_adapt\_automatically - optional; type *boolean*
- grid\_point\_count\_negative\_u - optional; type *int*
- grid\_point\_count\_positive\_u - optional; type *int*
- grid\_point\_count\_negative\_v - optional; type *int*
- grid\_point\_count\_positive\_v - optional; type *int*
- grid\_point\_count\_negative\_w - optional; type *int*
- grid\_point\_count\_positive\_w - optional; type *int*
- specific\_direction\_enabled - optional; type *boolean*
- coordinate\_system - optional; type *int*
- axes\_sequence - optional; type *solid\_axes\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_YXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
- rotated\_about\_angle\_x - optional; type *double*
- rotated\_about\_angle\_y - optional; type *double*
- rotated\_about\_angle\_z - optional; type *double*
- rotated\_about\_angle\_1 - optional; type *double*
- rotated\_about\_angle\_2 - optional; type *double*
- rotated\_about\_angle\_3 - optional; type *double*
- directed\_to\_node\_direction\_node - optional; type *int*
- directed\_to\_node\_plane\_node - optional; type *int*
- directed\_to\_node\_first\_axis - optional; type *solid\_directed\_to\_node\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- directed\_to\_node\_second\_axis - optional; type *solid\_directed\_to\_node\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_first\_node - optional; type *int*
- parallel\_to\_two\_nodes\_second\_node - optional; type *int*
- parallel\_to\_two\_nodes\_plane\_node - optional; type *int*
- parallel\_to\_two\_nodes\_first\_axis - optional; type *solid\_parallel\_to\_two\_nodes\_first\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_two\_nodes\_second\_axis - optional; type *solid\_parallel\_to\_two\_nodes\_second\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- parallel\_to\_line - optional; type *int*
- parallel\_to\_member - optional; type *int*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_solid\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_solidResponse*

## 206. set\_solid\_contacts

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_contacts

**Input:** set\_solid\_contacts\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_solid\_contacts*

- value type *solid\_contacts*

- no *type int*
- comment - optional; *type string*
- friction\_coefficient - optional; *type double*
- limit\_stress - optional; *type double*
- name - optional; *type string*
- parallel\_to\_surface - optional; *type solid\_contacts\_parallel\_to\_surface* - *type undefined* with restriction - enum { 'ELASTIC\_FRICTION', 'ELASTIC\_FRICTION\_LIMIT', 'ELASTIC\_SOLID', 'FAILURE\_IF\_CONTACT\_PERPENDICULAR\_TO\_SURFACES\_FAILED', 'FULL\_FORCE\_TRANSMISSION', 'RIGID\_FRICTION', 'RIGID\_FRICTION\_LIMIT' }
- perpendicular\_to\_surface - optional; *type solid\_contacts\_perpendicular\_to\_surface* - *type undefined* with restriction - enum { 'FAILURE\_UNDER\_COMPRESSION', 'FAILURE\_UNDER\_TENSION', 'FULL\_FORCE\_TRANSMISSION' }
- shear\_stiffness - optional; *type double*
- solids - optional; *type array\_of\_int*
- user\_defined\_name\_enabled - optional; *type boolean*
- id\_for\_export\_import - optional; *type string*
- metadata\_for\_export\_import - optional; *type string*

**Output:** set\_solid\_contacts\_response (soap:body, use = literal) [Source code](#)

parameters *type set\_solid\_contactsResponse*

## 207. set\_solid\_gas

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_gas

**Input:** set\_solid\_gas\_request (soap:body, use = literal) [Source code](#)

parameters *type set\_solid\_gas*

- value *type solid\_gas*
  - no *type int*
  - comment - optional; *type string*
  - name - optional; *type string*
  - pressure - optional; *type double*
  - solids - optional; *type array\_of\_int*
  - temperature - optional; *type double*
  - user\_defined\_name\_enabled - optional; *type boolean*
  - id\_for\_export\_import - optional; *type string*
  - metadata\_for\_export\_import - optional; *type string*

**Output:** set\_solid\_gas\_response (soap:body, use = literal) [Source code](#)

parameters *type set\_solid\_gasResponse*

## 208. set\_solid\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_load

**Input:** set\_solid\_load\_request (soap:body, use = literal) [Source code](#)

parameters *type set\_solid\_load*

- load\_case\_no *type int*
- value *type solid\_load*
  - no *type int*
  - load\_type - optional; *type solid\_load\_load\_type* - *type undefined* with restriction - enum { 'LOAD\_TYPE\_BUOYANCY', 'LOAD\_TYPE\_FORCE', 'LOAD\_TYPE\_GAS', 'LOAD\_TYPE\_ROTARY\_MOTION', 'LOAD\_TYPE\_STRAIN', 'LOAD\_TYPE\_TEMPERATURE' }
  - solids - optional; *type array\_of\_int*
  - load\_case - optional; *type int*
  - load\_distribution - optional; *type solid\_load\_load\_distribution* - *type undefined* with restriction - enum { 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_X', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Y', 'LOAD\_DISTRIBUTION\_LINEAR\_IN\_Z', 'LOAD\_DISTRIBUTION\_UNIFORM' }
  - load\_direction - optional; *type solid\_load\_load\_direction* - *type undefined* with restriction - enum { 'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE', 'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE' }
  - load\_direction\_orientation - optional; *type solid\_load\_load\_direction\_orientation* - *type undefined* with restriction - enum { 'LOAD\_DIRECTION\_FORWARD', 'LOAD\_DIRECTION\_REVERSED' }
  - uniform\_magnitude - optional; *type double*
  - magnitude\_1 - optional; *type double*
  - magnitude\_2 - optional; *type double*
  - strain\_uniform\_magnitude\_x - optional; *type double*
  - strain\_uniform\_magnitude\_y - optional; *type double*
  - strain\_uniform\_magnitude\_z - optional; *type double*
  - strain\_magnitude\_x1 - optional; *type double*
  - strain\_magnitude\_y1 - optional; *type double*
  - strain\_magnitude\_z1 - optional; *type double*
  - strain\_magnitude\_x2 - optional; *type double*
  - strain\_magnitude\_y2 - optional; *type double*
  - strain\_magnitude\_z2 - optional; *type double*
  - node\_1 - optional; *type int*
  - node\_2 - optional; *type int*
  - is\_density\_defined\_by\_altitude - optional; *type boolean*
  - altitude - optional; *type double*
  - angular\_velocity - optional; *type double*
  - angular\_acceleration - optional; *type double*
  - axis\_definition\_type - optional; *type solid\_load\_axis\_definition\_type* - *type undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
  - axis\_definition\_p1 - optional; *type vector\_3d*
    - x *type double*
    - y *type double*
    - z *type double*
  - axis\_definition\_p1\_x - optional; *type double*

- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *solid\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *solid\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- gas\_magnitude - optional; type *double*
- gas\_behaviour - optional; type *solid\_load\_gas\_behaviour* - type *undefined* with restriction - enum { 'GAS\_BEHAVIOUR\_OVERPRESSURE\_INCREMENT', 'GAS\_BEHAVIOUR\_RESULTING\_OVERPRESSURE', 'GAS\_BEHAVIOUR\_RESULTING\_VOLUME', 'GAS\_BEHAVIOUR\_VOLUME\_INCREMENT' }
- coordinate\_system - optional; type *int*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_solid\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_solid\_loadResponse*

## 209. set\_solid\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_mesh\_refinement

**Input:** set\_solid\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_solid\_mesh\_refinement*
- value type *solid\_mesh\_refinement*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - solids - optional; type *array\_of\_int*
    - target\_length - optional; type *double*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_solid\_mesh\_refinement\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_solid\_mesh\_refinementResponse*

## 210. set\_solid\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_set

**Input:** set\_solid\_set\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_solid\_set*
- value type *solid\_set*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - set\_type - optional; type *solid\_set\_set\_type* - type *undefined* with restriction - enum { 'SET\_TYPE\_CONTINUOUS', 'SET\_TYPE\_GROUP' }
    - solids - optional; type *array\_of\_int*
    - surface\_area - optional; type *double*
    - volume - optional; type *double*
    - mass - optional; type *double*
    - center\_of\_gravity - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - center\_of\_gravity\_x - optional; type *double*
    - center\_of\_gravity\_y - optional; type *double*
    - center\_of\_gravity\_z - optional; type *double*
    - stress\_analysis\_configuration - optional; type *int*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_solid\_set\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_solid\_setResponse*

## 211. set\_solid\_set\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_solid\_set\_load

**Input:** set\_solid\_set\_load\_request (soap:body, use = literal) [Source code](#)

```

parameters type set_solid_set_load
 ■ load_case_no type int
 ■ value type solid_set_load
 ■ no type int
 ■ load_type - optional; type solid_set_load_load_type - type undefined with restriction - enum {
 'LOAD_TYPE_BUOYANCY', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_GAS', 'LOAD_TYPE_ROTARY_MOTION',
 'LOAD_TYPE_STRAIN', 'LOAD_TYPE_TEMPERATURE' }
 ■ solid_sets - optional; type array_of_int
 ■ load_case - optional; type int
 ■ load_distribution - optional; type solid_set_load_load_distribution - type undefined with restriction - enum {
 'LOAD_DISTRIBUTION_LINEAR_IN_X', 'LOAD_DISTRIBUTION_LINEAR_IN_Y',
 'LOAD_DISTRIBUTION_LINEAR_IN_Z', 'LOAD_DISTRIBUTION_UNIFORM' }
 ■ load_direction - optional; type solid_set_load_load_direction - type undefined with restriction - enum {
 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE',
 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE',
 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE' }
 ■ load_direction_orientation - optional; type solid_set_load_load_direction_orientation - type undefined with
 restriction - enum { 'LOAD_DIRECTION_FORWARD', 'LOAD_DIRECTION_REVERSED' }
 ■ uniform_magnitude - optional; type double
 ■ magnitude_1 - optional; type double
 ■ magnitude_2 - optional; type double
 ■ strain_uniform_magnitude_x - optional; type double
 ■ strain_uniform_magnitude_y - optional; type double
 ■ strain_uniform_magnitude_z - optional; type double
 ■ strain_magnitude_x1 - optional; type double
 ■ strain_magnitude_y1 - optional; type double
 ■ strain_magnitude_z1 - optional; type double
 ■ strain_magnitude_x2 - optional; type double
 ■ strain_magnitude_y2 - optional; type double
 ■ strain_magnitude_z2 - optional; type double
 ■ node_1 - optional; type int
 ■ node_2 - optional; type int
 ■ is_density_defined_by_altitude - optional; type boolean
 ■ altitude - optional; type double
 ■ angular_acceleration - optional; type double
 ■ angular_velocity - optional; type double
 ■ axis_definition_type - optional; type solid_set_load_axis_definition_type - type undefined with restriction - enum {
 'AXIS_DEFINITION_POINT_AND_AXIS', 'AXIS_DEFINITION_TWO_POINTS' }
 ■ axis_definition_p1 - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ axis_definition_p1_x - optional; type double
 ■ axis_definition_p1_y - optional; type double
 ■ axis_definition_p1_z - optional; type double
 ■ axis_definition_p2 - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ axis_definition_p2_x - optional; type double
 ■ axis_definition_p2_y - optional; type double
 ■ axis_definition_p2_z - optional; type double
 ■ axis_definition_axis - optional; type solid_set_load_axis_definition_axis - type undefined with restriction - enum {
 'AXIS_X', 'AXIS_Y', 'AXIS_Z' }
 ■ axis_definition_axis_orientation - optional; type solid_set_load_axis_definition_axis_orientation - type
 undefined with restriction - enum { 'AXIS_NEGATIVE', 'AXIS_POSITIVE' }
 ■ gas_magnitude - optional; type double
 ■ gas_behaviour - optional; type solid_set_load_gas_behaviour - type undefined with restriction - enum {
 'GAS_BEHAVIOUR_OVERPRESSURE_INCREMENT', 'GAS_BEHAVIOUR_RESULTING_OVERPRESSURE',
 'GAS_BEHAVIOUR_RESULTING_VOLUME', 'GAS_BEHAVIOUR_VOLUME_INCREMENT' }
 ■ coordinate_system - optional; type int
 ■ comment - optional; type string
 ■ is_generated - optional; type boolean
 ■ generating_object_info - optional; type string
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string

```

**Output:** *set\_solid\_set\_load\_response* (soap:body, use = literal) [Source code](#)

```

parameters type set_solid_set_loadResponse

```

## 212. *set\_spectral\_analysis\_settings*

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** [http://localhost:8082/set\\_spectral\\_analysis\\_settings](http://localhost:8082/set_spectral_analysis_settings)

**Input:** *set\_spectral\_analysis\_settings\_request* (soap:body, use = literal) [Source code](#)

```

parameters type set_spectral_analysis_settings
 ■ value type spectral_analysis_settings
 ■ no type int
 ■ user_defined_name_enabled - optional; type boolean
 ■ name - optional; type string
 ■ comment - optional; type string
 ■ assigned_to - optional; type string
 ■ combination_rule_for_periodic_responses - optional; type
 spectral_analysis_settings_combination_rule_for_periodic_responses - type undefined with restriction -
 enum { 'ABSOLUTE_SUM', 'CQC', 'SRSS' }
 ■ use_equivalent_linear_combination - optional; type boolean
 ■ signed_results_using_dominant_mode - optional; type boolean
 ■ include_missing_masses - optional; type boolean
 ■ combination_rule_for_missing_masses - optional; type
 spectral_analysis_settings_combination_rule_for_missing_masses - type undefined with restriction - enum
 { 'ABSOLUTE_SUM', 'SRSS' }
 ■ save_results_of_all_selected_modes - optional; type boolean

```



- combination\_rule\_for\_directional\_components - optional; type *spectral\_analysis\_settings\_combination\_rule\_for\_directional\_components* - type *undefined* with restriction - enum { 'ABSOLUTE\_SUM', 'SCALED\_SUM', 'SRSS' }
- combination\_rule\_for\_directional\_components\_value - optional; type *double*
- damping\_for\_cqc\_rule - optional; type *spectral\_analysis\_settings\_damping\_for\_cqc\_rule* - type *undefined* with restriction - enum { 'CONSTANT\_FOR\_EACH\_MODE', 'DIFFERENT\_FOR\_EACH\_MODE' }
- constant\_d\_for\_each\_mode - optional; type *double*
- zero\_periodic\_acceleration\_type - optional; type *spectral\_analysis\_settings\_zero\_periodic\_acceleration\_type* - type *undefined* with restriction - enum { 'ACCORDING\_TO\_RESPONSE\_SPECTRUM', 'SPECTRAL\_ACCELERATION\_OF\_LAST\_CALCULATED\_FREQUENCY', 'USER\_DEFINED' }
- user\_defined\_zpa - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_spectral\_analysis\_settings\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_spectral\_analysis\_settingsResponse*

### 213. set\_static\_analysis\_settings

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_static\_analysis\_settings

Input: set\_static\_analysis\_settings\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_static\_analysis\_settings*

- value type *static\_analysis\_settings*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - analysis\_type - optional; type *static\_analysis\_settings\_analysis\_type* - type *undefined* with restriction - enum { 'GEOMETRICALLY\_LINEAR', 'LARGE\_DEFORMATIONS', 'SECOND\_ORDER\_P\_DELTA' }
  - assign\_reduce\_stiffness\_enabled - optional; type *static\_analysis\_settings\_assign\_reduce\_stiffness\_enabled* - type *undefined* with restriction - enum { 'ASSIGN\_REDUCED\_STIFFNESS\_TO\_FAILING\_MEMBERS', 'FAILING\_MEMBERS\_TO\_BE\_REMOVED\_INDIVIDUALLY\_DURING\_SUCCESSIVE\_ITERATIONS' }
  - comment - optional; type *string*
  - consider\_favorable\_effect\_due\_to\_tension\_in\_members - optional; type *boolean*
  - cutting\_patterns\_settings - optional; type *boolean*
  - deformation\_of\_failing\_members\_and\_reactivation\_enabled - optional; type *boolean*
  - displacements\_due\_to\_bourdon\_effect - optional; type *boolean*
  - divide\_results\_by\_loading\_factor - optional; type *boolean*
  - exceptional\_handling\_enabled - optional; type *boolean*
  - instability\_detection\_tolerance - optional; type *double*
  - integrate\_preliminary\_form\_finding\_enabled - optional; type *boolean*
  - iterative\_calculation\_robustness - optional; type *double*
  - iterative\_method\_for\_nonlinear\_analysis - optional; type *static\_analysis\_settings\_iterative\_method\_for\_nonlinear\_analysis* - type *undefined* with restriction - enum { 'DYNAMIC\_RELAXATION', 'NEWTON\_RAPHSON', 'NEWTON\_RAPHSON\_COMBINED\_WITH\_PICARD', 'NEWTON\_RAPHSON\_WITH\_CONSTANT\_STIFFNESS', 'NEWTON\_RAPHSON\_WITH\_POSTCRITICAL\_ANALYSIS', 'PICARD' }
  - loading\_multiplier\_factor - optional; type *double*
  - mass\_conversion\_acceleration\_in\_direction\_x - optional; type *double*
  - mass\_conversion\_acceleration\_in\_direction\_y - optional; type *double*
  - mass\_conversion\_acceleration\_in\_direction\_z - optional; type *double*
  - mass\_conversion\_defined\_as\_acceleration - optional; type *boolean*
  - mass\_conversion\_enabled - optional; type *boolean*
  - mass\_conversion\_factor\_in\_direction\_x - optional; type *double*
  - mass\_conversion\_factor\_in\_direction\_y - optional; type *double*
  - mass\_conversion\_factor\_in\_direction\_z - optional; type *double*
  - max\_number\_of\_iterations - optional; type *int*
  - maximum\_number\_of\_reactivations - optional; type *int*
  - method\_of\_equation\_system - optional; type *static\_analysis\_settings\_method\_of\_equation\_system* - type *undefined* with restriction - enum { 'METHOD\_OF\_EQUATION\_SYSTEM\_DIRECT', 'METHOD\_OF\_EQUATION\_SYSTEM\_ITERATIVE' }
  - modify\_loading\_by\_multiplier\_factor - optional; type *boolean*
  - nonsymmetric\_direct\_solver - optional; type *boolean*
  - number\_of\_iterations\_for\_loading\_prestress - optional; type *int*
  - number\_of\_load\_increments - optional; type *int*
  - percentage\_of\_iteration - optional; type *int*
  - plate\_bending\_theory - optional; type *static\_analysis\_settings\_plate\_bending\_theory* - type *undefined* with restriction - enum { 'PLATE\_BENDING\_THEORY\_KIRCHHOFF', 'PLATE\_BENDING\_THEORY\_MINDLIN' }
  - precision\_of\_convergence\_criteria\_for\_nonlinear\_calculation - optional; type *double*
  - ratio\_of\_distance\_of\_cutting\_lines\_node\_to\_mesh - optional; type *double*
  - reduction\_factor\_of\_stiffness - optional; type *int*
  - refer\_internal\_forces\_to\_deformed\_structure - optional; type *boolean*
  - refer\_internal\_forces\_to\_deformed\_structure\_for\_moments - optional; type *boolean*
  - refer\_internal\_forces\_to\_deformed\_structure\_for\_normal\_forces - optional; type *boolean*
  - refer\_internal\_forces\_to\_deformed\_structure\_for\_shear\_forces - optional; type *boolean*
  - relative\_setting\_of\_time\_step\_for\_dynamic\_relaxation - optional; type *double*
  - save\_results\_of\_all\_load\_increments - optional; type *boolean*
  - smoothness\_of\_boundary\_lines - optional; type *double*
  - speed\_of\_convergence - optional; type *double*
  - standard\_precision\_and\_tolerance\_settings\_enabled - optional; type *boolean*
  - ignore\_all\_nonlinearities\_enabled - optional; type *boolean*
  - try\_to\_calculate\_instabil\_structure - optional; type *boolean*
  - calculation\_diagrams\_enabled - optional; type *boolean*
  - calculation\_diagrams\_list - optional; type *array\_of\_static\_analysis\_settings\_calculation\_diagrams\_list*
    - static\_analysis\_settings\_calculation\_diagrams\_list - optional, unbounded; type *static\_analysis\_settings\_calculation\_diagrams\_list*
      - no - optional; type *int*
      - vertical\_axis\_result\_type - optional; type *vertical\_axis\_result\_type* - type *undefined* with restriction - enum { 'INCREMENT', 'MAXIMUM\_DEFORMATION', 'MEMBER\_CONTACT\_FORCES', 'MEMBER\_GLOBAL\_DEFORMATIONS', 'MEMBER\_INTERNAL\_FORCES', 'MEMBER\_LOCAL\_DEFORMATIONS', 'MEMBER\_STRAINS', 'NODE\_GLOBAL\_DEFORMATIONS', 'SOLID\_BASIC\_PLASTIC\_STRAINS', 'SOLID\_BASIC\_STRESSES', 'SOLID\_BASIC\_TOTAL\_STRAINS', 'SOLID\_EQUIVALENT\_PLASTIC\_STRAINS', 'SOLID\_EQUIVALENT\_STRESSES' }

'SOLID\_EQUIVALENT\_TOTAL\_STRAINS', 'SOLID\_GAS', 'SOLID\_GLOBAL\_DEFORMATIONS',  
'SOLID\_PRINCIPAL\_PLASTIC\_STRAINS', 'SOLID\_PRINCIPAL\_STRESSES',  
'SOLID\_PRINCIPAL\_TOTAL\_STRAINS', 'SUM\_OF\_SUPPORT\_FORCES',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS', 'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS',  
'SURFACE\_BASIC\_INTERNAL\_FORCES', 'SURFACE\_BASIC\_PLASTIC\_STRAINS',  
'SURFACE\_BASIC\_STRESSES', 'SURFACE\_BASIC\_TOTAL\_STRAINS',  
'SURFACE\_CONTACT\_STRESSES', 'SURFACE\_DESIGN\_INTERNAL\_FORCES',  
'SURFACE\_GLOBAL\_DEFORMATIONS', 'SURFACE\_LOCAL\_DEFORMATIONS',  
'SURFACE\_MAXIMUM\_PLASTIC\_STRAINS', 'SURFACE\_MAXIMUM\_TOTAL\_STRAINS',  
'SURFACE\_PLASTIC\_STRAINS\_BACH\_P', 'SURFACE\_PLASTIC\_STRAINS\_MISES\_P',  
'SURFACE\_PLASTIC\_STRAINS\_RANKINE\_P', 'SURFACE\_PLASTIC\_STRAINS\_TRESCA\_P',  
'SURFACE\_PRINCIPAL\_INTERNAL\_FORCES', 'SURFACE\_PRINCIPAL\_PLASTIC\_STRAINS',  
'SURFACE\_PRINCIPAL\_STRESSES', 'SURFACE\_PRINCIPAL\_TOTAL\_STRAINS',  
'SURFACE\_STRAINS\_BACH', 'SURFACE\_STRAINS\_MISES', 'SURFACE\_STRAINS\_RANKINE',  
'SURFACE\_STRAINS\_TRESCA', 'SURFACE\_STRESSES\_SIGMA\_EQV\_BACH',  
'SURFACE\_STRESSES\_SIGMA\_EQV\_MISES', 'SURFACE\_STRESSES\_SIGMA\_EQV\_RANKINE',  
'SURFACE\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SURFACE\_STRESS\_COMPONENTS' }

- vertical\_axis\_value\_type - optional; type *vertical\_axis\_value\_type* - type *undefined* with restriction -  
enum { 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U',  
'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_X',  
'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Z',  
'MEMBERS\_GLOBAL\_DEFORMATIONS\_U', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_X',  
'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_X',  
'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_U',  
'MEMBER\_RESULTS\_CONTACT\_FORCES\_PRINCIPAL\_AXES\_P\_V',  
'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Y',  
'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Z', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_PRI',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_SEC', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MY',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MZ',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_M\_OMEGA', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_N',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_P',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_U',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_V',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_U',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_V',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VY',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VZ',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_V',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_V',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_WARPING', 'MEMBER\_RESULTS\_STRAINS\_EPS\_X',  
'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XY', 'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_X', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Y',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Z',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XU',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XV',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_U',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_V',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'NODES\_GLOBAL\_DEFORMATIONS\_U',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_Z', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Z',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SOLID\_RESULTS\_GAS\_INITIAL\_PRESSURE', 'SOLID\_RESULTS\_GAS\_OVERPRESSURE',  
'SOLID\_RESULTS\_GAS\_PRESSURE', 'SOLID\_RESULTS\_GAS\_TEMPERATURE',  
'SOLID\_RESULTS\_GAS\_VOLUME', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_123',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_3',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_MISES',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Z',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_TRESCA',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XZ',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_YZ', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_123', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_3', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_MISES', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_TRESCA', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_X',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_Y', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_Z',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_XY', 'SOLID\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_YZ', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_1',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_123', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_2',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_3', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_BACH',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_MISES',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_RANKINE',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_X',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_Y', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_Z',  
'SOLID\_RESULTS\_STRESSES\_TAU\_MAX', 'SOLID\_RESULTS\_STRESSES\_TAU\_XY',  
'SOLID\_RESULTS\_STRESSES\_TAU\_XZ', 'SOLID\_RESULTS\_STRESSES\_TAU\_YZ',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_X',

'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Z',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VY',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_SIGMA\_Z',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_YZ',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NC\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NX\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NY\_D',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_PLUS',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_M',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_BETA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_MT\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_M',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_ALPHA\_MINUS', 'SURFACE\_RESULTS\_STRAINS\_ALPHA\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_PLUS',

'SURFACE\_RESULTS\_STRAINS\_EPSILON\_BACH', 'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_M',  
'SURFACE\_RESULTS\_STRESSES\_ALFA\_MINUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_M', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_B', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_X\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_B',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_Y\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_TAU\_MAX',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_B', 'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_M',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_XY\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_STRESSES\_TAU\_YZ', 'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_X',  
'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_Y', 'TYPE\_SUM\_OF\_SUPPORT\_FORCES\_Z' }

- vertical\_axis\_object - optional; type *int*
- vertical\_axis\_node - optional; type *int*
- horizontal\_axis\_result\_type - optional; type *horizontal\_axis\_result\_type* - type *undefined* with restriction - enum { 'INCREMENT', 'MAXIMUM\_DEFORMATION', 'MEMBER\_CONTACT\_FORCES', 'MEMBER\_GLOBAL\_DEFORMATIONS', 'MEMBER\_INTERNAL\_FORCES', 'MEMBER\_LOCAL\_DEFORMATIONS', 'MEMBER\_STRAINS', 'NODE\_GLOBAL\_DEFORMATIONS', 'SOLID\_BASIC\_PLASTIC\_STRAINS', 'SOLID\_BASIC\_STRESSES', 'SOLID\_BASIC\_TOTAL\_STRAINS', 'SOLID\_EQUIVALENT\_PLASTIC\_STRAINS', 'SOLID\_EQUIVALENT\_STRESSES', 'SOLID\_EQUIVALENT\_TOTAL\_STRAINS', 'SOLID\_GAS', 'SOLID\_GLOBAL\_DEFORMATIONS', 'SOLID\_PRINCIPAL\_PLASTIC\_STRAINS', 'SOLID\_PRINCIPAL\_STRESSES', 'SOLID\_PRINCIPAL\_TOTAL\_STRAINS', 'SUM\_OF\_SUPPORT\_FORCES', 'SUPPORT\_REACTIONS\_LINE\_SUPPORTS', 'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS', 'SURFACE\_BASIC\_INTERNAL\_FORCES', 'SURFACE\_BASIC\_PLASTIC\_STRAINS', 'SURFACE\_BASIC\_STRESSES', 'SURFACE\_BASIC\_TOTAL\_STRAINS', 'SURFACE\_CONTACT\_STRESSES', 'SURFACE\_DESIGN\_INTERNAL\_FORCES', 'SURFACE\_GLOBAL\_DEFORMATIONS', 'SURFACE\_LOCAL\_DEFORMATIONS', 'SURFACE\_MAXIMUM\_PLASTIC\_STRAINS', 'SURFACE\_MAXIMUM\_TOTAL\_STRAINS', 'SURFACE\_PLASTIC\_STRAINS\_BACH\_P', 'SURFACE\_PLASTIC\_STRAINS\_MISES\_P', 'SURFACE\_PLASTIC\_STRAINS\_RANKINE\_P', 'SURFACE\_PLASTIC\_STRAINS\_TRESCA\_P', 'SURFACE\_PRINCIPAL\_INTERNAL\_FORCES', 'SURFACE\_PRINCIPAL\_PLASTIC\_STRAINS', 'SURFACE\_PRINCIPAL\_STRESSES', 'SURFACE\_PRINCIPAL\_TOTAL\_STRAINS', 'SURFACE\_STRAINS\_BACH', 'SURFACE\_STRAINS\_MISES', 'SURFACE\_STRAINS\_RANKINE', 'SURFACE\_STRAINS\_TRESCA', 'SURFACE\_STRESSES\_SIGMA\_EQV\_BACH', 'SURFACE\_STRESSES\_SIGMA\_EQV\_MISES', 'SURFACE\_STRESSES\_SIGMA\_EQV\_RANKINE', 'SURFACE\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SURFACE\_STRESS\_COMPONENTS' }
- horizontal\_axis\_value\_type - optional; type *horizontal\_axis\_value\_type* - type *undefined* with restriction - enum { 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_X', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MAXIMUM\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_X', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'MEMBERS\_GLOBAL\_DEFORMATIONS\_U\_Z', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_Y', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_M\_Z', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_U', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_X', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Y', 'MEMBER\_RESULTS\_CONTACT\_FORCES\_P\_Z', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_PRI', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MT\_SEC', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MY', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_MZ', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_M\_OMEGA', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_N', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_P', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_U', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_M\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_U' }

'MEMBER\_RESULTS\_INTERNAL\_FORCES\_PRINCIPAL\_AXES\_V\_V',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_V', 'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VY',  
'MEMBER\_RESULTS\_INTERNAL\_FORCES\_VZ',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_PHI\_V',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_PRINCIPAL\_AXES\_U\_V',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'MEMBER\_RESULTS\_LOCAL\_DEFORMATIONS\_WARPING', 'MEMBER\_RESULTS\_STRAINS\_EPS\_X',  
'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XY', 'MEMBER\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_X', 'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Y',  
'MEMBER\_RESULTS\_STRAINS\_KAPPA\_Z',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XU',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_GAMMA\_XV',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_U',  
'MEMBER\_RESULTS\_STRAINS\_PRINCIPAL\_AXES\_KAPPA\_V',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'NODES\_GLOBAL\_DEFORMATIONS\_U',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_X', 'NODES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'NODES\_GLOBAL\_DEFORMATIONS\_U\_Z', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_PHI\_Z',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_X',  
'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Y', 'SOLIDS\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SOLID\_RESULTS\_GAS\_INITIAL\_PRESSURE', 'SOLID\_RESULTS\_GAS\_OVERPRESSURE',  
'SOLID\_RESULTS\_GAS\_PRESSURE', 'SOLID\_RESULTS\_GAS\_TEMPERATURE',  
'SOLID\_RESULTS\_GAS\_VOLUME', 'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_123',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_3',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_MISES',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Z',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_TRESCA',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XZ',  
'SOLID\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_YZ', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_1',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_123', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_2',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_3', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_BACH',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_MISES', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_TRESCA', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_X',  
'SOLID\_RESULTS\_STRAINS\_EPSILON\_Y', 'SOLID\_RESULTS\_STRAINS\_EPSILON\_Z',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_XY', 'SOLID\_RESULTS\_STRAINS\_GAMMA\_XZ',  
'SOLID\_RESULTS\_STRAINS\_GAMMA\_YZ', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_1',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_123', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_2',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_3', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_BACH',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_MISES',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_RANKINE',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_EQV\_TRESCA', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_X',  
'SOLID\_RESULTS\_STRESSES\_SIGMA\_Y', 'SOLID\_RESULTS\_STRESSES\_SIGMA\_Z',  
'SOLID\_RESULTS\_STRESSES\_TAU\_MAX', 'SOLID\_RESULTS\_STRESSES\_TAU\_XY',  
'SOLID\_RESULTS\_STRESSES\_TAU\_XZ', 'SOLID\_RESULTS\_STRESSES\_TAU\_YZ',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_LINE\_SUPPORTS\_LOCAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_GLOBAL\_P\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_M\_Z',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_X',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Y',  
'SUPPORT\_REACTIONS\_NODAL\_SUPPORTS\_LOCAL\_P\_Z',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_PHI\_Z', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_X', 'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Y',  
'SURFACES\_GLOBAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_MY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NXY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_NY',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VX',  
'SURFACE\_RESULTS\_BASIC\_INTERNAL\_FORCES\_VY',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_SIGMA\_Z',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_XZ',  
'SURFACE\_RESULTS\_CONTACT\_STRESSES\_TAU\_YZ',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MC\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_MINUS',

'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MX\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_MINUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_MY\_D\_PLUS',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NC\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NX\_D',  
'SURFACE\_RESULTS\_DESIGN\_INTERNAL\_FORCES\_NY\_D',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_PHI\_Z',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_X',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Y',  
'SURFACE\_RESULTS\_LOCAL\_DEFORMATIONS\_U\_Z',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ABS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_ALPHA\_P\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_1\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_2\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MAX\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MIN\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_BACH',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_MISES',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_RANKINE',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_TRESCA',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_X\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_EPSILON\_P\_Y\_PLUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_MINUS',  
'SURFACE\_RESULTS\_PLASTIC\_STRAINS\_GAMMA\_P\_XY\_PLUS',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_ALFA\_M',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_BETA\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_M2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_MT\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N1',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_N2',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_B',  
'SURFACE\_RESULTS\_PRINCIPAL\_INTERNAL\_FORCES\_V\_MAX\_M',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_ABS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_ALPHA\_MINUS', 'SURFACE\_RESULTS\_STRAINS\_ALPHA\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_1\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_2\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_BACH', 'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MAX\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MIN\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_BACH',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_MISES',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_PLUS\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_RANKINE',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_TRESCA',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_X\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_EPSILON\_Y\_PLUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_MINUS',  
'SURFACE\_RESULTS\_STRAINS\_GAMMA\_XY\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_M',  
'SURFACE\_RESULTS\_STRESSES\_ALFA\_MINUS', 'SURFACE\_RESULTS\_STRESSES\_ALFA\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_M', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_1\_PLUS', 'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_M',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_MINUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_2\_PLUS',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MAX\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_MISES',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_RANKINE',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_MINUS\_TRESCA',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_BACH',  
'SURFACE\_RESULTS\_STRESSES\_SIGMA\_EQV\_M\_MISES',

```
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_M_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_BACH',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_MISES',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_RANKINE',
'SURFACE_RESULTS_STRESSES_SIGMA_EQV_PLUS_TRESCA',
'SURFACE_RESULTS_STRESSES_SIGMA_X_B', 'SURFACE_RESULTS_STRESSES_SIGMA_X_M',
'SURFACE_RESULTS_STRESSES_SIGMA_X_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_X_PLUS', 'SURFACE_RESULTS_STRESSES_SIGMA_Y_B',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_M',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_MINUS',
'SURFACE_RESULTS_STRESSES_SIGMA_Y_PLUS', 'SURFACE_RESULTS_STRESSES_TAU_MAX',
'SURFACE_RESULTS_STRESSES_TAU_XY_B', 'SURFACE_RESULTS_STRESSES_TAU_XY_M',
'SURFACE_RESULTS_STRESSES_TAU_XY_MINUS',
'SURFACE_RESULTS_STRESSES_TAU_XY_PLUS', 'SURFACE_RESULTS_STRESSES_TAU_XZ',
'SURFACE_RESULTS_STRESSES_TAU_YZ', 'TYPE_SUM_OF_SUPPORT_FORCES_X',
'TYPE_SUM_OF_SUPPORT_FORCES_Y', 'TYPE_SUM_OF_SUPPORT_FORCES_Z' }
 ■ horizontal_axis_object - optional; type int
 ■ horizontal_axis_node - optional; type int
 ■ id_for_export_import - optional; type string
 ■ metadata_for_export_import - optional; type string
```

**Output:** set\_static\_analysis\_settings\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_static\_analysis\_settingsResponse*

## 214. set\_structure\_modification

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_structure\_modification

**Input:** set\_structure\_modification\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_structure\_modification*

- value type *structure\_modification*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to - optional; type *string*
  - comment - optional; type *string*
  - modify\_stiffnesses\_gamma\_m - optional; type *boolean*
  - modify\_stiffnesses\_materials - optional; type *boolean*
  - modify\_stiffnesses\_sections - optional; type *boolean*
  - modify\_stiffnesses\_members - optional; type *boolean*
  - modify\_stiffnesses\_surfaces - optional; type *boolean*
  - modify\_stiffnesses\_member\_hinges - optional; type *boolean*
  - modify\_stiffnesses\_line\_hinges - optional; type *boolean*
  - modify\_stiffnesses\_nodal\_supports - optional; type *boolean*
  - modify\_stiffnesses\_line\_supports - optional; type *boolean*
  - modify\_stiffnesses\_member\_supports - optional; type *boolean*
  - modify\_stiffnesses\_surface\_supports - optional; type *boolean*
  - modify\_stiffness\_member\_reinforcement - optional; type *boolean*
  - modify\_stiffness\_surface\_reinforcement - optional; type *boolean*
  - modify\_stiffness\_timber\_members\_due\_moisture\_class - optional; type *boolean*
  - nonlinearities\_disabled\_material\_nonlinearity\_models - optional; type *boolean*
  - nonlinearities\_disabled\_material\_temperature\_nonlinearities - optional; type *boolean*
  - nonlinearities\_disabled\_line\_hinges - optional; type *boolean*
  - nonlinearities\_disabled\_member\_types - optional; type *boolean*
  - nonlinearities\_disabled\_member\_hinges - optional; type *boolean*
  - nonlinearities\_disabled\_member\_nonlinearities - optional; type *boolean*
  - nonlinearities\_disabled\_solid\_types\_contact\_or\_surfaces\_contact - optional; type *boolean*
  - nonlinearities\_disabled\_nodal\_supports - optional; type *boolean*
  - nonlinearities\_disabled\_line\_supports - optional; type *boolean*
  - nonlinearities\_disabled\_member\_supports - optional; type *boolean*
  - nonlinearities\_disabled\_surface\_supports - optional; type *boolean*
  - modify\_stiffnesses\_material\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_material\_table*
    - structure\_modification\_modify\_stiffnesses\_material\_table - optional, unbounded; type *structure\_modification\_modify\_stiffnesses\_material\_table*
      - no - optional; type *int*
      - material\_name - optional; type *int*
      - modification\_type - optional; type *modification\_type* - type *undefined* with restriction - enum { 'DIVISION\_FACTOR', 'MULTIPLY\_FACTOR' }
      - E\_and\_G - optional; type *double*
      - comment - optional; type *string*
  - modify\_stiffnesses\_section\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_section\_table*
    - structure\_modification\_modify\_stiffnesses\_section\_table - optional, unbounded; type *structure\_modification\_modify\_stiffnesses\_section\_table*
      - no - optional; type *int*
      - section\_name - optional; type *string*
      - A - optional; type *double*
      - A\_y - optional; type *double*
      - A\_z - optional; type *double*
      - J - optional; type *double*
      - I\_y - optional; type *double*
      - I\_z - optional; type *double*
  - modify\_stiffnesses\_member\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_member\_table*
    - structure\_modification\_modify\_stiffnesses\_member\_table - optional, unbounded; type *structure\_modification\_modify\_stiffnesses\_member\_table*
      - no - optional; type *int*
      - member\_modification - optional; type *int*
      - members - optional; type *array\_of\_int*
      - comment - optional; type *string*
  - modify\_stiffnesses\_surface\_table - optional; type *array\_of\_structure\_modification\_modify\_stiffnesses\_surface\_table*

- `structure_modification_modify_stiffnesses_surface_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_surface_table`
  - `no` - optional; type `int`
  - `surface_modification` - optional; type `int`
  - `surfaces` - optional; type `array_of_int`
  - `comment` - optional; type `string`
- `modify_stiffnesses_member_hinges_table` - optional; type `array_of_structure_modification_modify_stiffnesses_member_hinges_table`
  - `structure_modification_modify_stiffnesses_member_hinges_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_member_hinges_table`
    - `no` - optional; type `int`
    - `member_side` - optional; type `string`
    - `C_u_x` - optional; type `double`
    - `C_u_y` - optional; type `double`
    - `C_u_z` - optional; type `double`
    - `C_phi_x` - optional; type `double`
    - `C_phi_y` - optional; type `double`
    - `C_phi_z` - optional; type `double`
- `modify_stiffnesses_line_hinges_table` - optional; type `array_of_structure_modification_modify_stiffnesses_line_hinges_table`
  - `structure_modification_modify_stiffnesses_line_hinges_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_line_hinges_table`
    - `no` - optional; type `int`
    - `C_u_x` - optional; type `double`
    - `C_u_y` - optional; type `double`
    - `C_u_z` - optional; type `double`
    - `C_phi_x` - optional; type `double`
- `modify_stiffnesses_nodal_supports_table` - optional; type `array_of_structure_modification_modify_stiffnesses_nodal_supports_table`
  - `structure_modification_modify_stiffnesses_nodal_supports_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_nodal_supports_table`
    - `no` - optional; type `int`
    - `C_u_X` - optional; type `double`
    - `C_u_Y` - optional; type `double`
    - `C_u_Z` - optional; type `double`
    - `C_phi_X` - optional; type `double`
    - `C_phi_Y` - optional; type `double`
    - `C_phi_Z` - optional; type `double`
- `modify_stiffnesses_line_supports_table` - optional; type `array_of_structure_modification_modify_stiffnesses_line_supports_table`
  - `structure_modification_modify_stiffnesses_line_supports_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_line_supports_table`
    - `no` - optional; type `int`
    - `C_u_X` - optional; type `double`
    - `C_u_Y` - optional; type `double`
    - `C_u_Z` - optional; type `double`
    - `C_phi_X` - optional; type `double`
    - `C_phi_Y` - optional; type `double`
    - `C_phi_Z` - optional; type `double`
- `modify_stiffnesses_member_supports_table` - optional; type `array_of_structure_modification_modify_stiffnesses_member_supports_table`
  - `structure_modification_modify_stiffnesses_member_supports_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_member_supports_table`
    - `no` - optional; type `int`
    - `C_u_x` - optional; type `double`
    - `C_u_y` - optional; type `double`
    - `C_u_z` - optional; type `double`
    - `C_s_x` - optional; type `double`
    - `C_s_y` - optional; type `double`
    - `C_s_z` - optional; type `double`
    - `C_phi_x` - optional; type `double`
- `modify_stiffnesses_surface_supports_table` - optional; type `array_of_structure_modification_modify_stiffnesses_surface_supports_table`
  - `structure_modification_modify_stiffnesses_surface_supports_table` - optional, unbounded; type `structure_modification_modify_stiffnesses_surface_supports_table`
    - `no` - optional; type `int`
    - `C_u_X` - optional; type `double`
    - `C_u_Y` - optional; type `double`
    - `C_u_Z` - optional; type `double`
    - `C_v_xz` - optional; type `double`
    - `C_v_yz` - optional; type `double`
- `deactivate_members_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_members` - optional; type `int`
- `deactivate_surfaces_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_surfaces` - optional; type `int`
- `deactivate_solids_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_solids` - optional; type `int`
- `deactivate_support_on_nodes_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_support_on_nodes` - optional; type `int`
- `deactivate_support_on_lines_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_support_on_lines` - optional; type `int`
- `deactivate_support_on_members_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_support_on_members` - optional; type `int`
- `deactivate_support_on_surfaces_enabled` - optional; type `boolean`
- `object_selection_for_deactivate_support_on_surfaces` - optional; type `int`
- `id_for_export_import` - optional; type `string`
- `metadata_for_export_import` - optional; type `string`

Output: `set_structure_modification_response` (soap:body, use = literal) [Source code](#)

parameters type `set_structure_modificationResponse`

## 215. `set_surface`

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.



SOAP action: http://localhost:8082/set\_surface

Input: set\_surface\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface
 value type Surface
 no type int
 geometry - optional; type Surface_geometry - type undefined with restriction - enum { 'GEOMETRY_NURBS',
'GEOMETRY_PIPE', 'GEOMETRY_PLANE', 'GEOMETRY_QUADRANGLE', 'GEOMETRY_ROTATED' }
 type - optional; type Surface_type - type undefined with restriction - enum { 'TYPE_GROUNDWATER',
'TYPE_LOAD_TRANSFER', 'TYPE_MEMBRANE', 'TYPE_RIGID', 'TYPE_STANDARD',
'TYPE_WITHOUT_MEMBRANE_TENSION', 'TYPE_WITHOUT_THICKNESS' }
 boundary_lines - optional; type array_of_int
 thickness - optional; type int
 material - optional; type int
 analytical_area - optional; type double
 analytical_volume - optional; type double
 analytical_mass - optional; type double
 analytical_center_of_gravity - optional; type vector_3d
 x type double
 y type double
 z type double
 analytical_center_of_gravity_x - optional; type double
 analytical_center_of_gravity_y - optional; type double
 analytical_center_of_gravity_z - optional; type double
 area - optional; type double
 volume - optional; type double
 mass - optional; type double
 center_of_gravity - optional; type vector_3d
 x type double
 y type double
 z type double
 center_of_gravity_x - optional; type double
 center_of_gravity_y - optional; type double
 center_of_gravity_z - optional; type double
 position - optional; type string
 position_short - optional; type string
 grid_enabled - optional; type boolean
 is_deactivated_for_calculation - optional; type boolean
 comment - optional; type string
 design_properties_via_surface - optional; type boolean
 load_transfer_direction - optional; type Surface_load_transfer_direction - type undefined with restriction - enum {
'LOAD_TRANSFER_DIRECTION_IN_BOTH', 'LOAD_TRANSFER_DIRECTION_IN_X',
'LOAD_TRANSFER_DIRECTION_IN_Y' }
 is_surface_weight_enabled - optional; type boolean
 surface_weight - optional; type double
 consider_member_eccentricity - optional; type boolean
 consider_section_distribution - optional; type boolean
 excluded_members - optional; type array_of_int
 excluded_parallel_to_members - optional; type array_of_int
 excluded_lines - optional; type array_of_int
 excluded_parallel_to_lines - optional; type array_of_int
 loaded_members - optional; type array_of_int
 loaded_lines - optional; type array_of_int
 nurbs_control_point_count_in_direction_u - optional; type int
 nurbs_control_point_count_in_direction_v - optional; type int
 nurbs_order_in_direction_u - optional; type int
 nurbs_order_in_direction_v - optional; type int
 nurbs_control_points - optional; type array_of_Surface_nurbs_control_points_rows
 surface_nurbs_control_points_rows - optional, unbounded; type Surface_nurbs_control_points_row
 surface_nurbs_control_points_row - optional, unbounded; type Surface_nurbs_control_points
 u - optional; type int
 v - optional; type int
 global_coordinates - optional; type vector_3d
 x type double
 y type double
 z type double
 coordinates - optional; type vector_3d
 x type double
 y type double
 z type double
 weight type double
 quadrangle_corner_nodes - optional; type array_of_int
 quadrangle_corner_node_1 - optional; type int
 quadrangle_corner_node_2 - optional; type int
 quadrangle_corner_node_3 - optional; type int
 quadrangle_corner_node_4 - optional; type int
 pipe_radius - optional; type double
 pipe_center_line - optional; type int
 pipe_generated_lines - optional; type array_of_int
 has_line_hinges - optional; type boolean
 support - optional; type int
 eccentricity - optional; type int
 mesh_refinement - optional; type int
 meshing_type - optional; type Surface_meshing_type - type undefined with restriction - enum {
'MESHING_TYPE_FREE', 'MESHING_TYPE_MAPPED', 'MESHING_TYPE_USE_GLOBAL_SETTINGS' }
 input_axes_rotation_specification_type - optional; type Surface_input_axes_rotation_specification_type - type
undefined with restriction - enum { 'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_ANGULAR_ROTATION',
'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_DIRECT_TO_POINT',
'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_COORDINATE_SYSTEM',
'INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_LINES' }
 input_axes_angular_rotation - optional; type double
 input_axes_axis - optional; type Surface_input_axes_axis - type undefined with restriction - enum { 'AXIS_X',
'AXIS_Y' }
 input_axes_lines - optional; type array_of_int
 input_axes_point_1 - optional; type vector_3d
 x type double
```

- y type *double*
- z type *double*
- input\_axes\_point\_1\_x - optional; type *double*
- input\_axes\_point\_1\_y - optional; type *double*
- input\_axes\_point\_1\_z - optional; type *double*
- input\_axes\_point\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- input\_axes\_point\_2\_x - optional; type *double*
- input\_axes\_point\_2\_y - optional; type *double*
- input\_axes\_point\_2\_z - optional; type *double*
- input\_axes\_coordinate\_system - optional; type *int*
- result\_axes\_rotation\_specification\_type - optional; type *surface\_result\_axes\_rotation\_specification\_type* - type *undefined* with restriction - enum { 'RESULT\_AXES\_ROTATION\_SPECIFICATION\_TYPE\_IDENTICAL\_TO\_INPUT\_AXES' }
- reversed\_normal - optional; type *boolean*
- grid\_type - optional; type *surface\_grid\_type* - type *undefined* with restriction - enum { 'GRID\_TYPE\_CARTESIAN', 'GRID\_TYPE\_POLAR' }
- grid\_origin - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- grid\_origin\_x - optional; type *double*
- grid\_origin\_y - optional; type *double*
- grid\_origin\_z - optional; type *double*
- grid\_point\_count\_negative\_x - optional; type *int*
- grid\_point\_count\_positive\_x - optional; type *int*
- grid\_point\_count\_negative\_y - optional; type *int*
- grid\_point\_count\_positive\_y - optional; type *int*
- grid\_numbering\_increment - optional; type *int*
- grid\_point\_count\_r - optional; type *int*
- grid\_distance\_x - optional; type *double*
- grid\_distance\_y - optional; type *double*
- grid\_distance\_r - optional; type *double*
- grid\_rotation\_alpha - optional; type *double*
- grid\_rotation\_beta - optional; type *double*
- grid\_angle\_gamma - optional; type *double*
- grid\_adapt\_automatically - optional; type *boolean*
- auto\_detection\_of\_integrated\_objects - optional; type *boolean*
- integrated\_nodes - optional; type *array\_of\_int*
- integrated\_lines - optional; type *array\_of\_int*
- integrated\_openings - optional; type *array\_of\_int*
- has\_integrated\_objects - optional; type *boolean*
- has\_input\_axes\_rotation - optional; type *boolean*
- has\_result\_axes\_rotation - optional; type *boolean*
- timber\_service\_class - optional; type *int*
- timber\_moisture\_class - optional; type *int*
- timber\_service\_conditions - optional; type *int*
- surface\_reinforcements - optional; type *array\_of\_int*
- is\_user\_defined\_concrete\_cover\_enabled - optional; type *boolean*
- concrete\_cover\_top - optional; type *double*
- concrete\_cover\_bottom - optional; type *double*
- user\_defined\_concrete\_cover\_top - optional; type *double*
- user\_defined\_concrete\_cover\_bottom - optional; type *double*
- concrete\_durability\_top - optional; type *int*
- concrete\_durability\_bottom - optional; type *int*
- reinforcement\_direction\_top - optional; type *int*
- reinforcement\_direction\_bottom - optional; type *int*
- deflection\_check\_surface\_type - optional; type *surface\_deflection\_check\_surface\_type* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_SURFACE\_TYPE\_CANTILEVER', 'DEFLECTION\_CHECK\_SURFACE\_TYPE\_DOUBLE\_SUPPORTED' }
- deflection\_check\_displacement\_reference - optional; type *surface\_deflection\_check\_displacement\_reference* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_DEFORMED\_USER\_DEFINED\_REFERENCE\_PLANE', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_PARALLEL\_SURFACE', 'DEFLECTION\_CHECK\_DISPLACEMENT\_REFERENCE\_UNDEFORMED\_SYSTEM' }
- deflection\_check\_reference\_length\_z - optional; type *double*
- deflection\_check\_reference\_length\_z\_definition\_type - optional; type *surface\_deflection\_check\_reference\_length\_z\_definition\_type* - type *undefined* with restriction - enum { 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_BY\_MAXIMUM\_BOUNDARY\_LINE', 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_BY\_MINIMUM\_BOUNDARY\_LINE', 'DEFLECTION\_CHECK\_REFERENCE\_LENGTH\_DEFINITION\_TYPE\_MANUALLY' }
- deflection\_check\_reference\_plane\_point\_1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_1\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_1\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_1\_z - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_2\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_2\_z - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- deflection\_check\_reference\_plane\_point\_3\_x - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3\_y - optional; type *double*
- deflection\_check\_reference\_plane\_point\_3\_z - optional; type *double*
- surface\_concrete\_design\_uls\_configuration - optional; type *int*
- surface\_concrete\_design\_sls\_configuration - optional; type *int*

- surface\_concrete\_design\_fr\_configuration - optional; type *int*
- surface\_concrete\_design\_seismic\_configuration - optional; type *int*
- rotated\_boundary\_line - optional; type *int*
- rotated\_angle\_of\_rotation - optional; type *double*
- rotated\_point\_p - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- rotated\_point\_p\_x - optional; type *double*
- rotated\_point\_p\_y - optional; type *double*
- rotated\_point\_p\_z - optional; type *double*
- rotated\_point\_r - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- rotated\_point\_r\_x - optional; type *double*
- rotated\_point\_r\_y - optional; type *double*
- rotated\_point\_r\_z - optional; type *double*
- rotated\_generated\_lines - optional; type *array\_of\_int*
- stress\_analysis\_configuration - optional; type *int*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surfaceResponse*

## 216. set\_surface\_eccentricity

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_eccentricity

**Input:** set\_surface\_eccentricity\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_eccentricity*

- value type *surface\_eccentricity*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - offset - optional; type *double*
  - assigned\_to\_surfaces - optional; type *array\_of\_int*
  - thickness\_alignment - optional; type *surface\_eccentricity\_thickness\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
  - transverse\_offset\_reference\_type - optional; type *surface\_eccentricity\_transverse\_offset\_reference\_type* - type *undefined* with restriction - enum { 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_MEMBER\_SECTION', 'TRANSVERSE\_OFFSET\_TYPE\_FROM\_SURFACE\_THICKNESS', 'TRANSVERSE\_OFFSET\_TYPE\_NONE' }
  - transverse\_offset\_reference\_member - optional; type *int*
  - transverse\_offset\_reference\_surface - optional; type *int*
  - transverse\_offset\_member\_reference\_node - optional; type *int*
  - transverse\_offset\_surface\_reference\_node - optional; type *int*
  - transverse\_offset\_alignment - optional; type *surface\_eccentricity\_transverse\_offset\_alignment* - type *undefined* with restriction - enum { 'ALIGN\_BOTTOM', 'ALIGN\_MIDDLE', 'ALIGN\_TOP' }
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_eccentricity\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_eccentricityResponse*

## 217. set\_surface\_imperfection

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_imperfection

**Input:** set\_surface\_imperfection\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_imperfection*

- imperfection\_case\_no type *int*
- value type *surface\_imperfection*
  - no type *int*
  - definition\_type - optional; type *surface\_imperfection\_definition\_type* - type *undefined* with restriction - enum { 'ABSOLUTE', 'RELATIVE' }
  - imperfection\_case - optional; type *int*
  - imperfection\_direction - optional; type *surface\_imperfection\_imperfection\_direction* - type *undefined* with restriction - enum { 'IMPERFECTION\_DIRECTION\_LOCAL\_Z', 'IMPERFECTION\_DIRECTION\_LOCAL\_Z\_NEGATIVE' }
  - initial\_bow - optional; type *double*
  - initial\_bow\_relative - optional; type *double*
  - parameters - optional; type *array\_of\_int*
  - reference\_length - optional; type *double*
  - surfaces - optional; type *array\_of\_int*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_imperfection\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_imperfectionResponse*

## 218. set\_surface\_load

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_load

**Input:** set\_surface\_load\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_load
 ■ load_case_no type int
 ■ value type surface_load
 ■ no type int
 ■ load_type - optional; type surface_load_load_type - type undefined with restriction - enum {
 'LOAD_TYPE_AXIAL_STRAIN', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_FORM_FINDING', 'LOAD_TYPE_MASS',
 'LOAD_TYPE_PRECAMBER', 'LOAD_TYPE_ROTARY_MOTION', 'LOAD_TYPE_TEMPERATURE' }
 ■ surfaces - optional; type array_of_int
 ■ load_case - optional; type int
 ■ load_distribution - optional; type surface_load_load_distribution - type undefined with restriction - enum {
 'LOAD_DISTRIBUTION_LINEAR', 'LOAD_DISTRIBUTION_LINEAR_IN_X', 'LOAD_DISTRIBUTION_LINEAR_IN_Y',
 'LOAD_DISTRIBUTION_LINEAR_IN_Z', 'LOAD_DISTRIBUTION_RADIAL', 'LOAD_DISTRIBUTION_UNIFORM',
 'LOAD_DISTRIBUTION_VARYING_IN_Z' }
 ■ coordinate_system - optional; type string
 ■ load_direction - optional; type surface_load_load_direction - type undefined with restriction - enum {
 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_PROJECTED',
 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_TRUE',
 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_PROJECTED',
 'LOAD_DIRECTION_GLOBAL_Y_OR_USER_DEFINED_V_TRUE',
 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_PROJECTED',
 'LOAD_DIRECTION_GLOBAL_Z_OR_USER_DEFINED_W_TRUE', 'LOAD_DIRECTION_LOCAL_X',
 'LOAD_DIRECTION_LOCAL_Y', 'LOAD_DIRECTION_LOCAL_Z' }
 ■ uniform_magnitude - optional; type double
 ■ magnitude_1 - optional; type double
 ■ magnitude_2 - optional; type double
 ■ magnitude_3 - optional; type double
 ■ uniform_magnitude_t_c - optional; type double
 ■ magnitude_t_c_1 - optional; type double
 ■ magnitude_t_c_2 - optional; type double
 ■ magnitude_t_c_3 - optional; type double
 ■ uniform_magnitude_delta_t - optional; type double
 ■ magnitude_delta_t_1 - optional; type double
 ■ magnitude_delta_t_2 - optional; type double
 ■ magnitude_delta_t_3 - optional; type double
 ■ magnitude_axial_strain_x - optional; type double
 ■ magnitude_axial_strain_y - optional; type double
 ■ magnitude_axial_strain_1x - optional; type double
 ■ magnitude_axial_strain_1y - optional; type double
 ■ magnitude_axial_strain_2x - optional; type double
 ■ magnitude_axial_strain_2y - optional; type double
 ■ magnitude_axial_strain_3x - optional; type double
 ■ magnitude_axial_strain_3y - optional; type double
 ■ angular_velocity - optional; type double
 ■ angular_acceleration - optional; type double
 ■ node_1 - optional; type int
 ■ node_2 - optional; type int
 ■ node_3 - optional; type int
 ■ axis_definition_type - optional; type surface_load_axis_definition_type - type undefined with restriction - enum {
 'AXIS_DEFINITION_POINT_AND_AXIS', 'AXIS_DEFINITION_TWO_POINTS' }
 ■ axis_definition_p1 - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ axis_definition_p1_x - optional; type double
 ■ axis_definition_p1_y - optional; type double
 ■ axis_definition_p1_z - optional; type double
 ■ axis_definition_p2 - optional; type vector_3d
 ■ x type double
 ■ y type double
 ■ z type double
 ■ axis_definition_p2_x - optional; type double
 ■ axis_definition_p2_y - optional; type double
 ■ axis_definition_p2_z - optional; type double
 ■ axis_definition_axis - optional; type surface_load_axis_definition_axis - type undefined with restriction - enum {
 'AXIS_X', 'AXIS_Y', 'AXIS_Z' }
 ■ axis_definition_axis_orientation - optional; type surface_load_axis_definition_axis_orientation - type undefined
 with restriction - enum { 'AXIS_NEGATIVE', 'AXIS_POSITIVE' }
 ■ varying_load_parameters - optional; type array_of_surface_load_varying_load_parameters
 ■ surface_load_varying_load_parameters - optional, unbounded; type
 surface_load_varying_load_parameters
 ■ no - optional; type int
 ■ distance - optional; type double
 ■ delta_distance - optional; type double
 ■ magnitude - optional; type double
 ■ note - optional; type string
 ■ varying_load_parameters_sorted - optional; type boolean
 ■ form_finding_definition - optional; type surface_load_form_finding_definition - type undefined with restriction -
 enum { 'FORM_FINDING_DEFINITION_FORCE', 'FORM_FINDING_DEFINITION_STRESS' }
 ■ magnitude_force_x - optional; type double
 ■ magnitude_force_y - optional; type double
 ■ magnitude_force_u - optional; type double
 ■ magnitude_force_v - optional; type double
 ■ magnitude_force_r - optional; type double
 ■ magnitude_force_t - optional; type double
 ■ magnitude_stress_x - optional; type double
 ■ magnitude_stress_y - optional; type double
 ■ magnitude_stress_u - optional; type double
 ■ magnitude_stress_v - optional; type double
 ■ magnitude_stress_r - optional; type double
 ■ magnitude_stress_t - optional; type double
```

- form\_finding\_calculation\_method - optional; type *surface\_load\_form\_finding\_calculation\_method* - type *undefined* with restriction - enum { 'FORM\_FINDING\_CALCULATION\_METHOD\_PROJECTION', 'FORM\_FINDING\_CALCULATION\_METHOD\_STANDARD' }
- individual\_mass\_components - optional; type *boolean*
- magnitude\_mass\_global - optional; type *double*
- magnitude\_mass\_y - optional; type *double*
- magnitude\_mass\_z - optional; type *double*
- magnitude\_mass\_x - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_load\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_loadResponse*

## 219. set\_surface\_mesh\_refinement

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_mesh\_refinement

**Input:** set\_surface\_mesh\_refinement\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_surface\_mesh\_refinement*
- value type *surface\_mesh\_refinement*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - surfaces - optional; type *array\_of\_int*
    - target\_length - optional; type *double*
    - comment - optional; type *string*
    - is\_generated - optional; type *boolean*
    - generating\_object\_info - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_mesh\_refinement\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_mesh\_refinementResponse*

## 220. set\_surface\_results\_adjustment

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_results\_adjustment

**Input:** set\_surface\_results\_adjustment\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_surface\_results\_adjustment*
- value type *surface\_results\_adjustment*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - surfaces - optional; type *array\_of\_int*
    - shape - optional; type *surface\_results\_adjustment\_shape* - type *undefined* with restriction - enum { 'SHAPE\_CIRCLE', 'SHAPE\_ELLIPSE', 'SHAPE\_RECTANGLE' }
    - dimension\_1 - optional; type *double*
    - dimension\_2 - optional; type *double*
    - angular\_rotation - optional; type *double*
    - center\_position - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - center\_position\_x - optional; type *double*
    - center\_position\_y - optional; type *double*
    - center\_position\_z - optional; type *double*
    - adjustment\_type\_in\_direction\_u - optional; type *surface\_results\_adjustment\_adjustment\_type\_in\_direction\_u* - type *undefined* with restriction - enum { 'AVERAGING\_OF\_MX\_MXY\_VX\_NX\_NXY', 'AVERAGING\_OF\_MY\_MXY\_VY\_NY\_NXY', 'NONE', 'USER\_DEFINED', 'ZERO' }
    - adjustment\_type\_in\_direction\_v - optional; type *surface\_results\_adjustment\_adjustment\_type\_in\_direction\_v* - type *undefined* with restriction - enum { 'AVERAGING\_OF\_MX\_MXY\_VX\_NX\_NXY', 'AVERAGING\_OF\_MY\_MXY\_VY\_NY\_NXY', 'NONE', 'USER\_DEFINED', 'ZERO' }
    - projection\_in\_direction\_type - optional; type *surface\_results\_adjustment\_projection\_in\_direction\_type* - type *undefined* with restriction - enum { 'GLOBAL\_IN\_X', 'GLOBAL\_IN\_Y', 'GLOBAL\_IN\_Z', 'PERPENDICULAR', 'VECTOR' }
    - vector\_of\_projection\_in\_direction\_coordinates - optional; type *vector\_3d*
      - x type *double*
      - y type *double*
      - z type *double*
    - vector\_of\_projection\_in\_direction\_coordinates\_x - optional; type *double*
    - vector\_of\_projection\_in\_direction\_coordinates\_y - optional; type *double*
    - vector\_of\_projection\_in\_direction\_coordinates\_z - optional; type *double*
    - comment - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_results\_adjustment\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_results\_adjustmentResponse*

## 221. set\_surface\_set

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_set

**Input:** set\_surface\_set\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_set
 value type surface_set
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 set_type - optional; type surface_set_set_type - type undefined with restriction - enum { 'SET_TYPE_CONTINUOUS', 'SET_TYPE_GROUP' }
 surfaces - optional; type array_of_int
 surface_area - optional; type double
 volume - optional; type double
 mass - optional; type double
 center_of_gravity - optional; type vector_3d
 x type double
 y type double
 z type double
 center_of_gravity_x - optional; type double
 center_of_gravity_y - optional; type double
 center_of_gravity_z - optional; type double
 position - optional; type string
 position_short - optional; type string
 stress_analysis_configuration - optional; type int
 timber_service_class - optional; type int
 timber_moisture_class - optional; type int
 timber_service_conditions - optional; type int
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 surface_concrete_design_uls_configuration - optional; type int
 surface_concrete_design_sls_configuration - optional; type int
 surface_concrete_design_fr_configuration - optional; type int
 surface_concrete_design_seismic_configuration - optional; type int
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_surface\_set\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_setResponse
```

## 222. set\_surface\_set\_imperfection

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_set\_imperfection

**Input:** set\_surface\_set\_imperfection\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_set_imperfection
 imperfection_case_no type int
 value type surface_set_imperfection
 no type int
 definition_type - optional; type surface_set_imperfection_definition_type - type undefined with restriction - enum { 'ABSOLUTE', 'RELATIVE' }
 imperfection_case - optional; type int
 imperfection_direction - optional; type surface_set_imperfection_imperfection_direction - type undefined with restriction - enum { 'IMPERFECTION_DIRECTION_LOCAL_Z', 'IMPERFECTION_DIRECTION_LOCAL_Z_NEGATIVE' }
 initial_bow - optional; type double
 initial_bow_relative - optional; type double
 parameters - optional; type array_of_int
 reference_length - optional; type double
 surface_sets - optional; type array_of_int
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_surface\_set\_imperfection\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_set_imperfectionResponse
```

## 223. set\_surface\_set\_load

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_set\_load

**Input:** set\_surface\_set\_load\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_set_load
 load_case_no type int
 value type surface_set_load
 no type int
 load_type - optional; type surface_set_load_load_type - type undefined with restriction - enum { 'LOAD_TYPE_AXIAL_STRAIN', 'LOAD_TYPE_FORCE', 'LOAD_TYPE_FORM_FINDING', 'LOAD_TYPE_MASS', 'LOAD_TYPE_PRECAMBER', 'LOAD_TYPE_ROTARY_MOTION', 'LOAD_TYPE_TEMPERATURE' }
 surface_sets - optional; type array_of_int
 load_case - optional; type int
 coordinate_system - optional; type string
 load_distribution - optional; type surface_set_load_load_distribution - type undefined with restriction - enum { 'LOAD_DISTRIBUTION_LINEAR', 'LOAD_DISTRIBUTION_LINEAR_IN_X', 'LOAD_DISTRIBUTION_LINEAR_IN_Y', 'LOAD_DISTRIBUTION_LINEAR_IN_Z', 'LOAD_DISTRIBUTION_RADIAL', 'LOAD_DISTRIBUTION_UNIFORM', 'LOAD_DISTRIBUTION_VARYING_IN_Z' }
 load_direction - optional; type surface_set_load_load_direction - type undefined with restriction - enum { 'LOAD_DIRECTION_GLOBAL_X_OR_USER_DEFINED_U_PROJECTED',
```

'LOAD\_DIRECTION\_GLOBAL\_X\_OR\_USER\_DEFINED\_U\_TRUE',  
'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_PROJECTED',  
'LOAD\_DIRECTION\_GLOBAL\_Y\_OR\_USER\_DEFINED\_V\_TRUE',  
'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_PROJECTED',  
'LOAD\_DIRECTION\_GLOBAL\_Z\_OR\_USER\_DEFINED\_W\_TRUE', 'LOAD\_DIRECTION\_LOCAL\_X',  
'LOAD\_DIRECTION\_LOCAL\_Y', 'LOAD\_DIRECTION\_LOCAL\_Z' }

- individual\_mass\_components - optional; type *boolean*
- uniform\_magnitude - optional; type *double*
- magnitude\_1 - optional; type *double*
- magnitude\_2 - optional; type *double*
- magnitude\_3 - optional; type *double*
- uniform\_magnitude\_t\_c - optional; type *double*
- magnitude\_t\_c\_1 - optional; type *double*
- magnitude\_t\_c\_2 - optional; type *double*
- magnitude\_t\_c\_3 - optional; type *double*
- uniform\_magnitude\_delta\_t - optional; type *double*
- magnitude\_delta\_t\_1 - optional; type *double*
- magnitude\_delta\_t\_2 - optional; type *double*
- magnitude\_delta\_t\_3 - optional; type *double*
- magnitude\_axial\_strain\_x - optional; type *double*
- magnitude\_axial\_strain\_y - optional; type *double*
- magnitude\_axial\_strain\_1x - optional; type *double*
- magnitude\_axial\_strain\_1y - optional; type *double*
- magnitude\_axial\_strain\_2x - optional; type *double*
- magnitude\_axial\_strain\_2y - optional; type *double*
- magnitude\_axial\_strain\_3x - optional; type *double*
- magnitude\_axial\_strain\_3y - optional; type *double*
- angular\_velocity - optional; type *double*
- angular\_acceleration - optional; type *double*
- node\_1 - optional; type *int*
- node\_2 - optional; type *int*
- node\_3 - optional; type *int*
- axis\_definition\_type - optional; type *surface\_set\_load\_axis\_definition\_type* - type *undefined* with restriction - enum { 'AXIS\_DEFINITION\_POINT\_AND\_AXIS', 'AXIS\_DEFINITION\_TWO\_POINTS' }
- axis\_definition\_p1 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p1\_x - optional; type *double*
- axis\_definition\_p1\_y - optional; type *double*
- axis\_definition\_p1\_z - optional; type *double*
- axis\_definition\_p2 - optional; type *vector\_3d*
  - x type *double*
  - y type *double*
  - z type *double*
- axis\_definition\_p2\_x - optional; type *double*
- axis\_definition\_p2\_y - optional; type *double*
- axis\_definition\_p2\_z - optional; type *double*
- axis\_definition\_axis - optional; type *surface\_set\_load\_axis\_definition\_axis* - type *undefined* with restriction - enum { 'AXIS\_X', 'AXIS\_Y', 'AXIS\_Z' }
- axis\_definition\_axis\_orientation - optional; type *surface\_set\_load\_axis\_definition\_axis\_orientation* - type *undefined* with restriction - enum { 'AXIS\_NEGATIVE', 'AXIS\_POSITIVE' }
- varying\_load\_parameters - optional; type *array\_of\_surface\_set\_load\_varying\_load\_parameters*
  - surface\_set\_load\_varying\_load\_parameters - optional, unbounded; type *surface\_set\_load\_varying\_load\_parameters*
    - no - optional; type *int*
    - distance - optional; type *double*
    - delta\_distance - optional; type *double*
    - magnitude - optional; type *double*
    - note - optional; type *string*
- varying\_load\_parameters\_sorted - optional; type *boolean*
- form\_finding\_definition - optional; type *surface\_set\_load\_form\_finding\_definition* - type *undefined* with restriction - enum { 'FORM\_FINDING\_DEFINITION\_FORCE', 'FORM\_FINDING\_DEFINITION\_STRESS' }
- magnitude\_force\_x - optional; type *double*
- magnitude\_force\_y - optional; type *double*
- magnitude\_force\_u - optional; type *double*
- magnitude\_force\_v - optional; type *double*
- magnitude\_force\_r - optional; type *double*
- magnitude\_force\_t - optional; type *double*
- magnitude\_mass\_x - optional; type *double*
- magnitude\_mass\_y - optional; type *double*
- magnitude\_mass\_z - optional; type *double*
- magnitude\_stress\_x - optional; type *double*
- magnitude\_stress\_y - optional; type *double*
- magnitude\_stress\_u - optional; type *double*
- magnitude\_stress\_v - optional; type *double*
- magnitude\_stress\_r - optional; type *double*
- magnitude\_stress\_t - optional; type *double*
- form\_finding\_calculation\_method - optional; type *surface\_set\_load\_form\_finding\_calculation\_method* - type *undefined* with restriction - enum { 'FORM\_FINDING\_CALCULATION\_METHOD\_PROJECTION', 'FORM\_FINDING\_CALCULATION\_METHOD\_STANDARD' }
- magnitude\_mass\_global - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_surface\_set\_load\_response (soap:body, use = literal) [Source code](#)

parameters type [set\\_surface\\_set\\_loadResponse](#)

## 224. set\_surface\_stiffness\_modification

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_stiffness\_modification

**Input:** set\_surface\_stiffness\_modification\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_stiffness_modification
 value type surface_stiffness_modification
 no type int
 type - optional; type surface_stiffness_modification_type - type undefined with restriction - enum {
 'TYPE_CONCRETE_STRUCTURES_ACI', 'TYPE_CONCRETE_STRUCTURES_CSA',
 'TYPE_PARTIAL_STIFFNESSES_FACTORS', 'TYPE_STIFFNESS_MATRIX_ELEMENTS_FACTORS',
 'TYPE_TOTAL_STIFFNESS_FACTOR' }
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 factor_of_total_stiffness - optional; type double
 factor_of_bending_stiffness - optional; type double
 factor_of_shear_stiffness - optional; type double
 factor_of_membrane_stiffness - optional; type double
 factor_of_eccentric_effects - optional; type double
 factor_of_weight - optional; type double
 kd11 - optional; type double
 kd12 - optional; type double
 kd13 - optional; type double
 kd22 - optional; type double
 kd23 - optional; type double
 kd33 - optional; type double
 kd44 - optional; type double
 kd45 - optional; type double
 kd55 - optional; type double
 kd66 - optional; type double
 kd67 - optional; type double
 kd68 - optional; type double
 kd77 - optional; type double
 kd78 - optional; type double
 kd88 - optional; type double
 kd16 - optional; type double
 kd17 - optional; type double
 kd18 - optional; type double
 kd27 - optional; type double
 kd28 - optional; type double
 kd38 - optional; type double
 kd11_note - optional; type string
 kd12_note - optional; type string
 kd13_note - optional; type string
 kd22_note - optional; type string
 kd23_note - optional; type string
 kd33_note - optional; type string
 kd44_note - optional; type string
 kd45_note - optional; type string
 kd55_note - optional; type string
 kd66_note - optional; type string
 kd67_note - optional; type string
 kd68_note - optional; type string
 kd77_note - optional; type string
 kd78_note - optional; type string
 kd88_note - optional; type string
 kd16_note - optional; type string
 kd17_note - optional; type string
 kd18_note - optional; type string
 kd27_note - optional; type string
 kd28_note - optional; type string
 kd38_note - optional; type string
 assigned_to_structural_modification - optional; type array_of_int
 concrete_structures_component_type - optional; type
 surface_stiffness_modification_concrete_structures_component_type - type undefined with restriction -
 enum { 'COMPONENT_TYPE_BEAMS', 'COMPONENT_TYPE_COLUMNS',
 'COMPONENT_TYPE_FLAT_PLATES_AND_FLAT_SLABS', 'COMPONENT_TYPE_WALLS_CRACKED',
 'COMPONENT_TYPE_WALLS_UNCRACKED' }
 concrete_structures_bending_stiffness_factor - optional; type double
 comment - optional; type string
 is_generated - optional; type boolean
 generating_object_info - optional; type string
 id_for_export_import - optional; type string
 metadata_for_export_import - optional; type string
```

**Output:** set\_surface\_stiffness\_modification\_response (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_stiffness_modificationResponse
```

## 225. set\_surface\_support

[Source code](#)

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surface\_support

**Input:** set\_surface\_support\_request (soap:body, use = literal) [Source code](#)

```
parameters type set_surface_support
 value type surface_support
 no type int
 user_defined_name_enabled - optional; type boolean
 name - optional; type string
 surfaces - optional; type array_of_int
 translation - optional; type vector_3d
 x type double
 y type double
 z type double
 translation_x - optional; type double
```



- translation\_y - optional; type *double*
- translation\_z - optional; type *double*
- shear\_xz - optional; type *double*
- shear\_yz - optional; type *double*
- nonlinearity - optional; type *surface\_support\_nonlinearity* - type *undefined* with restriction - enum { 'NONLINEARITY\_FAILURE\_IF\_NEGATIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_FAILURE\_IF\_POSITIVE\_CONTACT\_STRESS\_Z', 'NONLINEARITY\_NONE' }
- negative\_nonlinearity\_type - optional; type *surface\_support\_negative\_nonlinearity\_type* - type *undefined* with restriction - enum { 'NONLINEARITY\_DEFINITION\_TYPE\_BASIC\_UNDIRECTIONAL\_ACTION', 'NONLINEARITY\_DEFINITION\_TYPE\_FRICTION\_PLANE\_XY', 'NONLINEARITY\_DEFINITION\_TYPE\_YIELDING\_CONTACT\_STRESS\_SIGMA\_Z' }
- positive\_nonlinearity\_type - optional; type *surface\_support\_positive\_nonlinearity\_type* - type *undefined* with restriction - enum { 'NONLINEARITY\_DEFINITION\_TYPE\_BASIC\_UNDIRECTIONAL\_ACTION', 'NONLINEARITY\_DEFINITION\_TYPE\_FRICTION\_PLANE\_XY', 'NONLINEARITY\_DEFINITION\_TYPE\_YIELDING\_CONTACT\_STRESS\_SIGMA\_Z' }
- negative\_friction\_coefficient - optional; type *double*
- positive\_friction\_coefficient - optional; type *double*
- negative\_contact\_stress - optional; type *double*
- positive\_contact\_stress - optional; type *double*
- comment - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surface\_support\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surface\_supportResponse*

## 226. set\_surfaces\_contact

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surfaces\_contact

**Input:** set\_surfaces\_contact\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_surfaces\_contact*
- value type *surfaces\_contact*
    - no type *int*
    - surfaces\_contact\_type - optional; type *int*
    - surfaces\_group1 - optional; type *array\_of\_int*
    - surfaces\_group2 - optional; type *array\_of\_int*
    - comment - optional; type *string*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surfaces\_contact\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surfaces\_contactResponse*

## 227. set\_surfaces\_contact\_type

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_surfaces\_contact\_type

**Input:** set\_surfaces\_contact\_type\_request (soap:body, use = literal) [Source code](#)

- parameters type *set\_surfaces\_contact\_type*
- value type *surfaces\_contact\_type*
    - no type *int*
    - user\_defined\_name\_enabled - optional; type *boolean*
    - name - optional; type *string*
    - comment - optional; type *string*
    - surfaces\_contacts - optional; type *array\_of\_int*
    - perpendicular\_to\_surface - optional; type *surfaces\_contact\_type\_perpendicular\_to\_surface* - type *undefined* with restriction - enum { 'FAILURE\_UNDER\_COMPRESSION', 'FAILURE\_UNDER\_TENSION', 'FULL\_FORCE\_TRANSMISSION' }
    - parallel\_to\_surface - optional; type *surfaces\_contact\_type\_parallel\_to\_surface* - type *undefined* with restriction - enum { 'ELASTIC\_FRICTION', 'ELASTIC\_SURFACE', 'FAILURE\_IF\_CONTACT\_PERPENDICULAR\_TO\_SURFACES\_FAILED', 'FULL\_FORCE\_TRANSMISSION', 'RIGID\_FRICTION' }
    - rigid\_friction\_type - optional; type *surfaces\_contact\_type\_rigid\_friction\_type* - type *undefined* with restriction - enum { 'FRICTION\_COEFFICIENT', 'LIMIT\_STRESS' }
    - rigid\_friction\_coefficient - optional; type *double*
    - rigid\_friction\_limit\_stress - optional; type *double*
    - elastic\_behavior\_shear\_stiffness - optional; type *double*
    - elastic\_friction\_type - optional; type *surfaces\_contact\_type\_elastic\_friction\_type* - type *undefined* with restriction - enum { 'FRICTION\_COEFFICIENT', 'LIMIT\_STRESS' }
    - elastic\_friction\_coefficient - optional; type *double*
    - elastic\_friction\_limit\_stress - optional; type *double*
    - elastic\_friction\_shear\_stiffness - optional; type *double*
    - id\_for\_export\_import - optional; type *string*
    - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_surfaces\_contact\_type\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_surfaces\_contact\_typeResponse*

## 228. set\_terrain

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_terrain

**Input:** set\_terrain\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_terrain*

- value type *terrain*
  - no type *int*
  - type - optional; type *terrain\_type* - type *undefined* with restriction - enum { 'HORIZONTAL\_PLANE', 'INCLINED\_PLANE', 'NO\_TERRAIN', 'SOIL\_SAMPLES', 'TABLE' }
  - comment - optional; type *string*
  - bounding\_box\_offset\_x - optional; type *double*
  - bounding\_box\_offset\_y - optional; type *double*
  - center\_of\_terrain\_z - optional; type *double*
  - rotation\_around\_Z - optional; type *double*
  - consider\_soil\_samples - optional; type *boolean*
  - coordinate\_system - optional; type *int*
  - terrain\_table - optional; type *array\_of\_terrain\_terrain\_table*
    - terrain\_terrain\_table - optional, unbounded; type *terrain\_terrain\_table*
      - no - optional; type *int*
      - global\_x - optional; type *double*
      - global\_y - optional; type *double*
      - global\_z - optional; type *double*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

Output: set\_terrain\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_terrainResponse*

## 229. set\_thickness

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_thickness

Input: set\_thickness\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_thickness*

- value type *thickness*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - assigned\_to\_surfaces - optional; type *array\_of\_int*
  - type - optional; type *thickness\_type* - type *undefined* with restriction - enum { 'TYPE\_LAYERS', 'TYPE\_SHAPE\_ORTHOTROPY', 'TYPE\_STIFFNESS\_MATRIX', 'TYPE\_THICKNESS\_PHASE', 'TYPE\_UNIFORM', 'TYPE\_VARIABLE\_CIRCLE', 'TYPE\_VARIABLE\_FOUR\_SURFACE\_CORNERS', 'TYPE\_VARIABLE\_THREE\_NODES', 'TYPE\_VARIABLE\_TWO\_NODES\_AND\_DIRECTION' }
  - material - optional; type *int*
  - uniform\_thickness - optional; type *double*
  - advanced\_time\_dependent\_properties\_of\_concrete\_enabled - optional; type *boolean*
  - creep\_enabled - optional; type *boolean*
  - shrinkage\_enabled - optional; type *boolean*
  - relative\_humidity - optional; type *double*
  - thickness\_1 - optional; type *double*
  - node\_1 - optional; type *int*
  - thickness\_2 - optional; type *double*
  - node\_2 - optional; type *int*
  - thickness\_3 - optional; type *double*
  - node\_3 - optional; type *int*
  - direction - optional; type *thickness\_direction* - type *undefined* with restriction - enum { 'THICKNESS\_DIRECTION\_IN\_SMALL\_X', 'THICKNESS\_DIRECTION\_IN\_SMALL\_Y', 'THICKNESS\_DIRECTION\_IN\_X', 'THICKNESS\_DIRECTION\_IN\_Y', 'THICKNESS\_DIRECTION\_IN\_Z' }
  - thickness\_4 - optional; type *double*
  - node\_4 - optional; type *int*
  - thickness\_circle\_center - optional; type *double*
  - thickness\_circle\_line - optional; type *double*
  - comment - optional; type *string*
  - layers\_reference\_table - optional; type *array\_of\_thickness\_layers\_reference\_table*
    - thickness\_layers\_reference\_table - optional, unbounded; type *thickness\_layers\_reference\_table*
      - no - optional; type *int*
      - layer\_no - optional; type *int*
      - layer\_type - optional; type *layer\_type* - type *undefined* with restriction
      - thickness\_type - optional; type *int*
      - material - optional; type *int*
      - thickness - optional; type *double*
      - angle - optional; type *double*
      - connection\_with\_other\_topological\_elements - optional; type *boolean*
      - comment - optional; type *string*
      - specific\_weight - optional; type *double*
      - weight - optional; type *double*
  - layers\_solid\_model\_enabled - optional; type *boolean*
  - layers\_gas\_enabled - optional; type *boolean*
  - layers\_total\_thickness - optional; type *double*
  - layers\_total\_weight - optional; type *double*
  - stiffness\_reduction\_enabled - optional; type *boolean*
  - K33 - optional; type *double*
  - K44 - optional; type *double*
  - K55 - optional; type *double*
  - K88 - optional; type *double*
  - K33\_note - optional; type *string*
  - K44\_note - optional; type *string*
  - K55\_note - optional; type *string*
  - K88\_note - optional; type *string*
  - function\_data\_function\_type - optional; type *thickness\_function\_data\_function\_type* - type *undefined* with restriction - enum { 'FUNCTION\_TYPE\_CREEP\_COEFFICIENT', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CA', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CD', 'FUNCTION\_TYPE\_SHRINKAGE\_STRAIN\_CS' }
  - function\_data\_layer\_id - optional; type *int*
  - function\_data\_age\_of\_concrete\_at\_the\_considered\_moment - optional; type *double*
  - function\_data\_number\_of\_steps - optional; type *int*
  - function\_data\_coefficients - optional; type *array\_of\_thickness\_function\_data\_coefficients*

- thickness\_function\_data\_coefficients - optional, unbounded; type *thickness\_function\_data\_coefficients*
  - no - optional; type *int*
  - time - optional; type *double*
  - coefficient - optional; type *double*
- parent\_thickness - optional; type *int*
- parent\_thickness\_layers - optional; type *array\_of\_int*
- timber\_frame\_wall\_sheathing\_front\_material\_enabled - optional; type *boolean*
- timber\_frame\_wall\_sheathing\_front\_material - optional; type *int*
- timber\_frame\_wall\_sheathing\_front\_thickness - optional; type *double*
- timber\_frame\_wall\_sheathing\_back\_material\_enabled - optional; type *boolean*
- timber\_frame\_wall\_sheathing\_back\_material - optional; type *int*
- timber\_frame\_wall\_sheathing\_back\_thickness - optional; type *double*
- timber\_frame\_wall\_horizontal\_framing\_members\_cross\_section - optional; type *int*
- timber\_frame\_wall\_vertical\_framing\_members\_cross\_section - optional; type *int*
- timber\_frame\_wall\_framing\_members\_connector\_type - optional; type *thickness\_timber\_frame\_wall\_framing\_members\_connector\_type* - type *undefined* with restriction - enum { 'TYPE\_NAIL', 'TYPE\_STAPLE' }
- timber\_frame\_wall\_framing\_members\_connector\_diameter - optional; type *double*
- timber\_frame\_wall\_framing\_members\_connector\_dimension\_type - optional; type *thickness\_timber\_frame\_wall\_framing\_members\_connector\_dimension\_type* - type *undefined* with restriction - enum { 'TYPE\_0075\_X\_1060', 'TYPE\_0075\_X\_1130', 'TYPE\_0075\_X\_1300', 'TYPE\_0110\_X\_0610', 'TYPE\_0120\_X\_0200', 'TYPE\_0125\_X\_0620', 'TYPE\_0125\_X\_0770', 'TYPE\_0125\_X\_1060', 'TYPE\_0125\_X\_1130', 'TYPE\_0125\_X\_1250', 'TYPE\_0125\_X\_1300', 'TYPE\_0165\_X\_0280' }
- timber\_frame\_wall\_framing\_members\_connector\_spacing - optional; type *double*
- timber\_frame\_wall\_vertical\_studs\_cross\_section - optional; type *int*
- timber\_frame\_wall\_vertical\_studs\_distribution - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_distribution* - type *undefined* with restriction - enum { 'TYPE\_CENTER\_TO\_CENTER', 'TYPE\_GAP', 'TYPE\_UNIFORM', 'TYPE\_USER\_DEFINED' }
- timber\_frame\_wall\_vertical\_studs\_reverse\_distribution\_enabled - optional; type *boolean*
- timber\_frame\_wall\_vertical\_studs\_spacing - optional; type *double*
- timber\_frame\_wall\_vertical\_studs\_spacing\_table - optional; type *array\_of\_thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table*
  - thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table - optional, unbounded; type *thickness\_timber\_frame\_wall\_vertical\_studs\_spacing\_table*
    - no - optional; type *int*
    - spacing - optional; type *double*
    - note - optional; type *string*
- timber\_frame\_wall\_vertical\_studs\_spacing\_definition\_relative - optional; type *boolean*
- timber\_frame\_wall\_vertical\_studs\_connector\_type - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_connector\_type* - type *undefined* with restriction - enum { 'TYPE\_NAIL', 'TYPE\_STAPLE' }
- timber\_frame\_wall\_vertical\_studs\_connector\_diameter - optional; type *double*
- timber\_frame\_wall\_vertical\_studs\_connector\_dimension\_type - optional; type *thickness\_timber\_frame\_wall\_vertical\_studs\_connector\_dimension\_type* - type *undefined* with restriction - enum { 'TYPE\_0075\_X\_1060', 'TYPE\_0075\_X\_1130', 'TYPE\_0075\_X\_1300', 'TYPE\_0110\_X\_0610', 'TYPE\_0120\_X\_0200', 'TYPE\_0125\_X\_0620', 'TYPE\_0125\_X\_0770', 'TYPE\_0125\_X\_1060', 'TYPE\_0125\_X\_1130', 'TYPE\_0125\_X\_1250', 'TYPE\_0125\_X\_1300', 'TYPE\_0165\_X\_0280' }
- timber\_frame\_wall\_vertical\_studs\_connector\_spacing - optional; type *double*
- timber\_frame\_wall\_blockings\_cross\_section\_enabled - optional; type *boolean*
- timber\_frame\_wall\_blockings\_cross\_section - optional; type *int*
- timber\_frame\_wall\_blockings\_distribution - optional; type *thickness\_timber\_frame\_wall\_blockings\_distribution* - type *undefined* with restriction - enum { 'TYPE\_ALTERNATING', 'TYPE\_INLINE', 'TYPE\_STAGGERED' }
- timber\_frame\_wall\_blockings\_offset - optional; type *double*
- orthotropy\_type - optional; type *thickness\_orthotropy\_type* - type *undefined* with restriction - enum { 'ORTHOTROPIC\_THICKNESS\_TYPE\_BIDIRECTIONAL\_RIBBED\_PLATE', 'ORTHOTROPIC\_THICKNESS\_TYPE\_COUPLING', 'ORTHOTROPIC\_THICKNESS\_TYPE\_EFFECTIVE\_THICKNESS', 'ORTHOTROPIC\_THICKNESS\_TYPE\_GRILLAGE', 'ORTHOTROPIC\_THICKNESS\_TYPE\_HOLLOW\_CORE\_SLAB', 'ORTHOTROPIC\_THICKNESS\_TYPE\_TRAPEZOIDAL\_SHEET', 'ORTHOTROPIC\_THICKNESS\_TYPE\_UNIDIRECTIONAL\_RIBBED\_PLATE' }
- orthotropy\_rotation\_beta - optional; type *double*
- orthotropy\_fictitious\_thickness - optional; type *double*
- shape\_orthotropy\_self\_weight\_definition\_type - optional; type *thickness\_shape\_orthotropy\_self\_weight\_definition\_type* - type *undefined* with restriction - enum { 'SELF\_WEIGHT\_COMPUTED\_FROM\_PARAMETERS', 'SELF\_WEIGHT\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS', 'SELF\_WEIGHT\_DEFINED\_VIA\_WEIGHT' }
- shape\_orthotropy\_effective\_thickness\_x - optional; type *double*
- shape\_orthotropy\_effective\_thickness\_y - optional; type *double*
- shape\_orthotropy\_self\_weight - optional; type *double*
- stiffness\_matrix\_self\_weight\_definition\_type - optional; type *thickness\_stiffness\_matrix\_self\_weight\_definition\_type* - type *undefined* with restriction - enum { 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_BULK\_DENSITY\_AND\_AREA\_DENSITY', 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS\_AND\_AREA\_DENSITY', 'SELF\_WEIGHT\_DEFINITION\_TYPE\_DEFINED\_VIA\_FICTITIOUS\_THICKNESS\_AND\_BULK\_DENSITY' }
- stiffness\_matrix\_bulk\_density - optional; type *double*
- stiffness\_matrix\_area\_density - optional; type *double*
- stiffness\_matrix\_coefficient\_of\_thermal\_expansion - optional; type *double*
- coupling\_thickness - optional; type *double*
- coupling\_spacing - optional; type *double*
- coupling\_width - optional; type *double*
- slab\_thickness - optional; type *double*
- rib\_height - optional; type *double*
- rib\_spacing - optional; type *double*
- rib\_width - optional; type *double*
- rib\_spacing\_x - optional; type *double*
- rib\_spacing\_y - optional; type *double*
- rib\_width\_x - optional; type *double*
- rib\_width\_y - optional; type *double*
- sheet\_thickness - optional; type *double*
- total\_profile\_height - optional; type *double*
- top\_flange\_width - optional; type *double*
- bottom\_flange\_width - optional; type *double*
- void\_spacing - optional; type *double*
- void\_diameter - optional; type *double*
- rib\_height\_x - optional; type *double*
- rib\_height\_y - optional; type *double*
- D11 - optional; type *double*
- D11\_note - optional; type *string*

- D12 - optional; type *double*
- D12\_note - optional; type *string*
- D13 - optional; type *double*
- D13\_note - optional; type *string*
- D16 - optional; type *double*
- D16\_note - optional; type *string*
- D17 - optional; type *double*
- D17\_note - optional; type *string*
- D18 - optional; type *double*
- D18\_note - optional; type *string*
- D22 - optional; type *double*
- D22\_note - optional; type *string*
- D23 - optional; type *double*
- D23\_note - optional; type *string*
- D27 - optional; type *double*
- D27\_note - optional; type *string*
- D28 - optional; type *double*
- D28\_note - optional; type *string*
- D33 - optional; type *double*
- D33\_note - optional; type *string*
- D38 - optional; type *double*
- D38\_note - optional; type *string*
- D44 - optional; type *double*
- D44\_note - optional; type *string*
- D45 - optional; type *double*
- D45\_note - optional; type *string*
- D55 - optional; type *double*
- D55\_note - optional; type *string*
- D66 - optional; type *double*
- D66\_note - optional; type *string*
- D67 - optional; type *double*
- D67\_note - optional; type *string*
- D68 - optional; type *double*
- D68\_note - optional; type *string*
- D77 - optional; type *double*
- D77\_note - optional; type *string*
- D78 - optional; type *double*
- D78\_note - optional; type *string*
- D88 - optional; type *double*
- D88\_note - optional; type *string*
- is\_generated - optional; type *boolean*
- generating\_object\_info - optional; type *string*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_thickness\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_thicknessResponse*

### 230. set\_visual\_object

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_visual\_object

**Input:** set\_visual\_object\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_visual\_object*

- value type *visual\_object*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - filename - optional; type *string*
  - coordinate\_system - optional; type *int*
  - insert\_point - optional; type *visual\_object\_insert\_point* - type *undefined* with restriction - enum { 'INSERT\_CENTER', 'INSERT\_MX', 'INSERT\_MXMYMZ', 'INSERT\_MXMYPZ', 'INSERT\_MXPYMZ', 'INSERT\_MXPYPZ', 'INSERT\_MY', 'INSERT\_MZ', 'INSERT\_PX', 'INSERT\_PXMYMZ', 'INSERT\_PXMYPZ', 'INSERT\_PXPYMZ', 'INSERT\_PXPYPZ', 'INSERT\_PY', 'INSERT\_PZ' }
  - origin\_coordinates - optional; type *vector\_3d*
    - x type *double*
    - y type *double*
    - z type *double*
  - origin\_coordinate\_x - optional; type *double*
  - origin\_coordinate\_y - optional; type *double*
  - origin\_coordinate\_z - optional; type *double*
  - rotation\_angles\_sequence - optional; type *visual\_object\_rotation\_angles\_sequence* - type *undefined* with restriction - enum { 'SEQUENCE\_UVW', 'SEQUENCE\_UWV', 'SEQUENCE\_VUW', 'SEQUENCE\_VWU', 'SEQUENCE\_WUV', 'SEQUENCE\_WVU', 'SEQUENCE\_XYZ', 'SEQUENCE\_XZY', 'SEQUENCE\_XXZ', 'SEQUENCE\_YZX', 'SEQUENCE\_ZXY', 'SEQUENCE\_ZYX' }
  - rotation\_angle\_1 - optional; type *double*
  - rotation\_angle\_2 - optional; type *double*
  - rotation\_angle\_3 - optional; type *double*
  - scale\_is\_nonuniform - optional; type *boolean*
  - scale\_is\_defined\_as\_relative - optional; type *boolean*
  - scale\_absolute - optional; type *double*
  - scale\_relative - optional; type *double*
  - scale\_absolute\_x - optional; type *double*
  - scale\_absolute\_y - optional; type *double*
  - scale\_absolute\_z - optional; type *double*
  - scale\_relative\_x - optional; type *double*
  - scale\_relative\_y - optional; type *double*
  - scale\_relative\_z - optional; type *double*
  - comment - optional; type *string*
  - is\_generated - optional; type *boolean*
  - generating\_object\_info - optional; type *string*
  - id\_for\_export\_import - optional; type *string*

▪ metadata\_for\_export\_import - optional; type *string*

Output: set\_visual\_object\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_visual\_objectResponse*

### 231. set\_wind\_profile

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/set\_wind\_profile

Input: set\_wind\_profile\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_profile*

- value type *wind\_profile*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - type - optional; type *wind\_profile\_type* - type *undefined* with restriction - enum { 'ACCORDING\_TO\_STANDARD', 'USER\_DEFINED' }
  - user\_defined\_wind\_profile\_step\_enabled - optional; type *boolean*
  - user\_defined\_wind\_profile\_sorted - optional; type *boolean*
  - user\_defined\_wind\_profile\_uniform\_intensity\_enabled - optional; type *boolean*
  - user\_defined\_wind\_profile\_uniform\_intensity - optional; type *double*
  - user\_defined\_wind\_profile\_period\_step - optional; type *double*
  - user\_defined\_wind\_profile - optional; type *array\_of\_wind\_profile\_user\_defined\_wind\_profile*
    - wind\_profile\_user\_defined\_wind\_profile - optional, unbounded; type *wind\_profile\_user\_defined\_wind\_profile*
      - no - optional; type *int*
      - height - optional; type *double*
      - velocity - optional; type *double*
      - turbulence\_intensity - optional; type *double*
  - comment - optional; type *string*
  - load\_zone - optional; type *wind\_profile\_load\_zone* - type *undefined* with restriction - enum { 'E\_LOAD\_ZONE\_NOT\_DEFINED', 'E\_LOAD\_ZONE\_TYPE\_0\_18\_TO\_0\_38', 'E\_LOAD\_ZONE\_TYPE\_0\_39\_TO\_0\_48', 'E\_LOAD\_ZONE\_TYPE\_0\_4', 'E\_LOAD\_ZONE\_TYPE\_0\_49\_TO\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_0\_5', 'E\_LOAD\_ZONE\_TYPE\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_0\_61\_TO\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_0\_71\_TO\_0\_8', 'E\_LOAD\_ZONE\_TYPE\_0\_9', 'E\_LOAD\_ZONE\_TYPE\_0\_M', 'E\_LOAD\_ZONE\_TYPE\_1', 'E\_LOAD\_ZONE\_TYPE\_10', 'E\_LOAD\_ZONE\_TYPE\_100', 'E\_LOAD\_ZONE\_TYPE\_15', 'E\_LOAD\_ZONE\_TYPE\_19', 'E\_LOAD\_ZONE\_TYPE\_1\_0\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_1\_2', 'E\_LOAD\_ZONE\_TYPE\_1\_25', 'E\_LOAD\_ZONE\_TYPE\_1\_5', 'E\_LOAD\_ZONE\_TYPE\_1\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_1\_75', 'E\_LOAD\_ZONE\_TYPE\_1\_8', 'E\_LOAD\_ZONE\_TYPE\_1\_A', 'E\_LOAD\_ZONE\_TYPE\_1\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_1\_A\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_1\_B', 'E\_LOAD\_ZONE\_TYPE\_1\_TO\_2', 'E\_LOAD\_ZONE\_TYPE\_1\_V', 'E\_LOAD\_ZONE\_TYPE\_2', 'E\_LOAD\_ZONE\_TYPE\_20', 'E\_LOAD\_ZONE\_TYPE\_20\_9\_LESS', 'E\_LOAD\_ZONE\_TYPE\_21', 'E\_LOAD\_ZONE\_TYPE\_22', 'E\_LOAD\_ZONE\_TYPE\_23', 'E\_LOAD\_ZONE\_TYPE\_23\_6', 'E\_LOAD\_ZONE\_TYPE\_24', 'E\_LOAD\_ZONE\_TYPE\_24\_2\_LESS', 'E\_LOAD\_ZONE\_TYPE\_25', 'E\_LOAD\_ZONE\_TYPE\_26', 'E\_LOAD\_ZONE\_TYPE\_27', 'E\_LOAD\_ZONE\_TYPE\_27\_5\_LESS', 'E\_LOAD\_ZONE\_TYPE\_28', 'E\_LOAD\_ZONE\_TYPE\_29', 'E\_LOAD\_ZONE\_TYPE\_2\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_0\_2\_5', 'E\_LOAD\_ZONE\_TYPE\_2\_0\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_2\_1', 'E\_LOAD\_ZONE\_TYPE\_2\_3', 'E\_LOAD\_ZONE\_TYPE\_2\_4', 'E\_LOAD\_ZONE\_TYPE\_2\_5', 'E\_LOAD\_ZONE\_TYPE\_2\_5\_2\_75', 'E\_LOAD\_ZONE\_TYPE\_2\_5\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_2\_7', 'E\_LOAD\_ZONE\_TYPE\_2\_75', 'E\_LOAD\_ZONE\_TYPE\_2\_75\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_2\_75\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_2\_A', 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK\_2', 'E\_LOAD\_ZONE\_TYPE\_2\_ASTERISK\_3', 'E\_LOAD\_ZONE\_TYPE\_2\_B', 'E\_LOAD\_ZONE\_TYPE\_2\_V', 'E\_LOAD\_ZONE\_TYPE\_3', 'E\_LOAD\_ZONE\_TYPE\_30', 'E\_LOAD\_ZONE\_TYPE\_30\_8\_LESS', 'E\_LOAD\_ZONE\_TYPE\_32', 'E\_LOAD\_ZONE\_TYPE\_33', 'E\_LOAD\_ZONE\_TYPE\_34\_1\_LESS', 'E\_LOAD\_ZONE\_TYPE\_35', 'E\_LOAD\_ZONE\_TYPE\_36\_0', 'E\_LOAD\_ZONE\_TYPE\_37\_4\_LESS', 'E\_LOAD\_ZONE\_TYPE\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_3\_0\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_3\_0\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_3\_3', 'E\_LOAD\_ZONE\_TYPE\_3\_4', 'E\_LOAD\_ZONE\_TYPE\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_3\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_3\_6', 'E\_LOAD\_ZONE\_TYPE\_3\_A', 'E\_LOAD\_ZONE\_TYPE\_3\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_3\_A\_GREATER', 'E\_LOAD\_ZONE\_TYPE\_4', 'E\_LOAD\_ZONE\_TYPE\_40', 'E\_LOAD\_ZONE\_TYPE\_40\_7\_LESS', 'E\_LOAD\_ZONE\_TYPE\_44\_0\_LESS', 'E\_LOAD\_ZONE\_TYPE\_45', 'E\_LOAD\_ZONE\_TYPE\_47\_3\_LESS', 'E\_LOAD\_ZONE\_TYPE\_48', 'E\_LOAD\_ZONE\_TYPE\_4\_0', 'E\_LOAD\_ZONE\_TYPE\_4\_5', 'E\_LOAD\_ZONE\_TYPE\_4\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_5', 'E\_LOAD\_ZONE\_TYPE\_50', 'E\_LOAD\_ZONE\_TYPE\_50\_6\_LESS', 'E\_LOAD\_ZONE\_TYPE\_5\_0', 'E\_LOAD\_ZONE\_TYPE\_5\_5', 'E\_LOAD\_ZONE\_TYPE\_5\_5\_COMMA', 'E\_LOAD\_ZONE\_TYPE\_6', 'E\_LOAD\_ZONE\_TYPE\_60', 'E\_LOAD\_ZONE\_TYPE\_6\_0', 'E\_LOAD\_ZONE\_TYPE\_6\_5', 'E\_LOAD\_ZONE\_TYPE\_7', 'E\_LOAD\_ZONE\_TYPE\_70', 'E\_LOAD\_ZONE\_TYPE\_7\_0', 'E\_LOAD\_ZONE\_TYPE\_7\_5', 'E\_LOAD\_ZONE\_TYPE\_8', 'E\_LOAD\_ZONE\_TYPE\_80', 'E\_LOAD\_ZONE\_TYPE\_8\_0', 'E\_LOAD\_ZONE\_TYPE\_9', 'E\_LOAD\_ZONE\_TYPE\_90', 'E\_LOAD\_ZONE\_TYPE\_A', 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_200\_M', 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_400\_M', 'E\_LOAD\_ZONE\_TYPE\_ABOVE\_500\_M', 'E\_LOAD\_ZONE\_TYPE\_A\_1', 'E\_LOAD\_ZONE\_TYPE\_A\_2', 'E\_LOAD\_ZONE\_TYPE\_A\_2\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_A\_3', 'E\_LOAD\_ZONE\_TYPE\_A\_3\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_A\_4', 'E\_LOAD\_ZONE\_TYPE\_A\_4\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_B', 'E\_LOAD\_ZONE\_TYPE\_B\_1', 'E\_LOAD\_ZONE\_TYPE\_B\_2', 'E\_LOAD\_ZONE\_TYPE\_CS', 'E\_LOAD\_ZONE\_TYPE\_C\_1', 'E\_LOAD\_ZONE\_TYPE\_C\_2', 'E\_LOAD\_ZONE\_TYPE\_D', 'E\_LOAD\_ZONE\_TYPE\_E', 'E\_LOAD\_ZONE\_TYPE\_GAMMA', 'E\_LOAD\_ZONE\_TYPE\_GAMMA\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_GEQ\_8', 'E\_LOAD\_ZONE\_TYPE\_GUADELOUPE', 'E\_LOAD\_ZONE\_TYPE\_GUYANE', 'E\_LOAD\_ZONE\_TYPE\_H', 'E\_LOAD\_ZONE\_TYPE\_I', 'E\_LOAD\_ZONE\_TYPE\_II', 'E\_LOAD\_ZONE\_TYPE\_III', 'E\_LOAD\_ZONE\_TYPE\_III\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_II\_ASTERISK', 'E\_LOAD\_ZONE\_TYPE\_IV', 'E\_LOAD\_ZONE\_TYPE\_IX', 'E\_LOAD\_ZONE\_TYPE\_I\_A', 'E\_LOAD\_ZONE\_TYPE\_I\_ASTERISK\_A', 'E\_LOAD\_ZONE\_TYPE\_I\_ASTERISK\_M', 'E\_LOAD\_ZONE\_TYPE\_I\_M', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_25', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_30', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_35', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_40', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_45', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_50', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_55', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_6', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_60', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_75', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_8', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_9', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_0\_95', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_100', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_110', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_120', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_130', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_140', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_150', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_160', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_170', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_180', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_0', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_1', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_2', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_1\_75', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_200', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_0', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_25', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_2\_75', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_3\_0', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_3\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_4\_0', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_4\_5', 'E\_LOAD\_ZONE\_TYPE\_LEQ\_95', 'E\_LOAD\_ZONE\_TYPE\_MARTINIQUE', 'E\_LOAD\_ZONE\_TYPE\_MAYOTTE', 'E\_LOAD\_ZONE\_TYPE\_MEQ\_0\_7', 'E\_LOAD\_ZONE\_TYPE\_M\_1', 'E\_LOAD\_ZONE\_TYPE\_N\_A', 'E\_LOAD\_ZONE\_TYPE\_REUNION', 'E\_LOAD\_ZONE\_TYPE\_SR', 'E\_LOAD\_ZONE\_TYPE\_UNDER\_200\_M', 'E\_LOAD\_ZONE\_TYPE\_V', 'E\_LOAD\_ZONE\_TYPE\_VI',

'E\_LOAD\_ZONE\_TYPE\_VII', 'E\_LOAD\_ZONE\_TYPE\_VIII', 'E\_LOAD\_ZONE\_TYPE\_Z1', 'E\_LOAD\_ZONE\_TYPE\_Z2', 'E\_LOAD\_ZONE\_TYPE\_Z3', 'E\_LOAD\_ZONE\_TYPE\_ZERO' }

- definition\_type - optional; type *wind\_profile\_definition\_type* - type *undefined* with restriction - enum { 'E\_DEFINITION\_TYPE\_PARAMETERS\_FROM\_MAP', 'E\_DEFINITION\_TYPE\_USER\_DEFINED' }
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

**Output:** set\_wind\_profile\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_profileResponse*

### 232. set\_wind\_simulation

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_wind\_simulation

**Input:** set\_wind\_simulation\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_simulation*

- value type *wind\_simulation*
  - no type *int*
  - type - optional; type *wind\_simulation\_type* - type *undefined* with restriction - enum { 'STANDARD' }
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - active - optional; type *boolean*
  - wind\_profile - optional; type *int*
  - wind\_simulation\_analysis\_settings - optional; type *int*
  - wind\_direction\_type - optional; type *wind\_simulation\_wind\_direction\_type* - type *undefined* with restriction - enum { 'UNIFORM', 'USER\_DEFINED' }
  - uniform\_wind\_direction\_step - optional; type *double*
  - uniform\_wind\_direction\_range\_start - optional; type *double*
  - uniform\_wind\_direction\_range\_end - optional; type *double*
  - user\_defined\_list\_of\_wind\_directions - optional; type *array\_of\_int*
  - generate\_into\_load\_cases - optional; type *array\_of\_wind\_simulation\_generate\_into\_load\_cases*
    - wind\_simulation\_generate\_into\_load\_cases - optional, unbounded; type *wind\_simulation\_generate\_into\_load\_cases*
      - no - optional; type *int*
      - direction - optional; type *double*
      - load\_case - optional; type *int*
  - consider\_initial\_state - optional; type *boolean*
  - initial\_state\_case - optional; type *int*
  - initial\_state\_definition\_type - optional; type *wind\_simulation\_initial\_state\_definition\_type* - type *undefined* with restriction - enum { 'DEFINITION\_TYPE\_FINAL\_STATE', 'DEFINITION\_TYPE\_STIFFNESS', 'DEFINITION\_TYPE\_STRAINS', 'DEFINITION\_TYPE\_STRAINS\_WITH\_USER\_DEFINED\_FACTORS' }
  - individual\_factors\_of\_selected\_objects\_table - optional; type *array\_of\_wind\_simulation\_individual\_factors\_of\_selected\_objects\_table*
    - wind\_simulation\_individual\_factors\_of\_selected\_objects\_table - optional, unbounded; type *wind\_simulation\_individual\_factors\_of\_selected\_objects\_table*
      - no - optional; type *int*
      - object\_type - optional; type *object\_type* - type *undefined* with restriction - enum { 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_LINE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_MEMBER\_HINGE', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_NODE\_WITH\_SUPPORT', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SOLID', 'INITIAL\_STATE\_FACTORS\_OBJECT\_TYPE\_SURFACE' }
      - object\_list - optional; type *array\_of\_int*
      - strain\_type - optional; type *strain\_type* - type *undefined* with restriction - enum { 'ALL', 'ALONG\_X', 'ALONG\_Y', 'ALONG\_Z', 'AROUND\_X', 'AROUND\_Y', 'AROUND\_Z' }
      - factor - optional; type *double*
      - comment - optional; type *string*
  - comment - optional; type *string*
  - id\_for\_export\_import - optional; type *string*
  - metadata\_for\_export\_import - optional; type *string*

**Output:** set\_wind\_simulation\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_simulationResponse*

### 233. set\_wind\_simulation\_analysis\_settings

[Source code](#)

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** http://localhost:8082/set\_wind\_simulation\_analysis\_settings

**Input:** set\_wind\_simulation\_analysis\_settings\_request (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_simulation\_analysis\_settings*

- value type *wind\_simulation\_analysis\_settings*
  - no type *int*
  - user\_defined\_name\_enabled - optional; type *boolean*
  - name - optional; type *string*
  - comment - optional; type *string*
  - assigned\_to - optional; type *string*
  - density - optional; type *double*
  - kinematic\_viscosity - optional; type *double*
  - numerical\_solver - optional; type *wind\_simulation\_analysis\_settings\_numerical\_solver* - type *undefined* with restriction - enum { 'OPEN\_FOAM' }
  - finite\_volume\_mesh\_density - optional; type *double*
  - maximum\_number\_of\_iterations - optional; type *int*
  - target\_residue - optional; type *double*
  - mesh\_refinement\_type - optional; type *wind\_simulation\_analysis\_settings\_mesh\_refinement\_type* - type *undefined* with restriction - enum { 'DISTANCE\_FROM\_SURFACE', 'SURFACE\_CURVATURE' }
  - snap\_to\_model\_edges - optional; type *boolean*
  - boundary\_layers\_checked - optional; type *boolean*

- boundary\_layers\_value - optional; type *int*
- consider\_turbulence - optional; type *boolean*
- slip\_boundary\_condition\_on\_bottom\_boundary - optional; type *boolean*
- use\_potential\_flow\_for\_initial\_condition - optional; type *boolean*
- use\_second\_order\_numerical\_scheme - optional; type *boolean*
- consider\_wall\_roughness - optional; type *boolean*
- user\_defined\_dimensions\_of\_wind\_tunnel - optional; type *boolean*
- member\_load\_distribution - optional; type *wind\_simulation\_analysis\_settings\_member\_load\_distribution* - type *undefined* with restriction - enum { 'CONCENTRATED', 'TRAPEZOIDAL', 'UNIFORM' }
- turbulence\_model\_type - optional; type *wind\_simulation\_analysis\_settings\_turbulence\_model\_type* - type *undefined* with restriction - enum { 'TURBULENCE\_TYPE\_EPSILON', 'TURBULENCE\_TYPE\_OMEGA' }
- sand\_grain\_roughness\_height - optional; type *double*
- roughness\_constant - optional; type *double*
- id\_for\_export\_import - optional; type *string*
- metadata\_for\_export\_import - optional; type *string*

Output: set\_wind\_simulation\_analysis\_settings\_response (soap:body, use = literal) [Source code](#)

parameters type *set\_wind\_simulation\_analysis\_settingsResponse*

## 234. unite\_nodes\_and\_supports

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://localhost:8082/unite\_nodes\_and\_supports

Input: unite\_nodes\_and\_supports\_request (soap:body, use = literal) [Source code](#)

parameters type *unite\_nodes\_and\_supports*

- tolerance type *double*

Output: unite\_nodes\_and\_supports\_response (soap:body, use = literal) [Source code](#)

parameters type *unite\_nodes\_and\_supportsResponse*

## WSDL source code

```
<?xml version="1.0"?>
<definitions name="RfemModel" targetNamespace="http://www.dlupal.com"
 xmlns:tns="http://www.dlupal.com"
 xmlns:xsd="http://www.dlupal.com/rfem.xsd"
 xmlns="http://schemas.xmlsoap.org/wsdl/"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 >
 <types>
 <schema targetNamespace="http://www.dlupal.com/rfem.xsd"
 xmlns="http://www.w3.org/2001/XMLSchema"
 >
 <simpleType name="array_of_int">
 <list itemType="int"/>
 </simpleType>
 <complexType name="vector_3d">
 <sequence>
 <element name="x" type="double"/>
 <element name="y" type="double"/>
 <element name="z" type="double"/>
 </sequence>
 </complexType>
 <simpleType name="color">
 <restriction base="string"/>
 </simpleType>
 <complexType name="run_script">
 <sequence>
 <element name="script_file_path" type="string"/>
 </sequence>
 </complexType>
 <element name="run_script" type="xsd:run_script"/>
 <complexType name="run_scriptResponse">
 <sequence/>
 </complexType>
 <element name="run_scriptResponse" type="xsd:run_scriptResponse"/>
 <complexType name="model_main_parameters">
 <sequence>
 <element name="model_id" type="string"/>
 <element name="model_name" type="string"/>
 <element name="model_description" type="string"/>
 <element name="model_comment" type="string"/>
 <element name="model_path" type="string"/>
 <element name="project_id" type="string"/>
 <element name="project_name" type="string"/>
 <element name="project_description" type="string"/>
 <element name="project_folder" type="string"/>
 </sequence>
 </complexType>
 </schema>
 </types>

```

```

 </sequence>
 </complexType>
 <complexType name="get_model_main_parameters">
 <sequence/>
 </complexType>
 <element name="get_model_main_parameters" type="xsd:get_model_main_parameters"/>
 <complexType name="get_model_main_parametersResponse">
 <sequence>
 <element name="value" type="xsd:model_main_parameters"/>
 </sequence>
 </complexType>
 <element name="get_model_main_parametersResponse" type="xsd:get_model_main_parametersResponse"/>
 <complexType name="begin_modification">
 <sequence>
 <element name="modification_name" type="string"/>
 </sequence>
 </complexType>
 <element name="begin_modification" type="xsd:begin_modification"/>
 <complexType name="begin_modificationResponse">
 <sequence/>
 </complexType>
 <element name="begin_modificationResponse" type="xsd:begin_modificationResponse"/>
 <complexType name="finish_modification">
 <sequence/>
 </complexType>
 <element name="finish_modification" type="xsd:finish_modification"/>
 <complexType name="finish_modificationResponse">
 <sequence/>
 </complexType>
 <element name="finish_modificationResponse" type="xsd:finish_modificationResponse"/>
 <complexType name="cancel_modification">
 <sequence/>
 </complexType>
 <element name="cancel_modification" type="xsd:cancel_modification"/>
 <complexType name="cancel_modificationResponse">
 <sequence/>
 </complexType>
 <element name="cancel_modificationResponse" type="xsd:cancel_modificationResponse"/>
 <complexType name="reset">
 <sequence/>
 </complexType>
 <element name="reset" type="xsd:reset"/>
 <complexType name="resetResponse">
 <sequence/>
 </complexType>
 <element name="resetResponse" type="xsd:resetResponse"/>
 <complexType name="delete_all">
 <sequence/>
 </complexType>
 <element name="delete_all" type="xsd:delete_all"/>
 <complexType name="delete_allResponse">
 <sequence/>
 </complexType>
 <element name="delete_allResponse" type="xsd:delete_allResponse"/>
 <complexType name="unite_nodes_and_supports">
 <sequence>
 <element name="tolerance" type="double"/>
 </sequence>
 </complexType>
 <element name="unite_nodes_and_supports" type="xsd:unite_nodes_and_supports"/>
 <complexType name="unite_nodes_and_supportsResponse">
 <sequence/>
 </complexType>
 <element name="unite_nodes_and_supportsResponse" type="xsd:unite_nodes_and_supportsResponse"/>
 <complexType name="generate_load_cases_and_combinations">
 <sequence/>
 </complexType>
 <element name="generate_load_cases_and_combinations" type="xsd:generate_load_cases_and_combinations"/>
 <complexType name="generate_load_cases_and_combinationsResponse">
 <sequence/>
 </complexType>
 <element name="generate_load_cases_and_combinationsResponse" type="xsd:generate_load_cases_and_combinationsResponse"/>
 <simpleType name="model_type">
 <restriction base="string">
 <enumeration value="E_MODEL_TYPE_1D_X_3D"/>
 <enumeration value="E_MODEL_TYPE_1D_X_AXIAL"/>
 <enumeration value="E_MODEL_TYPE_2D_XY_3D"/>
 </restriction>
 </simpleType>

```



```

 <enumeration value="E_MODEL_TYPE_2D_XY_PLATE"/>
 <enumeration value="E_MODEL_TYPE_2D_XZ_3D"/>
 <enumeration value="E_MODEL_TYPE_2D_XZ_PLANE_STRAIN"/>
 <enumeration value="E_MODEL_TYPE_2D_XZ_PLANE_STRESS"/>
 <enumeration value="E_MODEL_TYPE_3D"/>
 </restriction>
</simpleType>
<complexType name="get_model_type">
 <sequence/>
</complexType>
<element name="get_model_type" type="xsd:get_model_type"/>
<complexType name="get_model_typeResponse">
 <sequence>
 <element name="value" type="xsd:model_type"/>
 </sequence>
</complexType>
<element name="get_model_typeResponse" type="xsd:get_model_typeResponse"/>
<complexType name="set_model_type">
 <sequence>
 <element name="model_type" type="xsd:model_type"/>
 </sequence>
</complexType>
<element name="set_model_type" type="xsd:set_model_type"/>
<complexType name="set_model_typeResponse">
 <sequence/>
</complexType>
<element name="set_model_typeResponse" type="xsd:set_model_typeResponse"/>
<complexType name="get_load_cases_and_combinations">
 <sequence/>
</complexType>
<element name="get_load_cases_and_combinations" type="xsd:get_load_cases_and_combinations"/>
<complexType name="get_load_cases_and_combinationsResponse">
 <sequence>
 <element name="value" type="xsd:load_cases_and_combinations"/>
 </sequence>
</complexType>
<element name="get_load_cases_and_combinationsResponse" type="xsd:get_load_cases_and_combinationsResponse"/>
<complexType name="set_load_cases_and_combinations">
 <sequence>
 <element name="value" type="xsd:load_cases_and_combinations"/>
 </sequence>
</complexType>
<element name="set_load_cases_and_combinations" type="xsd:set_load_cases_and_combinations"/>
<complexType name="set_load_cases_and_combinationsResponse">
 <sequence/>
</complexType>
<element name="set_load_cases_and_combinationsResponse" type="xsd:set_load_cases_and_combinationsResponse"/>
<complexType name="load_cases_and_combinations">
 <sequence>
 <element name="current_standard_for_combination_wizard" minOccurs="0" type="int"/>
 <element name="activate_combination_wizard_and_classification" minOccurs="0" type="boolean"/>
 <element name="activate_combination_wizard" minOccurs="0" type="boolean"/>
 <element name="result_combinations_active" minOccurs="0" type="boolean"/>
 <element name="result_combinations_parentheses_active" minOccurs="0" type="boolean"/>
 <element name="result_combinations_consider_sub_results" minOccurs="0" type="boolean"/>
 <element name="combination_name_according_to_action_category" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>
<complexType name="addon_list_design_addons_list_type">
 <sequence>
 <element name="stress_analysis_active" minOccurs="0" type="boolean"/>
 <element name="concrete_design_active" minOccurs="0" type="boolean"/>
 <element name="steel_design_active" minOccurs="0" type="boolean"/>
 <element name="timber_design_active" minOccurs="0" type="boolean"/>
 <element name="aluminum_design_active" minOccurs="0" type="boolean"/>
 <element name="steel_joints_active" minOccurs="0" type="boolean"/>
 <element name="timber_joints_active" minOccurs="0" type="boolean"/>
 <element name="craneway_design_active" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>
<element name="addon_list_design_addons_list_type" type="xsd:addon_list_design_addons_list_type"/>
<complexType name="addon_list_analysis_list_type">
 <sequence>
 <element name="material_nonlinear_analysis_active" minOccurs="0" type="boolean"/>
 <element name="structure_stability_active" minOccurs="0" type="boolean"/>
 <element name="construction_stages_active" minOccurs="0" type="boolean"/>
 </sequence>

```

```

 <element name="time_dependent_active" minOccurs="0" type="boolean"/>
 <element name="form_finding_active" minOccurs="0" type="boolean"/>
 <element name="cutting_patterns_active" minOccurs="0" type="boolean"/>
 <element name="torsional_warping_active" minOccurs="0" type="boolean"/>
 <element name="cost_estimation_active" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>
<element name="addon_list_analysis_list_type" type="xsd:addon_list_analysis_list_type"/>
<complexType name="addon_list_dynamic_analysis_settings_list_type">
 <sequence>
 <element name="modal_active" minOccurs="0" type="boolean"/>
 <element name="spectral_active" minOccurs="0" type="boolean"/>
 <element name="time_history_active" minOccurs="0" type="boolean"/>
 <element name="pushover_active" minOccurs="0" type="boolean"/>
 <element name="harmonic_response_active" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>
<element name="addon_list_dynamic_analysis_settings_list_type" type="xsd:addon_list_dynamic_analysis_settings_list_type"/>
<complexType name="addon_list_special_solutions_list_type">
 <sequence>
 <element name="building_model_active" minOccurs="0" type="boolean"/>
 <element name="wind_simulation_active" minOccurs="0" type="boolean"/>
 <element name="geotechnical_analysis_active" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>
<element name="addon_list_special_solutions_list_type" type="xsd:addon_list_special_solutions_list_type"/>
<complexType name="addon_list_type">
 <sequence>
 <element name="design_addons" minOccurs="0" type="xsd:addon_list_design_addons_list_type"/>
 <element name="masonry_design_active" minOccurs="0" type="boolean"/>
 <element name="multilayer_surfaces_design_active" minOccurs="0" type="boolean"/>
 <element name="analysis" minOccurs="0" type="xsd:addon_list_analysis_list_type"/>
 <element name="dynamic_analysis_settings" minOccurs="0" type="xsd:addon_list_dynamic_analysis_settings_list_type"/>
 <element name="special_solutions" minOccurs="0" type="xsd:addon_list_special_solutions_list_type"/>
 </sequence>
</complexType>
<element name="addon_list_type" type="xsd:addon_list_type"/>
<complexType name="get_addon_statuses">
 <sequence/>
</complexType>
<element name="get_addon_statuses" type="xsd:get_addon_statuses"/>
<complexType name="get_addon_statusesResponse">
 <sequence>
 <element name="value" type="xsd:addon_list_type"/>
 </sequence>
</complexType>
<element name="get_addon_statusesResponse" type="xsd:get_addon_statusesResponse"/>
<complexType name="set_addon_statuses">
 <sequence>
 <element name="addon_statuses" type="xsd:addon_list_type"/>
 </sequence>
</complexType>
<element name="set_addon_statuses" type="xsd:set_addon_statuses"/>
<complexType name="set_addon_statusesResponse">
 <sequence/>
</complexType>
<element name="set_addon_statusesResponse" type="xsd:set_addon_statusesResponse"/>
<simpleType name="material_material_type">
 <restriction base="string">
 <enumeration value="TYPE_ALUMINUM"/>
 <enumeration value="TYPE_BASIC"/>
 <enumeration value="TYPE_CONCRETE"/>
 <enumeration value="TYPE_FABRIC"/>
 <enumeration value="TYPE_FOIL"/>
 <enumeration value="TYPE_GAS"/>
 <enumeration value="TYPE_GLASS"/>
 <enumeration value="TYPE_MASONRY"/>
 <enumeration value="TYPE_METAL"/>
 <enumeration value="TYPE_REINFORCING_STEEL"/>
 <enumeration value="TYPE_SOIL"/>
 <enumeration value="TYPE_STEEL"/>
 <enumeration value="TYPE_TIMBER"/>
 </restriction>
</simpleType>
<simpleType name="material_material_model">
 <restriction base="string">

```

```

<enumeration value="MODEL_ISOTROPIC_DAMAGE_2D_3D"/>
<enumeration value="MODEL_ISOTROPIC_LINEAR_ELASTIC"/>
<enumeration value="MODEL_ISOTROPIC_MASONRY_PLASTIC_2D"/>
<enumeration value="MODEL_ISOTROPIC_NONLINEAR_ELASTIC_1D"/>
<enumeration value="MODEL_ISOTROPIC_NONLINEAR_ELASTIC_2D_3D"/>
<enumeration value="MODEL_ISOTROPIC_PLASTIC_1D"/>
<enumeration value="MODEL_ISOTROPIC_PLASTIC_2D_3D"/>
<enumeration value="MODEL_ISOTROPIC_SOIL_NONLINEAR_ELASTIC_3D"/>
<enumeration value="MODEL_ISOTROPIC_SOIL_PLASTIC_3D"/>
<enumeration value="MODEL_ISOTROPIC_TIMBER_LINEAR_ELASTIC_MEMBERS"/>
<enumeration value="MODEL_ORTHOTROPIC_2D"/>
<enumeration value="MODEL_ORTHOTROPIC_3D"/>
<enumeration value="MODEL_ORTHOTROPIC_MASONRY_PLASTIC_2D"/>
<enumeration value="MODEL_ORTHOTROPIC_PLASTIC_2D"/>
<enumeration value="MODEL_ORTHOTROPIC_PLASTIC_3D"/>
<enumeration value="MODEL_ORTHOTROPIC_TIMBER_LINEAR_ELASTIC_SURFACES"/>
</restriction>
</simpleType>
<simpleType name="material_application_context">
 <restriction base="string">
 <enumeration value="ALUMINUM_DESIGN"/>
 <enumeration value="COMBINATION_WIZARD"/>
 <enumeration value="CONCRETE_DESIGN"/>
 <enumeration value="CONCRETE_FOUNDATION_DESIGN"/>
 <enumeration value="CRANEWAY_DESIGN"/>
 <enumeration value="DYNAMIC_ANALYSIS"/>
 <enumeration value="GEOTECHNICAL_ANALYSIS"/>
 <enumeration value="GLASS_DESIGN"/>
 <enumeration value="INVALID"/>
 <enumeration value="INVALID_DESIGN"/>
 <enumeration value="LOAD_WIZARD"/>
 <enumeration value="MASONRY_DESIGN"/>
 <enumeration value="PIPING_DESIGN"/>
 <enumeration value="STEEL_DESIGN"/>
 <enumeration value="STEEL_JOINT_DESIGN"/>
 <enumeration value="TIMBER_DESIGN"/>
 <enumeration value="TIMBER_JOINT_DESIGN"/>
 <enumeration value="TOWER_DESIGN"/>
 </restriction>
</simpleType>
<simpleType name="material_diagram_type">
 <restriction base="string">
 <enumeration value="DIAGRAM_TYPE_BASIC"/>
 <enumeration value="DIAGRAM_TYPE_BILINEAR"/>
 <enumeration value="DIAGRAM_TYPE_STRESS_STRAIN_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="material_definition_type">
 <restriction base="string">
 <enumeration value="DERIVED_G"/>
 <enumeration value="DERIVED_NU"/>
 <enumeration value="E_G_NO_NU"/>
 <enumeration value="E_G_NU"/>
 <enumeration value="NONE"/>
 </restriction>
</simpleType>
<simpleType name="material_stress_failure_hypothesis">
 <restriction base="string">
 <enumeration value="E_STRESS_FAILURE_HYPOTHESIS_OEDOMETRIC_CONDITIONS_WITH_SMALL_STRAIN_STIFFNESS"/>
 <enumeration value="STRESS_FAILURE_HYPOTHESIS_DRUCKER_PRAGER"/>
 <enumeration value="STRESS_FAILURE_HYPOTHESIS_MOHR_COULOMB"/>
 <enumeration value="STRESS_FAILURE_HYPOTHESIS_OEDOMETRIC_CONDITIONS"/>
 <enumeration value="STRESS_FAILURE_HYPOTHESIS_TRESCA"/>
 <enumeration value="STRESS_FAILURE_HYPOTHESIS_VON_MISES"/>
 </restriction>
</simpleType>
<simpleType name="material_stiffness_modification_type">
 <restriction base="string">
 <enumeration value="STIFFNESS_MODIFICATION_TYPE_DIVISION"/>
 <enumeration value="STIFFNESS_MODIFICATION_TYPE_MULTIPLICATION"/>
 </restriction>
</simpleType>
<simpleType name="material_members_weight_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CWT"/>
 </restriction>

```

```

<enumeration value="COST_ESTIMATION_G"/>
<enumeration value="COST_ESTIMATION_KG"/>
<enumeration value="COST_ESTIMATION_LB"/>
<enumeration value="COST_ESTIMATION_OZ"/>
<enumeration value="COST_ESTIMATION_SLUG"/>
<enumeration value="COST_ESTIMATION_T"/>
<enumeration value="COST_ESTIMATION_TON"/>
</restriction>
</simpleType>
<simpleType name="material_members_volume_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM3"/>
 <enumeration value="COST_ESTIMATION_FLOZ"/>
 <enumeration value="COST_ESTIMATION_FT3"/>
 <enumeration value="COST_ESTIMATION_IMPGAL"/>
 <enumeration value="COST_ESTIMATION_IN3"/>
 <enumeration value="COST_ESTIMATION_L"/>
 <enumeration value="COST_ESTIMATION_M3"/>
 <enumeration value="COST_ESTIMATION_MM3"/>
 <enumeration value="COST_ESTIMATION_PT"/>
 <enumeration value="COST_ESTIMATION_QT"/>
 <enumeration value="COST_ESTIMATION_USGAL"/>
 <enumeration value="COST_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="material_members_surface_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM2"/>
 <enumeration value="COST_ESTIMATION_FT2"/>
 <enumeration value="COST_ESTIMATION_IN2"/>
 <enumeration value="COST_ESTIMATION_M2"/>
 <enumeration value="COST_ESTIMATION_MM2"/>
 <enumeration value="COST_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="material_surfaces_weight_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CWT"/>
 <enumeration value="COST_ESTIMATION_G"/>
 <enumeration value="COST_ESTIMATION_KG"/>
 <enumeration value="COST_ESTIMATION_LB"/>
 <enumeration value="COST_ESTIMATION_OZ"/>
 <enumeration value="COST_ESTIMATION_SLUG"/>
 <enumeration value="COST_ESTIMATION_T"/>
 <enumeration value="COST_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="material_surfaces_volume_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM3"/>
 <enumeration value="COST_ESTIMATION_FLOZ"/>
 <enumeration value="COST_ESTIMATION_FT3"/>
 <enumeration value="COST_ESTIMATION_IMPGAL"/>
 <enumeration value="COST_ESTIMATION_IN3"/>
 <enumeration value="COST_ESTIMATION_L"/>
 <enumeration value="COST_ESTIMATION_M3"/>
 <enumeration value="COST_ESTIMATION_MM3"/>
 <enumeration value="COST_ESTIMATION_PT"/>
 <enumeration value="COST_ESTIMATION_QT"/>
 <enumeration value="COST_ESTIMATION_USGAL"/>
 <enumeration value="COST_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="material_surfaces_top_face_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM2"/>
 <enumeration value="COST_ESTIMATION_FT2"/>
 <enumeration value="COST_ESTIMATION_IN2"/>
 <enumeration value="COST_ESTIMATION_M2"/>
 <enumeration value="COST_ESTIMATION_MM2"/>
 <enumeration value="COST_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="material_surfaces_area_unit">
 <restriction base="string">

```

```
<enumeration value="COST_ESTIMATION_CM2"/>
<enumeration value="COST_ESTIMATION_FT2"/>
<enumeration value="COST_ESTIMATION_IN2"/>
<enumeration value="COST_ESTIMATION_M2"/>
<enumeration value="COST_ESTIMATION_MM2"/>
<enumeration value="COST_ESTIMATION_YD2"/>
</restriction>
</simpleType>
<simpleType name="material_surfaces_bottom_face_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM2"/>
 <enumeration value="COST_ESTIMATION_FT2"/>
 <enumeration value="COST_ESTIMATION_IN2"/>
 <enumeration value="COST_ESTIMATION_M2"/>
 <enumeration value="COST_ESTIMATION_MM2"/>
 <enumeration value="COST_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="material_solids_weight_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CWT"/>
 <enumeration value="COST_ESTIMATION_G"/>
 <enumeration value="COST_ESTIMATION_KG"/>
 <enumeration value="COST_ESTIMATION_LB"/>
 <enumeration value="COST_ESTIMATION_OZ"/>
 <enumeration value="COST_ESTIMATION_SLUG"/>
 <enumeration value="COST_ESTIMATION_T"/>
 <enumeration value="COST_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="material_solids_volume_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM3"/>
 <enumeration value="COST_ESTIMATION_FLOZ"/>
 <enumeration value="COST_ESTIMATION_FT3"/>
 <enumeration value="COST_ESTIMATION_IMPGAL"/>
 <enumeration value="COST_ESTIMATION_IN3"/>
 <enumeration value="COST_ESTIMATION_L"/>
 <enumeration value="COST_ESTIMATION_M3"/>
 <enumeration value="COST_ESTIMATION_MM3"/>
 <enumeration value="COST_ESTIMATION_PT"/>
 <enumeration value="COST_ESTIMATION_QT"/>
 <enumeration value="COST_ESTIMATION_USGAL"/>
 <enumeration value="COST_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="material_solids_area_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM2"/>
 <enumeration value="COST_ESTIMATION_FT2"/>
 <enumeration value="COST_ESTIMATION_IN2"/>
 <enumeration value="COST_ESTIMATION_M2"/>
 <enumeration value="COST_ESTIMATION_MM2"/>
 <enumeration value="COST_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="material_emissions_members_weight_unit">
 <restriction base="string">
 <enumeration value="EMISSION_ESTIMATION_CWT"/>
 <enumeration value="EMISSION_ESTIMATION_G"/>
 <enumeration value="EMISSION_ESTIMATION_KG"/>
 <enumeration value="EMISSION_ESTIMATION_LB"/>
 <enumeration value="EMISSION_ESTIMATION_OZ"/>
 <enumeration value="EMISSION_ESTIMATION_SLUG"/>
 <enumeration value="EMISSION_ESTIMATION_T"/>
 <enumeration value="EMISSION_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="material_emissions_members_volume_unit">
 <restriction base="string">
 <enumeration value="EMISSION_ESTIMATION_CM3"/>
 <enumeration value="EMISSION_ESTIMATION_FLOZ"/>
 <enumeration value="EMISSION_ESTIMATION_FT3"/>
 <enumeration value="EMISSION_ESTIMATION_IMPGAL"/>
 <enumeration value="EMISSION_ESTIMATION_IN3"/>
```



```

 </restriction>
</simpleType>
<simpleType name="material_emissions_solid_weight_unit">
 <restriction base="string">
 <enumeration value="EMISSION_ESTIMATION_CWT"/>
 <enumeration value="EMISSION_ESTIMATION_G"/>
 <enumeration value="EMISSION_ESTIMATION_KG"/>
 <enumeration value="EMISSION_ESTIMATION_LB"/>
 <enumeration value="EMISSION_ESTIMATION_OZ"/>
 <enumeration value="EMISSION_ESTIMATION_SLUG"/>
 <enumeration value="EMISSION_ESTIMATION_T"/>
 <enumeration value="EMISSION_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="material_emissions_solid_volume_unit">
 <restriction base="string">
 <enumeration value="EMISSION_ESTIMATION_CM3"/>
 <enumeration value="EMISSION_ESTIMATION_FLOZ"/>
 <enumeration value="EMISSION_ESTIMATION_FT3"/>
 <enumeration value="EMISSION_ESTIMATION_IMP GAL"/>
 <enumeration value="EMISSION_ESTIMATION_IN3"/>
 <enumeration value="EMISSION_ESTIMATION_L"/>
 <enumeration value="EMISSION_ESTIMATION_M3"/>
 <enumeration value="EMISSION_ESTIMATION_MM3"/>
 <enumeration value="EMISSION_ESTIMATION_PT"/>
 <enumeration value="EMISSION_ESTIMATION_QT"/>
 <enumeration value="EMISSION_ESTIMATION_US GAL"/>
 <enumeration value="EMISSION_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="material_emissions_solid_area_unit">
 <restriction base="string">
 <enumeration value="EMISSION_ESTIMATION_CM2"/>
 <enumeration value="EMISSION_ESTIMATION_FT2"/>
 <enumeration value="EMISSION_ESTIMATION_IN2"/>
 <enumeration value="EMISSION_ESTIMATION_M2"/>
 <enumeration value="EMISSION_ESTIMATION_MM2"/>
 <enumeration value="EMISSION_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<complexType name="get_material">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_material" type="xsd:get_material"/>
<complexType name="get_materialResponse">
 <sequence>
 <element name="value" type="xsd:material"/>
 </sequence>
</complexType>
<element name="get_materialResponse" type="xsd:get_materialResponse"/>
<complexType name="set_material">
 <sequence>
 <element name="value" type="xsd:material"/>
 </sequence>
</complexType>
<element name="set_material" type="xsd:set_material"/>
<complexType name="set_materialResponse">
 <sequence/>
</complexType>
<element name="set_materialResponse" type="xsd:set_materialResponse"/>
<complexType name="material">
 <sequence>
 <element name="no" type="int"/>
 <element name="material_type" minOccurs="0" type="xsd:material_material_type"/>
 <element name="material_model" minOccurs="0" type="xsd:material_material_model"/>
 <element name="application_context" minOccurs="0" type="xsd:material_application_context"/>
 <element name="diagram_type" minOccurs="0" type="xsd:material_diagram_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="user_defined" minOccurs="0" type="boolean"/>
 <element name="definition_type" minOccurs="0" type="xsd:material_definition_type"/>
 <element name="stress_failure_hypothesis" minOccurs="0" type="xsd:material_stress_failure_hypothesis"/>
 <element name="is_temperature_dependent" minOccurs="0" type="boolean"/>
 <element name="is_dynamic_increase_factor" minOccurs="0" type="boolean"/>
 </sequence>

```

```
<element name="has_cost_estimation" minOccurs="0" type="boolean"/>
<element name="optimization" minOccurs="0" type="boolean"/>
<element name="has_emissions_estimation" minOccurs="0" type="boolean"/>
<element name="reference_temperature" minOccurs="0" type="double"/>
<element name="temperature_properties_sorted" minOccurs="0" type="boolean"/>
<element name="stiffness_matrix_editable" minOccurs="0" type="boolean"/>
<element name="stiffness_modification" minOccurs="0" type="boolean"/>
<element name="stiffness_modification_type" minOccurs="0" type="xsd:material_stiffness_modification_type"/>
<element name="has_linear_elastic_with_nonlinear_criteria" minOccurs="0" type="boolean"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="members_weight_active" minOccurs="0" type="boolean"/>
<element name="members_weight_unit_cost" minOccurs="0" type="double"/>
<element name="members_weight_unit" minOccurs="0" type="xsd:material_members_weight_unit"/>
<element name="members_weight_quantity" minOccurs="0" type="double"/>
<element name="members_weight_cost" minOccurs="0" type="double"/>
<element name="members_volume_active" minOccurs="0" type="boolean"/>
<element name="members_volume_unit_cost" minOccurs="0" type="double"/>
<element name="members_volume_unit" minOccurs="0" type="xsd:material_members_volume_unit"/>
<element name="members_volume_quantity" minOccurs="0" type="double"/>
<element name="members_volume_cost" minOccurs="0" type="double"/>
<element name="members_surface_active" minOccurs="0" type="boolean"/>
<element name="members_surface_unit_cost" minOccurs="0" type="double"/>
<element name="members_surface_unit" minOccurs="0" type="xsd:material_members_surface_unit"/>
<element name="members_surface_quantity" minOccurs="0" type="double"/>
<element name="members_surface_cost" minOccurs="0" type="double"/>
<element name="surfaces_weight_active" minOccurs="0" type="boolean"/>
<element name="surfaces_weight_unit_cost" minOccurs="0" type="double"/>
<element name="surfaces_weight_unit" minOccurs="0" type="xsd:material_surfaces_weight_unit"/>
<element name="surfaces_weight_quantity" minOccurs="0" type="double"/>
<element name="surfaces_weight_cost" minOccurs="0" type="double"/>
<element name="surfaces_volume_active" minOccurs="0" type="boolean"/>
<element name="surfaces_volume_unit_cost" minOccurs="0" type="double"/>
<element name="surfaces_volume_unit" minOccurs="0" type="xsd:material_surfaces_volume_unit"/>
<element name="surfaces_volume_quantity" minOccurs="0" type="double"/>
<element name="surfaces_volume_cost" minOccurs="0" type="double"/>
<element name="surfaces_top_face_active" minOccurs="0" type="boolean"/>
<element name="surfaces_top_face_unit_cost" minOccurs="0" type="double"/>
<element name="surfaces_top_face_unit" minOccurs="0" type="xsd:material_surfaces_top_face_unit"/>
<element name="surfaces_top_face_quantity" minOccurs="0" type="double"/>
<element name="surfaces_top_face_cost" minOccurs="0" type="double"/>
<element name="surfaces_area_active" minOccurs="0" type="boolean"/>
<element name="surfaces_area_unit_cost" minOccurs="0" type="double"/>
<element name="surfaces_area_unit" minOccurs="0" type="xsd:material_surfaces_area_unit"/>
<element name="surfaces_area_quantity" minOccurs="0" type="double"/>
<element name="surfaces_area_cost" minOccurs="0" type="double"/>
<element name="surfaces_bottom_face_active" minOccurs="0" type="boolean"/>
<element name="surfaces_bottom_face_unit_cost" minOccurs="0" type="double"/>
<element name="surfaces_bottom_face_unit" minOccurs="0" type="xsd:material_surfaces_bottom_face_unit"/>
<element name="surfaces_bottom_face_quantity" minOccurs="0" type="double"/>
<element name="surfaces_bottom_face_cost" minOccurs="0" type="double"/>
<element name="solids_weight_active" minOccurs="0" type="boolean"/>
<element name="solids_weight_unit_cost" minOccurs="0" type="double"/>
<element name="solids_weight_unit" minOccurs="0" type="xsd:material_solids_weight_unit"/>
<element name="solids_weight_quantity" minOccurs="0" type="double"/>
<element name="solids_weight_cost" minOccurs="0" type="double"/>
<element name="solids_volume_active" minOccurs="0" type="boolean"/>
<element name="solids_volume_unit_cost" minOccurs="0" type="double"/>
<element name="solids_volume_unit" minOccurs="0" type="xsd:material_solids_volume_unit"/>
<element name="solids_volume_quantity" minOccurs="0" type="double"/>
<element name="solids_volume_cost" minOccurs="0" type="double"/>
<element name="solids_area_active" minOccurs="0" type="boolean"/>
<element name="solids_area_unit_cost" minOccurs="0" type="double"/>
<element name="solids_area_unit" minOccurs="0" type="xsd:material_solids_area_unit"/>
<element name="solids_area_quantity" minOccurs="0" type="double"/>
<element name="solids_area_cost" minOccurs="0" type="double"/>
<element name="sum_weight" minOccurs="0" type="double"/>
<element name="weight_percentage" minOccurs="0" type="double"/>
<element name="cost_percentage" minOccurs="0" type="double"/>
<element name="sum_cost" minOccurs="0" type="double"/>
<element name="total_cost" minOccurs="0" type="double"/>
<element name="emissions_members_weight_active" minOccurs="0" type="boolean"/>
```



```

<element name="emissions_members_weight_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_weight_unit" minOccurs="0" type="xsd:material_emissions_members_weight_unit"/>
<element name="emissions_members_weight_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_weight_cost" minOccurs="0" type="double"/>
<element name="emissions_members_volume_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_volume_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_volume_unit" minOccurs="0" type="xsd:material_emissions_members_volume_unit"/>
<element name="emissions_members_volume_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_volume_cost" minOccurs="0" type="double"/>
<element name="emissions_members_surface_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_surface_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_surface_unit" minOccurs="0" type="xsd:material_emissions_members_surface_unit"/>
<element name="emissions_members_surface_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_surface_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_weight_active" minOccurs="0" type="boolean"/>
<element name="emissions_surfaces_weight_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_weight_unit" minOccurs="0" type="xsd:material_emissions_surfaces_weight_unit"/>
<element name="emissions_surfaces_weight_quantity" minOccurs="0" type="double"/>
<element name="emissions_surfaces_weight_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_volume_active" minOccurs="0" type="boolean"/>
<element name="emissions_surfaces_volume_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_volume_unit" minOccurs="0" type="xsd:material_emissions_surfaces_volume_unit"/>
<element name="emissions_surfaces_volume_quantity" minOccurs="0" type="double"/>
<element name="emissions_surfaces_volume_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_top_face_active" minOccurs="0" type="boolean"/>
<element name="emissions_surfaces_top_face_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_top_face_unit" minOccurs="0" type="xsd:material_emissions_surfaces_top_face_unit"/>
<element name="emissions_surfaces_top_face_quantity" minOccurs="0" type="double"/>
<element name="emissions_surfaces_top_face_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_area_active" minOccurs="0" type="boolean"/>
<element name="emissions_surfaces_area_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_area_unit" minOccurs="0" type="xsd:material_emissions_surfaces_area_unit"/>
<element name="emissions_surfaces_area_quantity" minOccurs="0" type="double"/>
<element name="emissions_surfaces_area_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_bottom_face_active" minOccurs="0" type="boolean"/>
<element name="emissions_surfaces_bottom_face_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_surfaces_bottom_face_unit" minOccurs="0" type="xsd:material_emissions_surfaces_bottom_face_unit"/>
<element name="emissions_surfaces_bottom_face_quantity" minOccurs="0" type="double"/>
<element name="emissions_surfaces_bottom_face_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_weight_active" minOccurs="0" type="boolean"/>
<element name="emissions_solids_weight_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_weight_unit" minOccurs="0" type="xsd:material_emissions_solids_weight_unit"/>
<element name="emissions_solids_weight_quantity" minOccurs="0" type="double"/>
<element name="emissions_solids_weight_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_volume_active" minOccurs="0" type="boolean"/>
<element name="emissions_solids_volume_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_volume_unit" minOccurs="0" type="xsd:material_emissions_solids_volume_unit"/>
<element name="emissions_solids_volume_quantity" minOccurs="0" type="double"/>
<element name="emissions_solids_volume_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_area_active" minOccurs="0" type="boolean"/>
<element name="emissions_solids_area_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_solids_area_unit" minOccurs="0" type="xsd:material_emissions_solids_area_unit"/>
<element name="emissions_solids_area_quantity" minOccurs="0" type="double"/>
<element name="emissions_solids_area_cost" minOccurs="0" type="double"/>
<element name="emissions_percentage" minOccurs="0" type="double"/>
<element name="emissions_sum" minOccurs="0" type="double"/>
<element name="emissions_total" minOccurs="0" type="double"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="section_type">
 <restriction base="string">
 <enumeration value="TYPE_BASIC"/>
 <enumeration value="TYPE_BUILT_UP_STEEL"/>
 <enumeration value="TYPE_BUILT_UP_TIMBER"/>
 <enumeration value="TYPE_GENERAL_BY_RSECTION"/>
 <enumeration value="TYPE_PARAMETRIC_BARS"/>
 <enumeration value="TYPE_PARAMETRIC_BRIDGES"/>
 <enumeration value="TYPE_PARAMETRIC_MASSIVE_I"/>
 <enumeration value="TYPE_PARAMETRIC_MASSIVE_II"/>
 <enumeration value="TYPE_PARAMETRIC_THIN_WALLED"/>
 <enumeration value="TYPE_PHASE"/>
 </restriction>
</simpleType>

```

```
<enumeration value="TYPE_STANDARDIZED_STEEL"/>
<enumeration value="TYPE_STANDARDIZED_TIMBER"/>
</restriction>
</simpleType>
<simpleType name="section_manufacturing_type">
 <restriction base="string">
 <enumeration value="MANUFACTURING_TYPE_COLD_FORMED"/>
 <enumeration value="MANUFACTURING_TYPE_GLULAM"/>
 <enumeration value="MANUFACTURING_TYPE_HOT_ROLLED"/>
 <enumeration value="MANUFACTURING_TYPE_NONE"/>
 <enumeration value="MANUFACTURING_TYPE_SAWN"/>
 <enumeration value="MANUFACTURING_TYPE_WELDED"/>
 </restriction>
</simpleType>
<simpleType name="section_parametrization_type">
 <restriction base="string">
 <enumeration value="BUILT_UP_STEEL_CHANNELS_U_BU"/>
 <enumeration value="BUILT_UP_STEEL_CUT_I_SECTIONS_WITH_FLAT_BAR_CIF_BU"/>
 <enumeration value="BUILT_UP_STEEL_CUT_I_SECTIONS_WITH_MODIFIED_DEPTH_CIMD_BU"/>
 <enumeration value="BUILT_UP_STEEL_EQUAL_LEG_ANGLES_LE_BU"/>
 <enumeration value="BUILT_UP_STEEL_I_SECTIONS_OR_CHANNELS_WITH_SIDE_FLAT_BARS_SF_BU"/>
 <enumeration value="BUILT_UP_STEEL_I_SECTIONS_WITH_CHANNELS_IU_BU"/>
 <enumeration value="BUILT_UP_STEEL_I_SECTIONS_WITH_OTHER_SECTION_ON_FLANGE_I_OF_BU"/>
 <enumeration value="BUILT_UP_STEEL_I_SECTIONS_I_BU"/>
 <enumeration value="BUILT_UP_STEEL_UNEQUAL_LEG_ANGLES_LU_BU"/>
 <enumeration value="BUILT_UP_TIMBER_RECTANGLES_R_BU"/>
 <enumeration value="PARAMETRIC_BARS_FLAT_BAR_FLAT"/>
 <enumeration value="PARAMETRIC_BARS_HALF_OVAL_BAR_HALFOVAL"/>
 <enumeration value="PARAMETRIC_BARS_HALF_ROUND_BAR_HALFRound"/>
 <enumeration value="PARAMETRIC_BARS_HEXAGON_BAR_HEXAGON"/>
 <enumeration value="PARAMETRIC_BARS_ISOSCELES_RIGHT_TRIANGLE_BAR_TRIANGLEIR"/>
 <enumeration value="PARAMETRIC_BARS_OCTAGON_BAR_OCTAGON"/>
 <enumeration value="PARAMETRIC_BARS_ROUND_BAR_ROUND"/>
 <enumeration value="PARAMETRIC_BARS_ROUND_CORNER_SQUARE_BAR_SQUARER"/>
 <enumeration value="PARAMETRIC_BARS_SHARP_CORNER_SQUARE_BAR_SQUARES"/>
 <enumeration value="PARAMETRIC_BRIDGES_MULTI_PIECE_TAPERED_FLANGE_T_SECTION_TTF_B"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_2X2_MASSIVE_RECTANGLES_WITH_FULLY_RIGID_CONNECTION_2X2R_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_2X3_MASSIVE_RECTANGLES_WITH_FULLY_RIGID_CONNECTION_2X3R_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_2_MASSIVE_RECTANGLES_WITH_A_CONTINUOUS_PACK_AND_2_SPLICES_2RP2S_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_2_MASSIVE_RECTANGLES_2R_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_3_MASSIVE_RECTANGLES_3R_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_4_MASSIVE_RECTANGLES_4R_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_BOX_WITH_HORIZONTAL_CONNECTION_LINES_BOXH_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_BOX_WITH_OVERHANGING_FLANGES_AND_HORIZONTAL_CONNECTION_LINES_BOXOFV"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_BOX_WITH_OVERHANGING_FLANGES_AND_VERTICAL_CONNECTION_LINES_BOXOFV"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_BOX_WITH_VERTICAL_CONNECTION_LINES_BOXV_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_FIVE_LAYER_RECTANGLE_WITH_3_DIFFERENT_MATERIALS_5LR_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_OVAL_WITH_VERTICAL_CONNECTION_LINES_OVAL_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_RECTANGLE_WITH_2_CONTINUOUS_SPLICES_R2S_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_SINGLY_SYMMETRIC_I_SECTION_WITH_HORIZONTAL_CONNECTION_LINES_ISH_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_SINGLY_SYMMETRIC_I_SECTION_WITH_THIN_FLANGES_AND_4_REINFORCING_RECTA"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_SINGLY_SYMMETRIC_I_SECTION_WITH_VERTICAL_CONNECTION_LINES_ISV_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_THREE_LAYER_RECTANGLE_WITH_2_DIFFERENT_MATERIALS_3LR_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_TWO_BOX_WITH_HORIZONTAL_CONNECTION_LINES_2BOXH_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_TWO_BOX_WITH_VERTICAL_CONNECTION_LINES_2BOXV_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_TWO_LAYER_RECTANGLE_WITH_2_DIFFERENT_MATERIALS_2LR_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_T_SECTION_WITH_A_HORIZONTAL_CONNECTION_LINE_TH_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_T_SECTION_WITH_VERTICAL_CONNECTION_LINES_TV_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_U_SECTION_WITH_HORIZONTAL_CONNECTION_LINES_UH_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_II_MASSIVE_U_SECTION_WITH_VERTICAL_CONNECTION_LINES_UV_M2"/>
 <enumeration value="PARAMETRIC_MASSIVE_I_MASSIVE_CIRCLE_CIRCLE_M1"/>
 <enumeration value="PARAMETRIC_MASSIVE_I_MASSIVE_DOUBLE_T_SECTION_WITH_CHAMFERED_INNER_CORNERS_DTC_M1"/>
 <enumeration value="PARAMETRIC_MASSIVE_I_MASSIVE_DOUBLE_T_SECTION_DT_M1"/>
 <enumeration value="PARAMETRIC_MASSIVE_I_MASSIVE_DOUBLY_SYMMETRIC_I_SECTION_ID_M1"/>
 </restriction>
</simpleType>
```

<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_OVAL\_HALF\_OVAL\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HALF\_ROUND\_HALF\_ROUND\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HEXAGON\_HEXAGON\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_HOLLOW\_CIRCLE\_HCIRCLE\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_I\_SECTION\_I\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_L\_SECTION\_L\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OCTAGON\_OCTAGON\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_OVAL\_OVAL\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_CHAMFERED\_RECTANGULAR\_OPENING\_RROC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_WITH\_RECTANGULAR\_OPENING\_RRO\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_RECTANGLE\_R\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_RECTANGLE\_RR\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_ROUND\_CORNER\_SQUARE\_SQR\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_SQUARE\_SQ\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_DOUBLE\_T\_SECTION\_DTT\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_I\_SECTION\_ITF\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_FLANGE\_T\_SECTION\_TTF\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_U\_SECTION\_UT\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TTWC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TAPERED\_WEB\_T\_SECTION\_TTW\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_TRAPEZOID\_TR\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_TC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_T\_SECTION\_T\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_IUC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_I\_SECTION\_IU\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_UNSYMMETRIC\_T\_SECTION\_TU\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_WITH\_CHAMFERED\_INNER\_CORNERS\_UC\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_U\_SECTION\_U\_M1"/>  
<enumeration value="PARAMETRIC\_MASSIVE\_I\_MASSIVE\_Z\_SECTION\_Z\_M1"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_AX\_BOX\_AX"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_A\_BOX\_A"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_BX\_BOX\_BX"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_BOX\_TYPE\_B\_BOX\_B"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_CHANNEL\_U"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_CIRCULAR\_HOLLOW\_SECTION\_CHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_CROSS\_SHAPED\_SECTION\_CROSS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_ELLIPTICAL\_HOLLOW\_SECTION\_EHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_EQUAL\_LEG\_ANGLE\_LE"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_2\_SIDE\_FLAT\_BARS\_I2SFB"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_WITH\_T\_SECTION\_IT"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_I\_SECTION\_I"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_A\_PI\_A"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_PI\_SECTION\_TYPE\_B\_PI\_B"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_POLYGONAL\_HOLLOW\_SECTION\_PHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAKS\_OUTWARD\_AND\_INWARD\_RHSPOI"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_WITH\_PEAK\_OUTWARD\_RHSPO"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_RECTANGULAR\_HOLLOW\_SECTION\_RHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_RIB\_HOLLOW\_SECTION\_RIBHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_FLAT\_BARS\_IS2FB"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_SLOPING\_STIFFENERS\_IS2SS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_2\_WELDED\_FLAT\_BARS\_IS2FBW"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_DOUBLE\_WEB\_THICKNESS\_IS2WT"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_WITH\_EDGE\_STIFFENERS\_ON\_1\_FLANGE\_IS1FES"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SINGLY\_SYMMETRIC\_I\_SECTION\_IS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_SQUARE\_HOLLOW\_SECTION\_SHS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_CHANNEL\_UTF"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_I\_SECTION\_ITF"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_WITH\_TAPERED\_WEB\_TTFW"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_TAPERED\_FLANGE\_T\_SECTION\_TTF"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_TRAPEZOIDAL\_HOLLOW\_SECTION\_THS"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_T\_SECTION\_T"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_UNEQUAL\_LEG\_ANGLE\_LU"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_INWARD\_UUESI"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_AND\_INWARD\_UUESOI"/>  
<enumeration value="PARAMETRIC\_THIN\_WALLED\_UNSYMMETRIC\_CHANNEL\_WITH\_EDGE\_STIFFENERS\_OUTWARD\_UUESO"/>

```
<enumeration value="PARAMETRIC_THIN_WALLED_UNSYMMETRIC_CHANNEL_UU"/>
<enumeration value="PARAMETRIC_THIN_WALLED_UNSYMMETRIC_I_SECTION_IU"/>
<enumeration value="PARAMETRIC_THIN_WALLED_UNSYMMETRIC_RECTANGULAR_HOLLOW_SECTION_RHSU"/>
<enumeration value="PARAMETRIC_THIN_WALLED_UNSYMMETRIC_T_SECTION_TU"/>
<enumeration value="PARAMETRIC_THIN_WALLED_Z_SECTION_Z"/>
<enumeration value="STANDARDIZED_STEEL_60_DEGREE_ANGLES_L60_S"/>
<enumeration value="STANDARDIZED_STEEL_BAR_SECTIONS_BAR_S"/>
<enumeration value="STANDARDIZED_STEEL_CIRCULAR_HOLLOW_SECTIONS_PIPES_CHS_S"/>
<enumeration value="STANDARDIZED_STEEL_ELLIPTICAL_AND_SEMI_ELLIPTICAL_HOLLOW_SECTIONS_EHS_S"/>
<enumeration value="STANDARDIZED_STEEL_I_BEAMS_H_BEAMS_I_S"/>
<enumeration value="STANDARDIZED_STEEL_L_SECTIONS_STEEL_ANGLES_L_S"/>
<enumeration value="STANDARDIZED_STEEL_RAIL_SECTIONS_RAIL_S"/>
<enumeration value="STANDARDIZED_STEEL_SHEETS_SHEET_S"/>
<enumeration value="STANDARDIZED_STEEL_SQUARE_AND_RECTANGULAR_HOLLOW_SECTIONS_RHS_S"/>
<enumeration value="STANDARDIZED_STEEL_T_SECTIONS_STEEL_TEES_T_S"/>
<enumeration value="STANDARDIZED_STEEL_U_SECTIONS_STEEL_CHANNELS_U_S"/>
<enumeration value="STANDARDIZED_STEEL_Z_SECTIONS_STEEL_ZEES_Z_S"/>
<enumeration value="STANDARDIZED_TIMBER_TIMBER_COMPOSED_SECTIONS_COMP_S"/>
<enumeration value="STANDARDIZED_TIMBER_TIMBER_RECTANGLES_RECT_S"/>
```

</restriction>

</simpleType>

<simpleType name="section\_combination\_type">

<restriction base="string">

```
<enumeration value="BUILT_UP_NG_2I"/>
<enumeration value="BUILT_UP_NG_2I_A"/>
<enumeration value="BUILT_UP_NG_2LCLI"/>
<enumeration value="BUILT_UP_NG_2LCLIR"/>
<enumeration value="BUILT_UP_NG_2LCLIR_A"/>
<enumeration value="BUILT_UP_NG_2LCLI_A"/>
<enumeration value="BUILT_UP_NG_2LCLO"/>
<enumeration value="BUILT_UP_NG_2LCLOC"/>
<enumeration value="BUILT_UP_NG_2LCLOC_A"/>
<enumeration value="BUILT_UP_NG_2LCLO_A"/>
<enumeration value="BUILT_UP_NG_2LHLO"/>
<enumeration value="BUILT_UP_NG_2LHLO_A"/>
<enumeration value="BUILT_UP_NG_2LHLO_A"/>
<enumeration value="BUILT_UP_NG_2LLHLO"/>
<enumeration value="BUILT_UP_NG_2LLHLO_A"/>
<enumeration value="BUILT_UP_NG_2LSHLO"/>
<enumeration value="BUILT_UP_NG_2LSHLO_A"/>
<enumeration value="BUILT_UP_NG_2U"/>
<enumeration value="BUILT_UP_NG_2UI"/>
<enumeration value="BUILT_UP_NG_2UI_A"/>
<enumeration value="BUILT_UP_NG_2UO"/>
<enumeration value="BUILT_UP_NG_2UO_A"/>
<enumeration value="BUILT_UP_NG_3I"/>
<enumeration value="BUILT_UP_NG_4LHLO"/>
<enumeration value="BUILT_UP_NG_4LHLO_A"/>
<enumeration value="BUILT_UP_NG_4LLI"/>
<enumeration value="BUILT_UP_NG_4LLI_A"/>
<enumeration value="BUILT_UP_NG_4LLO"/>
<enumeration value="BUILT_UP_NG_4LLO_A"/>
<enumeration value="BUILT_UP_NG_CIHF"/>
<enumeration value="BUILT_UP_NG_CIMD"/>
<enumeration value="BUILT_UP_NG_HNBH"/>
<enumeration value="BUILT_UP_NG_I1F"/>
<enumeration value="BUILT_UP_NG_I1HIC"/>
<enumeration value="BUILT_UP_NG_I1IC"/>
<enumeration value="BUILT_UP_NG_I2F"/>
<enumeration value="BUILT_UP_NG_I2HIC"/>
<enumeration value="BUILT_UP_NG_I2I"/>
<enumeration value="BUILT_UP_NG_ICI-MAX"/>
<enumeration value="BUILT_UP_NG_ICI-MIN"/>
<enumeration value="BUILT_UP_NG_IHF"/>
<enumeration value="BUILT_UP_NG_IUD"/>
<enumeration value="BUILT_UP_NG_IUU"/>
```

```

 <enumeration value="BUILT_UP_NG_IVF"/>
 <enumeration value="BUILT_UP_NG_IWT"/>
 <enumeration value="BUILT_UP_NG_UF"/>
 </restriction>
</simpleType>
<simpleType name="section_function_data_function_type">
 <restriction base="string">
 <enumeration value="FUNCTION_TYPE_CREEP_COEFFICIENT"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CA"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CD"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CS"/>
 </restriction>
</simpleType>
<complexType name="section_function_data_coefficients">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="time" minOccurs="0" type="double"/>
 <element name="coefficient" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_section_function_data_coefficients">
 <sequence>
 <element name="section_function_data_coefficients" type="xsd:section_function_data_coefficients" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_section_function_data_coefficients" type="xsd:array_of_section_function_data_coefficients"/>
<simpleType name="section_members_weight_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CWT"/>
 <enumeration value="COST_ESTIMATION_G"/>
 <enumeration value="COST_ESTIMATION_KG"/>
 <enumeration value="COST_ESTIMATION_LB"/>
 <enumeration value="COST_ESTIMATION_OZ"/>
 <enumeration value="COST_ESTIMATION_SLUG"/>
 <enumeration value="COST_ESTIMATION_T"/>
 <enumeration value="COST_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="section_members_volume_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM3"/>
 <enumeration value="COST_ESTIMATION_FLOZ"/>
 <enumeration value="COST_ESTIMATION_FT3"/>
 <enumeration value="COST_ESTIMATION_IMPGAL"/>
 <enumeration value="COST_ESTIMATION_IN3"/>
 <enumeration value="COST_ESTIMATION_L"/>
 <enumeration value="COST_ESTIMATION_M3"/>
 <enumeration value="COST_ESTIMATION_MM3"/>
 <enumeration value="COST_ESTIMATION_PT"/>
 <enumeration value="COST_ESTIMATION_QT"/>
 <enumeration value="COST_ESTIMATION_USGAL"/>
 <enumeration value="COST_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="section_members_surface_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM2"/>
 <enumeration value="COST_ESTIMATION_FT2"/>
 <enumeration value="COST_ESTIMATION_IN2"/>
 <enumeration value="COST_ESTIMATION_M2"/>
 <enumeration value="COST_ESTIMATION_MM2"/>
 <enumeration value="COST_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="section_members_length_unit">
 <restriction base="string">
 <enumeration value="COST_ESTIMATION_CM"/>
 <enumeration value="COST_ESTIMATION_FT"/>
 <enumeration value="COST_ESTIMATION_IN"/>
 <enumeration value="COST_ESTIMATION_KM"/>
 <enumeration value="COST_ESTIMATION_M"/>
 <enumeration value="COST_ESTIMATION_MM"/>
 <enumeration value="COST_ESTIMATION_YD"/>
 </restriction>
</simpleType>

```

```

<simpleType name="section_emissions_members_weight_unit">
 <restriction base="string">
 <enumeration value="EMISSIONS_ESTIMATION_CWT"/>
 <enumeration value="EMISSIONS_ESTIMATION_G"/>
 <enumeration value="EMISSIONS_ESTIMATION_KG"/>
 <enumeration value="EMISSIONS_ESTIMATION_LB"/>
 <enumeration value="EMISSIONS_ESTIMATION_OZ"/>
 <enumeration value="EMISSIONS_ESTIMATION_SLUG"/>
 <enumeration value="EMISSIONS_ESTIMATION_T"/>
 <enumeration value="EMISSIONS_ESTIMATION_TON"/>
 </restriction>
</simpleType>
<simpleType name="section_emissions_members_volume_unit">
 <restriction base="string">
 <enumeration value="EMISSIONS_ESTIMATION_CM3"/>
 <enumeration value="EMISSIONS_ESTIMATION_FLOZ"/>
 <enumeration value="EMISSIONS_ESTIMATION_FT3"/>
 <enumeration value="EMISSIONS_ESTIMATION_IMPGAL"/>
 <enumeration value="EMISSIONS_ESTIMATION_IN3"/>
 <enumeration value="EMISSIONS_ESTIMATION_L"/>
 <enumeration value="EMISSIONS_ESTIMATION_M3"/>
 <enumeration value="EMISSIONS_ESTIMATION_MM3"/>
 <enumeration value="EMISSIONS_ESTIMATION_PT"/>
 <enumeration value="EMISSIONS_ESTIMATION_QT"/>
 <enumeration value="EMISSIONS_ESTIMATION_USGAL"/>
 <enumeration value="EMISSIONS_ESTIMATION_YD3"/>
 </restriction>
</simpleType>
<simpleType name="section_emissions_members_surface_unit">
 <restriction base="string">
 <enumeration value="EMISSIONS_ESTIMATION_CM2"/>
 <enumeration value="EMISSIONS_ESTIMATION_FT2"/>
 <enumeration value="EMISSIONS_ESTIMATION_IN2"/>
 <enumeration value="EMISSIONS_ESTIMATION_M2"/>
 <enumeration value="EMISSIONS_ESTIMATION_MM2"/>
 <enumeration value="EMISSIONS_ESTIMATION_YD2"/>
 </restriction>
</simpleType>
<simpleType name="section_emissions_members_length_unit">
 <restriction base="string">
 <enumeration value="EMISSIONS_ESTIMATION_CM"/>
 <enumeration value="EMISSIONS_ESTIMATION_FT"/>
 <enumeration value="EMISSIONS_ESTIMATION_IN"/>
 <enumeration value="EMISSIONS_ESTIMATION_KM"/>
 <enumeration value="EMISSIONS_ESTIMATION_M"/>
 <enumeration value="EMISSIONS_ESTIMATION_MM"/>
 <enumeration value="EMISSIONS_ESTIMATION_YD"/>
 </restriction>
</simpleType>
<complexType name="get_section">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_section" type="xsd:get_section"/>
<complexType name="get_sectionResponse">
 <sequence>
 <element name="value" type="xsd:section"/>
 </sequence>
</complexType>
<element name="get_sectionResponse" type="xsd:get_sectionResponse"/>
<complexType name="set_section">
 <sequence>
 <element name="value" type="xsd:section"/>
 </sequence>
</complexType>
<element name="set_section" type="xsd:set_section"/>
<complexType name="set_sectionResponse">
 <sequence/>
</complexType>
<element name="set_sectionResponse" type="xsd:set_sectionResponse"/>
<complexType name="section">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:section_type"/>
 <element name="manufacturing_type" minOccurs="0" type="xsd:section_manufacturing_type"/>
 </sequence>

```

```
<element name="name" minOccurs="0" type="string"/>
<element name="shear_stiffness_deactivated" minOccurs="0" type="boolean"/>
<element name="warping_stiffness_deactivated" minOccurs="0" type="boolean"/>
<element name="thin_walled_model" minOccurs="0" type="boolean"/>
<element name="us_spelling_of_properites" minOccurs="0" type="boolean"/>
<element name="has_cost_estimation" minOccurs="0" type="boolean"/>
<element name="has_emissions_estimation" minOccurs="0" type="boolean"/>
<element name="optimization" minOccurs="0" type="boolean"/>
<element name="stress_smoothing_to_avoid_singularities" minOccurs="0" type="boolean"/>
<element name="area_axial" minOccurs="0" type="double"/>
<element name="area_shear_y" minOccurs="0" type="double"/>
<element name="area_shear_z" minOccurs="0" type="double"/>
<element name="inclination_principal_axes" minOccurs="0" type="double"/>
<element name="rotation_angle" minOccurs="0" type="double"/>
<element name="mirrored_axis_y" minOccurs="0" type="boolean"/>
<element name="mirrored_axis_z" minOccurs="0" type="boolean"/>
<element name="moment_of_inertia_bending_y" minOccurs="0" type="double"/>
<element name="moment_of_inertia_bending_z" minOccurs="0" type="double"/>
<element name="moment_of_inertia_torsion" minOccurs="0" type="double"/>
<element name="warping" minOccurs="0" type="double"/>
<element name="depth_temperature_load" minOccurs="0" type="double"/>
<element name="width_temperature_load" minOccurs="0" type="double"/>
<element name="material" minOccurs="0" type="int"/>
<element name="reference_material" minOccurs="0" type="int"/>
<element name="material_part_1" minOccurs="0" type="int"/>
<element name="material_part_2" minOccurs="0" type="int"/>
<element name="material_part_3" minOccurs="0" type="int"/>
<element name="material_part_4" minOccurs="0" type="int"/>
<element name="material_part_5" minOccurs="0" type="int"/>
<element name="material_part_6" minOccurs="0" type="int"/>
<element name="material_part_7" minOccurs="0" type="int"/>
<element name="material_part_8" minOccurs="0" type="int"/>
<element name="material_part_9" minOccurs="0" type="int"/>
<element name="material_part_10" minOccurs="0" type="int"/>
<element name="hybrid_active" minOccurs="0" type="boolean"/>
<element name="parametrization_type" minOccurs="0" type="xsd:section_parametrization_type"/>
<element name="combination_type" minOccurs="0" type="xsd:section_combination_type"/>
<element name="corrugated_sheet_width" minOccurs="0" type="double"/>
<element name="rail_worn_out" minOccurs="0" type="double"/>
<element name="phase_parent" minOccurs="0" type="int"/>
<element name="phase_parts" minOccurs="0" type="string"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="advanced_time_dependent_properties_of_concrete_enabled" minOccurs="0" type="boolean"/>
<element name="creep_enabled" minOccurs="0" type="boolean"/>
<element name="shrinkage_enabled" minOccurs="0" type="boolean"/>
<element name="relative_humidity" minOccurs="0" type="double"/>
<element name="function_data_function_type" minOccurs="0" type="xsd:section_function_data_function_type"/>
<element name="function_data_part_id" minOccurs="0" type="int"/>
<element name="function_data_age_of_concrete_at_the_considered_moment" minOccurs="0" type="double"/>
<element name="function_data_number_of_steps" minOccurs="0" type="int"/>
<element name="function_data_coefficients" minOccurs="0" type="xsd:array_of_section_function_data_coefficients"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="cost_estimation_apply_from_material" minOccurs="0" type="boolean"/>
<element name="members_weight_active" minOccurs="0" type="boolean"/>
<element name="members_weight_unit_cost" minOccurs="0" type="double"/>
<element name="members_weight_unit" minOccurs="0" type="xsd:section_members_weight_unit"/>
<element name="members_weight_quantity" minOccurs="0" type="double"/>
<element name="members_weight_cost" minOccurs="0" type="double"/>
<element name="members_volume_active" minOccurs="0" type="boolean"/>
<element name="members_volume_unit_cost" minOccurs="0" type="double"/>
<element name="members_volume_unit" minOccurs="0" type="xsd:section_members_volume_unit"/>
<element name="members_volume_quantity" minOccurs="0" type="double"/>
<element name="members_volume_cost" minOccurs="0" type="double"/>
<element name="members_surface_active" minOccurs="0" type="boolean"/>
<element name="members_surface_unit_cost" minOccurs="0" type="double"/>
<element name="members_surface_unit" minOccurs="0" type="xsd:section_members_surface_unit"/>
<element name="members_surface_quantity" minOccurs="0" type="double"/>
<element name="members_surface_cost" minOccurs="0" type="double"/>
<element name="members_length_active" minOccurs="0" type="boolean"/>
<element name="members_length_unit_cost" minOccurs="0" type="double"/>
<element name="members_length_unit" minOccurs="0" type="xsd:section_members_length_unit"/>
<element name="members_length_quantity" minOccurs="0" type="double"/>
```

```

<element name="members_length_cost" minOccurs="0" type="double"/>
<element name="sum_cost" minOccurs="0" type="double"/>
<element name="cost_percentage" minOccurs="0" type="double"/>
<element name="total_cost" minOccurs="0" type="double"/>
<element name="emissions_estimation_apply_from_material" minOccurs="0" type="boolean"/>
<element name="emissions_members_weight_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_weight_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_weight_unit" minOccurs="0" type="xsd:section_emissions_members_weight_unit"/>
<element name="emissions_members_weight_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_weight_cost" minOccurs="0" type="double"/>
<element name="emissions_members_volume_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_volume_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_volume_unit" minOccurs="0" type="xsd:section_emissions_members_volume_unit"/>
<element name="emissions_members_volume_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_volume_cost" minOccurs="0" type="double"/>
<element name="emissions_members_surface_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_surface_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_surface_unit" minOccurs="0" type="xsd:section_emissions_members_surface_unit"/>
<element name="emissions_members_surface_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_surface_cost" minOccurs="0" type="double"/>
<element name="emissions_members_length_active" minOccurs="0" type="boolean"/>
<element name="emissions_members_length_unit_cost" minOccurs="0" type="double"/>
<element name="emissions_members_length_unit" minOccurs="0" type="xsd:section_emissions_members_length_unit"/>
<element name="emissions_members_length_quantity" minOccurs="0" type="double"/>
<element name="emissions_members_length_cost" minOccurs="0" type="double"/>
<element name="emissions_sum_cost" minOccurs="0" type="double"/>
<element name="emissions_cost_percentage" minOccurs="0" type="double"/>
<element name="emissions_total_cost" minOccurs="0" type="double"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="thickness_type">
 <restriction base="string">
 <enumeration value="TYPE_LAYERS"/>
 <enumeration value="TYPE_SHAPE_ORTHOTROPY"/>
 <enumeration value="TYPE_STIFFNESS_MATRIX"/>
 <enumeration value="TYPE_THICKNESS_PHASE"/>
 <enumeration value="TYPE_UNIFORM"/>
 <enumeration value="TYPE_VARIABLE_CIRCLE"/>
 <enumeration value="TYPE_VARIABLE_FOUR_SURFACE_CORNERS"/>
 <enumeration value="TYPE_VARIABLE_THREE_NODES"/>
 <enumeration value="TYPE_VARIABLE_TWO_NODES_AND_DIRECTION"/>
 </restriction>
</simpleType>
<simpleType name="thickness_direction">
 <restriction base="string">
 <enumeration value="THICKNESS_DIRECTION_IN_SMALL_X"/>
 <enumeration value="THICKNESS_DIRECTION_IN_SMALL_Y"/>
 <enumeration value="THICKNESS_DIRECTION_IN_X"/>
 <enumeration value="THICKNESS_DIRECTION_IN_Y"/>
 <enumeration value="THICKNESS_DIRECTION_IN_Z"/>
 </restriction>
</simpleType>
<simpleType name="layer_type">
 <restriction base="string"/>
</simpleType>
<complexType name="thickness_layers_reference_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="layer_no" minOccurs="0" type="int"/>
 <element name="layer_type" minOccurs="0" type="xsd:layer_type"/>
 <element name="thickness_type" minOccurs="0" type="int"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="thickness" minOccurs="0" type="double"/>
 <element name="angle" minOccurs="0" type="double"/>
 <element name="connection_with_other_topological_elements" minOccurs="0" type="boolean"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="specific_weight" minOccurs="0" type="double"/>
 <element name="weight" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_thickness_layers_reference_table">
 <sequence>

```



```

<element name="thickness_layers_reference_table" type="xsd:thickness_layers_reference_table" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_thickness_layers_reference_table" type="xsd:array_of_thickness_layers_reference_table"/>
<simpleType name="thickness_function_data_function_type">
 <restriction base="string">
 <enumeration value="FUNCTION_TYPE_CREEP_COEFFICIENT"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CA"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CD"/>
 <enumeration value="FUNCTION_TYPE_SHRINKAGE_STRAIN_CS"/>
 </restriction>
</simpleType>
<complexType name="thickness_function_data_coefficients">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="time" minOccurs="0" type="double"/>
 <element name="coefficient" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_thickness_function_data_coefficients">
 <sequence>
 <element name="thickness_function_data_coefficients" type="xsd:thickness_function_data_coefficients" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_thickness_function_data_coefficients" type="xsd:array_of_thickness_function_data_coefficients"/>
<simpleType name="thickness_timber_frame_wall_framing_members_connector_type">
 <restriction base="string">
 <enumeration value="TYPE_NAIL"/>
 <enumeration value="TYPE_STAPLE"/>
 </restriction>
</simpleType>
<simpleType name="thickness_timber_frame_wall_framing_members_connector_dimension_type">
 <restriction base="string">
 <enumeration value="TYPE_0075_X_1060"/>
 <enumeration value="TYPE_0075_X_1130"/>
 <enumeration value="TYPE_0075_X_1300"/>
 <enumeration value="TYPE_0110_X_0610"/>
 <enumeration value="TYPE_0120_X_0200"/>
 <enumeration value="TYPE_0125_X_0620"/>
 <enumeration value="TYPE_0125_X_0770"/>
 <enumeration value="TYPE_0125_X_1060"/>
 <enumeration value="TYPE_0125_X_1130"/>
 <enumeration value="TYPE_0125_X_1250"/>
 <enumeration value="TYPE_0125_X_1300"/>
 <enumeration value="TYPE_0165_X_0280"/>
 </restriction>
</simpleType>
<simpleType name="thickness_timber_frame_wall_vertical_studs_distribution">
 <restriction base="string">
 <enumeration value="TYPE_CENTER_TO_CENTER"/>
 <enumeration value="TYPE_GAP"/>
 <enumeration value="TYPE_UNIFORM"/>
 <enumeration value="TYPE_USER_DEFINED"/>
 </restriction>
</simpleType>
<complexType name="thickness_timber_frame_wall_vertical_studs_spacing_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="spacing" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_thickness_timber_frame_wall_vertical_studs_spacing_table">
 <sequence>
 <element name="thickness_timber_frame_wall_vertical_studs_spacing_table"
type="xsd:thickness_timber_frame_wall_vertical_studs_spacing_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_thickness_timber_frame_wall_vertical_studs_spacing_table"
type="xsd:array_of_thickness_timber_frame_wall_vertical_studs_spacing_table"/>
<simpleType name="thickness_timber_frame_wall_vertical_studs_connector_type">
 <restriction base="string">
 <enumeration value="TYPE_NAIL"/>
 <enumeration value="TYPE_STAPLE"/>
 </restriction>
</simpleType>

```

```

<simpleType name="thickness_timber_frame_wall_vertical_studs_connector_dimension_type">
 <restriction base="string">
 <enumeration value="TYPE_0075_X_1060"/>
 <enumeration value="TYPE_0075_X_1130"/>
 <enumeration value="TYPE_0075_X_1300"/>
 <enumeration value="TYPE_0110_X_0610"/>
 <enumeration value="TYPE_0120_X_0200"/>
 <enumeration value="TYPE_0125_X_0620"/>
 <enumeration value="TYPE_0125_X_0770"/>
 <enumeration value="TYPE_0125_X_1060"/>
 <enumeration value="TYPE_0125_X_1130"/>
 <enumeration value="TYPE_0125_X_1250"/>
 <enumeration value="TYPE_0125_X_1300"/>
 <enumeration value="TYPE_0165_X_0280"/>
 </restriction>
</simpleType>
<simpleType name="thickness_timber_frame_wall_blockings_distribution">
 <restriction base="string">
 <enumeration value="TYPE_ALTEMATING"/>
 <enumeration value="TYPE_INLINE"/>
 <enumeration value="TYPE_STAGGERED"/>
 </restriction>
</simpleType>
<simpleType name="thickness_orthotropy_type">
 <restriction base="string">
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_BIDIRECTIONAL_RIBBED_PLATE"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_COUPLING"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_EFFECTIVE_THICKNESS"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_GRILLAGE"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_HOLLOW_CORE_SLAB"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_TRAPEZOIDAL_SHEET"/>
 <enumeration value="ORTHOTROPIC_THICKNESS_TYPE_UNIDIRECTIONAL_RIBBED_PLATE"/>
 </restriction>
</simpleType>
<simpleType name="thickness_shape_orthotropy_self_weight_definition_type">
 <restriction base="string">
 <enumeration value="SELF_WEIGHT_COMPUTED_FROM_PARAMETERS"/>
 <enumeration value="SELF_WEIGHT_DEFINED_VIA_FICTITIOUS_THICKNESS"/>
 <enumeration value="SELF_WEIGHT_DEFINED_VIA_WEIGHT"/>
 </restriction>
</simpleType>
<simpleType name="thickness_stiffness_matrix_self_weight_definition_type">
 <restriction base="string">
 <enumeration value="SELF_WEIGHT_DEFINITION_TYPE_DEFINED_VIA_BULK_DENSITY_AND_AREA_DENSITY"/>
 <enumeration value="SELF_WEIGHT_DEFINITION_TYPE_DEFINED_VIA_FICTITIOUS_THICKNESS_AND_AREA_DENSITY"/>
 <enumeration value="SELF_WEIGHT_DEFINITION_TYPE_DEFINED_VIA_FICTITIOUS_THICKNESS_AND_BULK_DENSITY"/>
 </restriction>
</simpleType>
<complexType name="get_thickness">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_thickness" type="xsd:get_thickness"/>
<complexType name="get_thicknessResponse">
 <sequence>
 <element name="value" type="xsd:thickness"/>
 </sequence>
</complexType>
<element name="get_thicknessResponse" type="xsd:get_thicknessResponse"/>
<complexType name="set_thickness">
 <sequence>
 <element name="value" type="xsd:thickness"/>
 </sequence>
</complexType>
<element name="set_thickness" type="xsd:set_thickness"/>
<complexType name="set_thicknessResponse">
 <sequence/>
</complexType>
<element name="set_thicknessResponse" type="xsd:set_thicknessResponse"/>
<complexType name="thickness">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="assigned_to_surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="type" minOccurs="0" type="xsd:thickness_type"/>
 </sequence>

```

```
<element name="material" minOccurs="0" type="int"/>
<element name="uniform_thickness" minOccurs="0" type="double"/>
<element name="advanced_time_dependent_properties_of_concrete_enabled" minOccurs="0" type="boolean"/>
<element name="creep_enabled" minOccurs="0" type="boolean"/>
<element name="shrinkage_enabled" minOccurs="0" type="boolean"/>
<element name="relative_humidity" minOccurs="0" type="double"/>
<element name="thickness_1" minOccurs="0" type="double"/>
<element name="node_1" minOccurs="0" type="int"/>
<element name="thickness_2" minOccurs="0" type="double"/>
<element name="node_2" minOccurs="0" type="int"/>
<element name="thickness_3" minOccurs="0" type="double"/>
<element name="node_3" minOccurs="0" type="int"/>
<element name="direction" minOccurs="0" type="xsd:thickness_direction"/>
<element name="thickness_4" minOccurs="0" type="double"/>
<element name="node_4" minOccurs="0" type="int"/>
<element name="thickness_circle_center" minOccurs="0" type="double"/>
<element name="thickness_circle_line" minOccurs="0" type="double"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="layers_reference_table" minOccurs="0" type="xsd:array_of_thickness_layers_reference_table"/>
<element name="layers_solid_model_enabled" minOccurs="0" type="boolean"/>
<element name="layers_gas_enabled" minOccurs="0" type="boolean"/>
<element name="layers_total_thickness" minOccurs="0" type="double"/>
<element name="layers_total_weight" minOccurs="0" type="double"/>
<element name="stiffness_reduction_enabled" minOccurs="0" type="boolean"/>
<element name="K33" minOccurs="0" type="double"/>
<element name="K44" minOccurs="0" type="double"/>
<element name="K55" minOccurs="0" type="double"/>
<element name="K88" minOccurs="0" type="double"/>
<element name="K33_note" minOccurs="0" type="string"/>
<element name="K44_note" minOccurs="0" type="string"/>
<element name="K55_note" minOccurs="0" type="string"/>
<element name="K88_note" minOccurs="0" type="string"/>
<element name="function_data_function_type" minOccurs="0" type="xsd:thickness_function_data_function_type"/>
<element name="function_data_layer_id" minOccurs="0" type="int"/>
<element name="function_data_age_of_concrete_at_the_considered_moment" minOccurs="0" type="double"/>
<element name="function_data_number_of_steps" minOccurs="0" type="int"/>
<element name="function_data_coefficients" minOccurs="0" type="xsd:array_of_thickness_function_data_coefficients"/>
<element name="parent_thickness" minOccurs="0" type="int"/>
<element name="parent_thickness_layers" minOccurs="0" type="xsd:array_of_int"/>
<element name="timber_frame_wall_sheathing_front_material_enabled" minOccurs="0" type="boolean"/>
<element name="timber_frame_wall_sheathing_front_material" minOccurs="0" type="int"/>
<element name="timber_frame_wall_sheathing_front_thickness" minOccurs="0" type="double"/>
<element name="timber_frame_wall_sheathing_back_material_enabled" minOccurs="0" type="boolean"/>
<element name="timber_frame_wall_sheathing_back_material" minOccurs="0" type="int"/>
<element name="timber_frame_wall_sheathing_back_thickness" minOccurs="0" type="double"/>
<element name="timber_frame_wall_horizontal_framing_members_cross_section" minOccurs="0" type="int"/>
<element name="timber_frame_wall_vertical_framing_members_cross_section" minOccurs="0" type="int"/>
<element name="timber_frame_wall_framing_members_connector_type" minOccurs="0"
type="xsd:thickness_timber_frame_wall_framing_members_connector_type"/>
<element name="timber_frame_wall_framing_members_connector_diameter" minOccurs="0" type="double"/>
<element name="timber_frame_wall_framing_members_connector_dimension_type" minOccurs="0"
type="xsd:thickness_timber_frame_wall_framing_members_connector_dimension_type"/>
<element name="timber_frame_wall_framing_members_connector_spacing" minOccurs="0" type="double"/>
<element name="timber_frame_wall_vertical_studs_cross_section" minOccurs="0" type="int"/>
<element name="timber_frame_wall_vertical_studs_distribution" minOccurs="0"
type="xsd:thickness_timber_frame_wall_vertical_studs_distribution"/>
<element name="timber_frame_wall_vertical_studs_reverse_distribution_enabled" minOccurs="0" type="boolean"/>
<element name="timber_frame_wall_vertical_studs_spacing" minOccurs="0" type="double"/>
<element name="timber_frame_wall_vertical_studs_spacing_table" minOccurs="0"
type="xsd:array_of_thickness_timber_frame_wall_vertical_studs_spacing_table"/>
<element name="timber_frame_wall_vertical_studs_spacing_definition_relative" minOccurs="0" type="boolean"/>
<element name="timber_frame_wall_vertical_studs_connector_type" minOccurs="0"
type="xsd:thickness_timber_frame_wall_vertical_studs_connector_type"/>
<element name="timber_frame_wall_vertical_studs_connector_diameter" minOccurs="0" type="double"/>
<element name="timber_frame_wall_vertical_studs_connector_dimension_type" minOccurs="0"
type="xsd:thickness_timber_frame_wall_vertical_studs_connector_dimension_type"/>
<element name="timber_frame_wall_vertical_studs_connector_spacing" minOccurs="0" type="double"/>
<element name="timber_frame_wall_blockings_cross_section_enabled" minOccurs="0" type="boolean"/>
<element name="timber_frame_wall_blockings_cross_section" minOccurs="0" type="int"/>
<element name="timber_frame_wall_blockings_distribution" minOccurs="0"
type="xsd:thickness_timber_frame_wall_blockings_distribution"/>
<element name="timber_frame_wall_blockings_offset" minOccurs="0" type="double"/>
<element name="orthotropy_type" minOccurs="0" type="xsd:thickness_orthotropy_type"/>
<element name="orthotropy_rotation_beta" minOccurs="0" type="double"/>
<element name="orthotropy_fictitious_thickness" minOccurs="0" type="double"/>
```

```
<element name="shape_orthotropy_self_weight_definition_type" minOccurs="0"
type="xsd:thickness_shape_orthotropy_self_weight_definition_type"/>
<element name="shape_orthotropy_effective_thickness_x" minOccurs="0" type="double"/>
<element name="shape_orthotropy_effective_thickness_y" minOccurs="0" type="double"/>
<element name="shape_orthotropy_self_weight" minOccurs="0" type="double"/>
<element name="stiffness_matrix_self_weight_definition_type" minOccurs="0"
type="xsd:thickness_stiffness_matrix_self_weight_definition_type"/>
<element name="stiffness_matrix_bulk_density" minOccurs="0" type="double"/>
<element name="stiffness_matrix_area_density" minOccurs="0" type="double"/>
<element name="stiffness_matrix_coefficient_of_thermal_expansion" minOccurs="0" type="double"/>
<element name="coupling_thickness" minOccurs="0" type="double"/>
<element name="coupling_spacing" minOccurs="0" type="double"/>
<element name="coupling_width" minOccurs="0" type="double"/>
<element name="slab_thickness" minOccurs="0" type="double"/>
<element name="rib_height" minOccurs="0" type="double"/>
<element name="rib_spacing" minOccurs="0" type="double"/>
<element name="rib_width" minOccurs="0" type="double"/>
<element name="rib_spacing_x" minOccurs="0" type="double"/>
<element name="rib_spacing_y" minOccurs="0" type="double"/>
<element name="rib_width_x" minOccurs="0" type="double"/>
<element name="rib_width_y" minOccurs="0" type="double"/>
<element name="sheet_thickness" minOccurs="0" type="double"/>
<element name="total_profile_height" minOccurs="0" type="double"/>
<element name="top_flange_width" minOccurs="0" type="double"/>
<element name="bottom_flange_width" minOccurs="0" type="double"/>
<element name="void_spacing" minOccurs="0" type="double"/>
<element name="void_diameter" minOccurs="0" type="double"/>
<element name="rib_height_x" minOccurs="0" type="double"/>
<element name="rib_height_y" minOccurs="0" type="double"/>
<element name="D11" minOccurs="0" type="double"/>
<element name="D11_note" minOccurs="0" type="string"/>
<element name="D12" minOccurs="0" type="double"/>
<element name="D12_note" minOccurs="0" type="string"/>
<element name="D13" minOccurs="0" type="double"/>
<element name="D13_note" minOccurs="0" type="string"/>
<element name="D16" minOccurs="0" type="double"/>
<element name="D16_note" minOccurs="0" type="string"/>
<element name="D17" minOccurs="0" type="double"/>
<element name="D17_note" minOccurs="0" type="string"/>
<element name="D18" minOccurs="0" type="double"/>
<element name="D18_note" minOccurs="0" type="string"/>
<element name="D22" minOccurs="0" type="double"/>
<element name="D22_note" minOccurs="0" type="string"/>
<element name="D23" minOccurs="0" type="double"/>
<element name="D23_note" minOccurs="0" type="string"/>
<element name="D27" minOccurs="0" type="double"/>
<element name="D27_note" minOccurs="0" type="string"/>
<element name="D28" minOccurs="0" type="double"/>
<element name="D28_note" minOccurs="0" type="string"/>
<element name="D33" minOccurs="0" type="double"/>
<element name="D33_note" minOccurs="0" type="string"/>
<element name="D38" minOccurs="0" type="double"/>
<element name="D38_note" minOccurs="0" type="string"/>
<element name="D44" minOccurs="0" type="double"/>
<element name="D44_note" minOccurs="0" type="string"/>
<element name="D45" minOccurs="0" type="double"/>
<element name="D45_note" minOccurs="0" type="string"/>
<element name="D55" minOccurs="0" type="double"/>
<element name="D55_note" minOccurs="0" type="string"/>
<element name="D66" minOccurs="0" type="double"/>
<element name="D66_note" minOccurs="0" type="string"/>
<element name="D67" minOccurs="0" type="double"/>
<element name="D67_note" minOccurs="0" type="string"/>
<element name="D68" minOccurs="0" type="double"/>
<element name="D68_note" minOccurs="0" type="string"/>
<element name="D77" minOccurs="0" type="double"/>
<element name="D77_note" minOccurs="0" type="string"/>
<element name="D78" minOccurs="0" type="double"/>
<element name="D78_note" minOccurs="0" type="string"/>
<element name="D88" minOccurs="0" type="double"/>
<element name="D88_note" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
```

```

 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="node_type">
 <restriction base="string">
 <enumeration value="TYPE_BETWEEN_TWO_NODES"/>
 <enumeration value="TYPE_BETWEEN_TWO_POINTS"/>
 <enumeration value="TYPE_ON_LINE"/>
 <enumeration value="TYPE_ON_MEMBER"/>
 <enumeration value="TYPE_STANDARD"/>
 </restriction>
</simpleType>
<simpleType name="node_coordinate_system_type">
 <restriction base="string">
 <enumeration value="COORDINATE_SYSTEM_CARTESIAN"/>
 <enumeration value="COORDINATE_SYSTEM_POLAR"/>
 <enumeration value="COORDINATE_SYSTEM_X_CYLINDRICAL"/>
 <enumeration value="COORDINATE_SYSTEM_Y_CYLINDRICAL"/>
 <enumeration value="COORDINATE_SYSTEM_Z_CYLINDRICAL"/>
 </restriction>
</simpleType>
<simpleType name="node_reference_type">
 <restriction base="string">
 <enumeration value="REFERENCE_TYPE_L"/>
 <enumeration value="REFERENCE_TYPE_XY"/>
 <enumeration value="REFERENCE_TYPE_XZ"/>
 <enumeration value="REFERENCE_TYPE_YZ"/>
 </restriction>
</simpleType>
<complexType name="get_node">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_node" type="xsd:get_node"/>
<complexType name="get_nodeResponse">
 <sequence>
 <element name="value" type="xsd:node"/>
 </sequence>
</complexType>
<element name="get_nodeResponse" type="xsd:get_nodeResponse"/>
<complexType name="set_node">
 <sequence>
 <element name="value" type="xsd:node"/>
 </sequence>
</complexType>
<element name="set_node" type="xsd:set_node"/>
<complexType name="set_nodeResponse">
 <sequence/>
</complexType>
<element name="set_nodeResponse" type="xsd:set_nodeResponse"/>
<complexType name="node">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:node_type"/>
 <element name="reference_node" minOccurs="0" type="int"/>
 <element name="coordinate_system" minOccurs="0" type="int"/>
 <element name="coordinate_system_type" minOccurs="0" type="xsd:coordinate_system_type"/>
 <element name="coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="coordinate_1" minOccurs="0" type="double"/>
 <element name="coordinate_2" minOccurs="0" type="double"/>
 <element name="coordinate_3" minOccurs="0" type="double"/>
 <element name="global_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="global_coordinate_1" minOccurs="0" type="double"/>
 <element name="global_coordinate_2" minOccurs="0" type="double"/>
 <element name="global_coordinate_3" minOccurs="0" type="double"/>
 <element name="between_two_nodes_start_node" minOccurs="0" type="int"/>
 <element name="between_two_points_start_point_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="between_two_points_start_point_coordinate_1" minOccurs="0" type="double"/>
 <element name="between_two_points_start_point_coordinate_2" minOccurs="0" type="double"/>
 <element name="between_two_points_start_point_coordinate_3" minOccurs="0" type="double"/>
 <element name="between_two_nodes_end_node" minOccurs="0" type="int"/>
 <element name="between_two_points_end_point_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="between_two_points_end_point_coordinate_1" minOccurs="0" type="double"/>
 <element name="between_two_points_end_point_coordinate_2" minOccurs="0" type="double"/>
 <element name="between_two_points_end_point_coordinate_3" minOccurs="0" type="double"/>
 </sequence>

```

```

<element name="on_line_reference_line" minOccurs="0" type="int"/>
<element name="on_member_reference_member" minOccurs="0" type="int"/>
<element name="reference_type" minOccurs="0" type="xsd:node_reference_type"/>
<element name="reference_object_projected_length" minOccurs="0" type="double"/>
<element name="distance_from_start_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="distance_from_start_relative" minOccurs="0" type="double"/>
<element name="distance_from_start_absolute" minOccurs="0" type="double"/>
<element name="offset_in_local_direction_y" minOccurs="0" type="double"/>
<element name="offset_in_local_direction_z" minOccurs="0" type="double"/>
<element name="offset_in_global_direction_x" minOccurs="0" type="double"/>
<element name="offset_in_global_direction_y" minOccurs="0" type="double"/>
<element name="offset_in_global_direction_z" minOccurs="0" type="double"/>
<element name="support" minOccurs="0" type="int"/>
<element name="mesh_refinement" minOccurs="0" type="int"/>
<element name="concrete_design_ultimate_configuration" minOccurs="0" type="int"/>
<element name="concrete_design_fire_configuration" minOccurs="0" type="int"/>
<element name="punching_design" minOccurs="0" type="boolean"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="line_type">
 <restriction base="string">
 <enumeration value="TYPE_ARC"/>
 <enumeration value="TYPE_CIRCLE"/>
 <enumeration value="TYPE_CUT_VIA_SECTION"/>
 <enumeration value="TYPE_CUT_VIA_TWO_LINES"/>
 <enumeration value="TYPE_ELLIPSE"/>
 <enumeration value="TYPE_ELLIPTICAL_ARC"/>
 <enumeration value="TYPE_NURBS"/>
 <enumeration value="TYPE_PARABOLA"/>
 <enumeration value="TYPE_POLYLINE"/>
 <enumeration value="TYPE_SPLINE"/>
 </restriction>
</simpleType>
<simpleType name="line_arc_alpha_adjustment_target">
 <restriction base="string">
 <enumeration value="ALPHA_ADJUSTMENT_TARGET_ARC_CONTROL_POINT"/>
 <enumeration value="ALPHA_ADJUSTMENT_TARGET_BEGINNING_OF_ARC"/>
 <enumeration value="ALPHA_ADJUSTMENT_TARGET_END_OF_ARC"/>
 </restriction>
</simpleType>
<complexType name="line_nurbs_control_points_by_components">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="global_coordinate_x" minOccurs="0" type="double"/>
 <element name="global_coordinate_y" minOccurs="0" type="double"/>
 <element name="global_coordinate_z" minOccurs="0" type="double"/>
 <element name="weight" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_line_nurbs_control_points_by_components">
 <sequence>
 <element name="line_nurbs_control_points_by_components" type="xsd:line_nurbs_control_points_by_components" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_line_nurbs_control_points_by_components" type="xsd:array_of_line_nurbs_control_points_by_components"/>
<complexType name="line_nurbs_control_points">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="global_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="weight" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_line_nurbs_control_points">
 <sequence>
 <element name="line_nurbs_control_points" type="xsd:line_nurbs_control_points" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_line_nurbs_control_points" type="xsd:array_of_line_nurbs_control_points"/>
<complexType name="array_of_line_nurbs_knots">

```

```

 <sequence>
 <element name="line_nurbs_knots" type="double" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_line_nurbs_knots" type="xsd:array_of_line_nurbs_knots"/>
 <simpleType name="line_rotation_specification_type">
 <restriction base="string">
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_ANGLE"/>
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_HELP_NODE"/>
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_INSIDE_NODE"/>
 </restriction>
 </simpleType>
 <simpleType name="line_rotation_plane">
 <restriction base="string">
 <enumeration value="ROTATION_PLANE_XY"/>
 <enumeration value="ROTATION_PLANE_XZ"/>
 </restriction>
 </simpleType>
 <complexType name="line_line_weld_assignment">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="weld" minOccurs="0" type="int"/>
 <element name="surface1" minOccurs="0" type="int"/>
 <element name="surface2" minOccurs="0" type="int"/>
 <element name="surface3" minOccurs="0" type="int"/>
 </sequence>
 </complexType>
 <complexType name="array_of_line_line_weld_assignment">
 <sequence>
 <element name="line_line_weld_assignment" type="xsd:line_line_weld_assignment" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_line_line_weld_assignment" type="xsd:array_of_line_line_weld_assignment"/>
 <complexType name="get_line">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
 </complexType>
 <element name="get_line" type="xsd:get_line"/>
 <complexType name="get_lineResponse">
 <sequence>
 <element name="value" type="xsd:line"/>
 </sequence>
 </complexType>
 <element name="get_lineResponse" type="xsd:get_lineResponse"/>
 <complexType name="set_line">
 <sequence>
 <element name="value" type="xsd:line"/>
 </sequence>
 </complexType>
 <element name="set_line" type="xsd:set_line"/>
 <complexType name="set_lineResponse">
 <sequence/>
 </complexType>
 <element name="set_lineResponse" type="xsd:set_lineResponse"/>
 <complexType name="line">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:line_type"/>
 <element name="definition_nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="arc_first_node" minOccurs="0" type="int"/>
 <element name="arc_second_node" minOccurs="0" type="int"/>
 <element name="arc_control_point" minOccurs="0" type="xsd:vector_3d"/>
 <element name="arc_control_point_x" minOccurs="0" type="double"/>
 <element name="arc_control_point_y" minOccurs="0" type="double"/>
 <element name="arc_control_point_z" minOccurs="0" type="double"/>
 <element name="arc_center" minOccurs="0" type="xsd:vector_3d"/>
 <element name="arc_center_x" minOccurs="0" type="double"/>
 <element name="arc_center_y" minOccurs="0" type="double"/>
 <element name="arc_center_z" minOccurs="0" type="double"/>
 <element name="arc_radius" minOccurs="0" type="double"/>
 <element name="arc_height" minOccurs="0" type="double"/>
 <element name="arc_alpha" minOccurs="0" type="double"/>
 <element name="arc_alpha_adjustment_target" minOccurs="0" type="xsd:line_arc_alpha_adjustment_target"/>
 </sequence>
 </complexType>

```

```
<element name="circle_center" minOccurs="0" type="xsd:vector 3d"/>
<element name="circle_center_coordinate_1" minOccurs="0" type="double"/>
<element name="circle_center_coordinate_2" minOccurs="0" type="double"/>
<element name="circle_center_coordinate_3" minOccurs="0" type="double"/>
<element name="circle_normal" minOccurs="0" type="xsd:vector 3d"/>
<element name="circle_normal_coordinate_1" minOccurs="0" type="double"/>
<element name="circle_normal_coordinate_2" minOccurs="0" type="double"/>
<element name="circle_normal_coordinate_3" minOccurs="0" type="double"/>
<element name="circle_rotation" minOccurs="0" type="double"/>
<element name="circle_node" minOccurs="0" type="xsd:vector 3d"/>
<element name="circle_radius" minOccurs="0" type="double"/>
<element name="ellipse_first_node" minOccurs="0" type="int"/>
<element name="ellipse_second_node" minOccurs="0" type="int"/>
<element name="ellipse_control_point" minOccurs="0" type="xsd:vector 3d"/>
<element name="ellipse_control_point_x" minOccurs="0" type="double"/>
<element name="ellipse_control_point_y" minOccurs="0" type="double"/>
<element name="ellipse_control_point_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_first_node" minOccurs="0" type="int"/>
<element name="elliptical_arc_second_node" minOccurs="0" type="int"/>
<element name="elliptical_arc_alpha" minOccurs="0" type="double"/>
<element name="elliptical_arc_beta" minOccurs="0" type="double"/>
<element name="elliptical_arc_normal" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_normal_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_normal_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_normal_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_major_radius" minOccurs="0" type="double"/>
<element name="elliptical_arc_minor_radius" minOccurs="0" type="double"/>
<element name="elliptical_arc_center" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_center_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_center_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_center_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_1" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_focus_1_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_1_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_1_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_2" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_focus_2_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_2_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_focus_2_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_first_control_point" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_first_control_point_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_first_control_point_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_first_control_point_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_second_control_point" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_second_control_point_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_second_control_point_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_second_control_point_z" minOccurs="0" type="double"/>
<element name="elliptical_arc_perimeter_control_point" minOccurs="0" type="xsd:vector 3d"/>
<element name="elliptical_arc_perimeter_control_point_x" minOccurs="0" type="double"/>
<element name="elliptical_arc_perimeter_control_point_y" minOccurs="0" type="double"/>
<element name="elliptical_arc_perimeter_control_point_z" minOccurs="0" type="double"/>
<element name="parabola_first_node" minOccurs="0" type="int"/>
<element name="parabola_second_node" minOccurs="0" type="int"/>
<element name="parabola_control_point" minOccurs="0" type="xsd:vector 3d"/>
<element name="parabola_control_point_x" minOccurs="0" type="double"/>
<element name="parabola_control_point_y" minOccurs="0" type="double"/>
<element name="parabola_control_point_z" minOccurs="0" type="double"/>
<element name="parabola_alpha" minOccurs="0" type="double"/>
<element name="parabola_focus_directrix_distance" minOccurs="0" type="double"/>
<element name="parabola_focus" minOccurs="0" type="xsd:vector 3d"/>
<element name="parabola_focus_x" minOccurs="0" type="double"/>
<element name="parabola_focus_y" minOccurs="0" type="double"/>
<element name="parabola_focus_z" minOccurs="0" type="double"/>
<element name="nurbs_order" minOccurs="0" type="int"/>
<element name="nurbs_control_points_by_components" minOccurs="0" type="xsd:array_of_line_nurbs_control_points_by_components"/>
<element name="nurbs_control_points" minOccurs="0" type="xsd:array_of_line_nurbs_control_points"/>
<element name="nurbs_knots" minOccurs="0" type="xsd:array_of_line_nurbs_knots"/>
<element name="rotation_specification_type" minOccurs="0" type="xsd:line_rotation_specification_type"/>
<element name="rotation_angle" minOccurs="0" type="double"/>
<element name="rotation_help_node" minOccurs="0" type="int"/>
<element name="rotation_plane" minOccurs="0" type="xsd:line_rotation_plane"/>
<element name="is_rotated" minOccurs="0" type="boolean"/>
```



```

<element name="member" minOccurs="0" type="int"/>
<element name="support" minOccurs="0" type="int"/>
<element name="mesh_refinement" minOccurs="0" type="int"/>
<element name="line_weld_assignment" minOccurs="0" type="xsd:array_of_line_line_weld_assignment"/>
<element name="has_line_welds" minOccurs="0" type="boolean"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="member_type">
<restriction base="string">
<enumeration value="TYPE_BEAM"/>
<enumeration value="TYPE_BUCKLING"/>
<enumeration value="TYPE_CABLE"/>
<enumeration value="TYPE_COMPRESSION"/>
<enumeration value="TYPE_COUPLING_HINGE_HINGE"/>
<enumeration value="TYPE_COUPLING_HINGE_RIGID"/>
<enumeration value="TYPE_COUPLING_RIGID_HINGE"/>
<enumeration value="TYPE_COUPLING_RIGID_RIGID"/>
<enumeration value="TYPE_DEFINABLE_STIFFNESS"/>
<enumeration value="TYPE_RESULT_BEAM"/>
<enumeration value="TYPE_RIB"/>
<enumeration value="TYPE_RIGID"/>
<enumeration value="TYPE_TENSION"/>
<enumeration value="TYPE_TRUSS"/>
<enumeration value="TYPE_TRUSS_ONLY_N"/>
</restriction>
</simpleType>
<simpleType name="member_section_distribution_type">
<restriction base="string">
<enumeration value=""/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_LINEAR"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_SADDLE"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_UNIFORM"/>
</restriction>
</simpleType>
<simpleType name="member_reference_type">
<restriction base="string">
<enumeration value="REFERENCE_TYPE_L"/>
<enumeration value="REFERENCE_TYPE_XY"/>
<enumeration value="REFERENCE_TYPE_XZ"/>
<enumeration value="REFERENCE_TYPE_YZ"/>
</restriction>
</simpleType>
<simpleType name="member_member_type_rib_alignment">
<restriction base="string">
<enumeration value="ALIGNMENT_CENTRIC"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_NEGATIVE"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_POSITIVE"/>
<enumeration value="ALIGNMENT_USER_DEFINED_VIA_MEMBER_ECCENTRICITY"/>
</restriction>
</simpleType>
<simpleType name="reference_length_definition_type">
<restriction base="string">
<enumeration value="REFERENCE_LENGTH_TYPE_MEMBER_LENGTH"/>
<enumeration value="REFERENCE_LENGTH_TYPE_SEGMENT_LENGTH"/>
<enumeration value="REFERENCE_LENGTH_TYPE_USER_DEFINED"/>
</restriction>
</simpleType>
<simpleType name="reference_length_width_type">
<restriction base="string">
<enumeration value="REFERENCE_LENGTH_WIDTH_EC2"/>
<enumeration value="REFERENCE_LENGTH_WIDTH_EIGHTH"/>
<enumeration value="REFERENCE_LENGTH_WIDTH_NONE"/>
<enumeration value="REFERENCE_LENGTH_WIDTH_SIXTH"/>
</restriction>
</simpleType>
<simpleType name="distribution_linearity_type">

```

```

 <restriction base="string"/>
 </simpleType>
 <complexType name="member_flange_dimensions">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="end_ordinate" minOccurs="0" type="double"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="reference_length_definition_type" minOccurs="0" type="xsd:reference_length_definition_type"/>
 <element name="reference_length" minOccurs="0" type="double"/>
 <element name="reference_length_width_type" minOccurs="0" type="xsd:reference_length_width_type"/>
 <element name="width_minus_y_integrative" minOccurs="0" type="double"/>
 <element name="width_minus_y_effictive" minOccurs="0" type="double"/>
 <element name="width_minus_y_maximal" minOccurs="0" type="double"/>
 <element name="width_plus_y_integrative" minOccurs="0" type="double"/>
 <element name="width_plus_y_effictive" minOccurs="0" type="double"/>
 <element name="width_plus_y_maximal" minOccurs="0" type="double"/>
 <element name="distribution_linearity_type" minOccurs="0" type="xsd:distribution_linearity_type"/>
 </sequence>
 </complexType>
 <complexType name="array_of_member_flange_dimensions">
 <sequence>
 <element name="member_flange_dimensions" type="xsd:member_flange_dimensions" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_member_flange_dimensions" type="xsd:array_of_member_flange_dimensions"/>
 <simpleType name="member_result_beam_integrate_stresses_and_forces">
 <restriction base="string">
 <enumeration value="INTEGRATE_FROM_LISTED_OBJECT"/>
 <enumeration value="INTEGRATE_WITHIN_CUBOID_GENERAL"/>
 <enumeration value="INTEGRATE_WITHIN_CUBOID_QUADRATIC"/>
 <enumeration value="INTEGRATE_WITHIN_CYLINDER"/>
 </restriction>
 </simpleType>
 <simpleType name="member_section_alignment">
 <restriction base="string">
 <enumeration value="SECTION_ALIGNMENT_BOTTOM"/>
 <enumeration value="SECTION_ALIGNMENT_CENTRIC"/>
 <enumeration value="SECTION_ALIGNMENT_TOP"/>
 </restriction>
 </simpleType>
 <simpleType name="member_curved_member_cantilevers_type">
 <restriction base="string">
 <enumeration value="CANTILEVERS_TYPE_HORIZONTAL"/>
 <enumeration value="CANTILEVERS_TYPE_OFFSET"/>
 <enumeration value="CANTILEVERS_TYPE_PARALLEL"/>
 <enumeration value="CANTILEVERS_TYPE_TAPER"/>
 </restriction>
 </simpleType>
 <simpleType name="member_rotation_specification_type">
 <restriction base="string">
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_ANGLE"/>
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_HELP_NODE"/>
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_INSIDE_NODE"/>
 <enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_SURFACE"/>
 </restriction>
 </simpleType>
 <simpleType name="member_rotation_plane_type">
 <restriction base="string">
 <enumeration value="ROTATION_PLANE_XY"/>
 <enumeration value="ROTATION_PLANE_XZ"/>
 </restriction>
 </simpleType>
 <simpleType name="member_rotation_surface_plane_type">
 <restriction base="string">
 <enumeration value="ROTATION_PLANE_XY"/>
 <enumeration value="ROTATION_PLANE_XZ"/>
 </restriction>
 </simpleType>
 <simpleType name="stirrup_type">
 <restriction base="string">
 <enumeration value="STIRRUP_TYPE_FOUR_LEGGED_CLOSED_HOOK_135"/>
 <enumeration value="STIRRUP_TYPE_FOUR_LEGGED_CLOSED_HOOK_90"/>
 <enumeration value="STIRRUP_TYPE_FOUR_LEGGED_OVERLAP_HOOK_180"/>
 <enumeration value="STIRRUP_TYPE_THREE_LEGGED_CLOSED_HOOK_135"/>
 <enumeration value="STIRRUP_TYPE_THREE_LEGGED_CLOSED_HOOK_90"/>
 <enumeration value="STIRRUP_TYPE_THREE_LEGGED_OVERLAP_HOOK_180"/>
 </restriction>
 </simpleType>

```

```

 <enumeration value="STIRRUP_TYPE_TWO_LEGGED_CLOSED_HOOK_135"/>
 <enumeration value="STIRRUP_TYPE_TWO_LEGGED_CLOSED_HOOK_90"/>
 <enumeration value="STIRRUP_TYPE_TWO_LEGGED_OPEN"/>
 <enumeration value="STIRRUP_TYPE_TWO_LEGGED_OVERLAP_HOOK_180"/>
 </restriction>
</simpleType>
<simpleType name="stirrup_bar_size_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="span_position_reference_type">
 <restriction base="string">
 <enumeration value="SHEAR_REINFORCEMENT_SPAN_INTERNAL_NODE"/>
 <enumeration value="SHEAR_REINFORCEMENT_SPAN_REFERENCE_END"/>
 <enumeration value="SHEAR_REINFORCEMENT_SPAN_REFERENCE_START"/>
 </restriction>
</simpleType>
<simpleType name="span_position_definition_format_type">
 <restriction base="string">
 <enumeration value="SHEAR_REINFORCEMENT_SPAN_DEFINITION_FORMAT_ABSOLUTE"/>
 <enumeration value="SHEAR_REINFORCEMENT_SPAN_DEFINITION_FORMAT_RELATIVE"/>
 </restriction>
</simpleType>
<simpleType name="stirrup_layout_rule_type">
 <restriction base="string">
 <enumeration value="SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_END_DEFINED"/>
 <enumeration
value="SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_END_EQUALS_REST_LENGTH_TO_STIRRUP_DISTANCED"/>
 <enumeration value="SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_DEFINED"/>
 <enumeration value="SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_EQUALS_END"/>
 <enumeration
value="SHEAR_REINFORCEMENT_STIRRUP_LAYOUT_RULE_START_EQUALS_REST_LENGTH_TO_STIRRUP_DISTANCED"/>
 </restriction>
</simpleType>
<complexType name="member_concrete_shear_reinforcement_spans">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="stirrup_type" minOccurs="0" type="xsd:stirrup_type"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="stirrup_count" minOccurs="0" type="int"/>
 <element name="stirrup_diameter" minOccurs="0" type="double"/>
 <element name="stirrup_bar_size_type" minOccurs="0" type="xsd:stirrup_bar_size_type"/>
 <element name="stirrup_distances" minOccurs="0" type="double"/>
 <element name="reinforcement_area" minOccurs="0" type="double"/>
 <element name="span_position_reference_type" minOccurs="0" type="xsd:span_position_reference_type"/>
 <element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
 <element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
 <element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
 <element name="span_position_definition_format_type" minOccurs="0" type="xsd:span_position_definition_format_type"/>
 <element name="span_start_relative" minOccurs="0" type="double"/>
 <element name="span_start_absolute" minOccurs="0" type="double"/>
 <element name="span_end_relative" minOccurs="0" type="double"/>
 <element name="span_end_absolute" minOccurs="0" type="double"/>
 <element name="span_length" minOccurs="0" type="double"/>
 <element name="stirrup_layout_rule_type" minOccurs="0" type="xsd:stirrup_layout_rule_type"/>
 <element name="stirrup_start_offset" minOccurs="0" type="double"/>
 <element name="stirrup_end_offset" minOccurs="0" type="double"/>
 <element name="one_stirrup_length" minOccurs="0" type="double"/>
 <element name="minimal_and_maximal_stirrup_length" minOccurs="0" type="string"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="one_stirrup_weight" minOccurs="0" type="double"/>
 <element name="minimal_and_maximal_stirrup_weight" minOccurs="0" type="string"/>
 <element name="weight" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_concrete_shear_reinforcement_spans">
 <sequence>
 <element name="member_concrete_shear_reinforcement_spans" type="xsd:member_concrete_shear_reinforcement_spans"
minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_concrete_shear_reinforcement_spans" type="xsd:array_of_member_concrete_shear_reinforcement_spans"/>
<simpleType name="rebar_type">
 <restriction base="string">
 <enumeration value="REBAR_TYPE_LINE"/>
 <enumeration value="REBAR_TYPE_SINGLE"/>
 </restriction>

```

```

 <enumeration value="REBAR_TYPE_SYMMETRICAL"/>
 <enumeration value="REBAR_TYPE_UNIFORMLY_SURROUNDING"/>
 <enumeration value="REBAR_TYPE_UNSYMMETRICAL"/>
 </restriction>
</simpleType>
<simpleType name="bar_size_designation_symmetrical_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_unsymmetrical_at_side_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_unsymmetrical_top_side_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_unsymmetrical_bottom_side_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_uniformly_surrounding_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_line_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_single_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="bar_size_designation_corner_type">
 <restriction base="string"/>
</simpleType>
<simpleType name="additional_offset_type">
 <restriction base="string">
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_CONCRETE_COVER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_SECTION_SURFACE"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_STIRRUP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_NONE"/>
 </restriction>
</simpleType>
<simpleType name="additional_offset_type_single_line_type">
 <restriction base="string">
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_CONCRETE_COVER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_SECTION_SURFACE"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_FROM_STIRRUP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_TYPE_NONE"/>
 </restriction>
</simpleType>
<simpleType name="additional_offset_reference_type">
 <restriction base="string">
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_TOP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_TOP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_TOP"/>
 </restriction>
</simpleType>
<simpleType name="additional_offset_reference_type_at_start_type">
 <restriction base="string">
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_TOP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_TOP"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_TOP"/>
 </restriction>
</simpleType>
<simpleType name="additional_offset_reference_type_at_end_type">
 <restriction base="string">
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_BOTTOM"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_CENTER"/>
 <enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_CENTER_TOP"/>
 </restriction>
</simpleType>

```

```

<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_BOTTOM"/>
<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_CENTER"/>
<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_LEFT_TOP"/>
<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_BOTTOM"/>
<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_CENTER"/>
<enumeration value="LONGITUDINAL_REINFORCEMENT_ADDITIONAL_OFFSET_REFERENCE_TYPE_RIGHT_TOP"/>
</restriction>
</simpleType>
<simpleType name="anchorage_start_anchor_type">
 <restriction base="string">
 <enumeration value="ANCHORAGE_TYPE_BEND"/>
 <enumeration value="ANCHORAGE_TYPE_HOOK"/>
 <enumeration value="ANCHORAGE_TYPE_HOOK_WITH_TRANSVERSE_BAR"/>
 <enumeration value="ANCHORAGE_TYPE_NONE"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT_WITH_TRANSVERSE_BAR"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT_WITH_TWO_TRANSVERSE_BARS"/>
 </restriction>
</simpleType>
<simpleType name="anchorage_end_anchor_type">
 <restriction base="string">
 <enumeration value="ANCHORAGE_TYPE_BEND"/>
 <enumeration value="ANCHORAGE_TYPE_HOOK"/>
 <enumeration value="ANCHORAGE_TYPE_HOOK_WITH_TRANSVERSE_BAR"/>
 <enumeration value="ANCHORAGE_TYPE_NONE"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT_WITH_TRANSVERSE_BAR"/>
 <enumeration value="ANCHORAGE_TYPE_STRAIGHT_WITH_TWO_TRANSVERSE_BARS"/>
 </restriction>
</simpleType>
<complexType name="member_concrete_longitudinal_reinforcement_items">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="rebar_type" minOccurs="0" type="xsd:rebar_type"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="reinforcement_placed_in_bending_corner_enabled" minOccurs="0" type="boolean"/>
 <element name="bar_count_symmetrical" minOccurs="0" type="int"/>
 <element name="bar_count_unsymmetrical_at_side" minOccurs="0" type="int"/>
 <element name="bar_count_unsymmetrical_top_side" minOccurs="0" type="int"/>
 <element name="bar_count_unsymmetrical_bottom_side" minOccurs="0" type="int"/>
 <element name="bar_count_uniformly_surrounding" minOccurs="0" type="int"/>
 <element name="bar_count_line" minOccurs="0" type="int"/>
 <element name="bar_count_corner" minOccurs="0" type="int"/>
 <element name="bar_diameter_symmetrical" minOccurs="0" type="double"/>
 <element name="bar_diameter_unsymmetrical_at_side" minOccurs="0" type="double"/>
 <element name="bar_diameter_unsymmetrical_top_side" minOccurs="0" type="double"/>
 <element name="bar_diameter_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
 <element name="bar_diameter_uniformly_surrounding" minOccurs="0" type="double"/>
 <element name="bar_diameter_line" minOccurs="0" type="double"/>
 <element name="bar_diameter_single" minOccurs="0" type="double"/>
 <element name="bar_diameter_corner" minOccurs="0" type="double"/>
 <element name="bar_size_designation_symmetrical_type" minOccurs="0" type="xsd:bar_size_designation_symmetrical_type"/>
 <element name="bar_size_designation_unsymmetrical_at_side_type" minOccurs="0" type="xsd:bar_size_designation_unsymmetrical_at_side_type"/>
 <element name="bar_size_designation_unsymmetrical_top_side_type" minOccurs="0" type="xsd:bar_size_designation_unsymmetrical_top_side_type"/>
 <element name="bar_size_designation_unsymmetrical_bottom_side_type" minOccurs="0" type="xsd:bar_size_designation_unsymmetrical_bottom_side_type"/>
 <element name="bar_size_designation_uniformly_surrounding_type" minOccurs="0" type="xsd:bar_size_designation_uniformly_surrounding_type"/>
 <element name="bar_size_designation_line_type" minOccurs="0" type="xsd:bar_size_designation_line_type"/>
 <element name="bar_size_designation_single_type" minOccurs="0" type="xsd:bar_size_designation_single_type"/>
 <element name="bar_size_designation_corner_type" minOccurs="0" type="xsd:bar_size_designation_corner_type"/>
 <element name="corner_reinforcement_enabled" minOccurs="0" type="boolean"/>
 <element name="reinforcement_area_symmetrical" minOccurs="0" type="double"/>
 <element name="reinforcement_area_unsymmetrical_at_side" minOccurs="0" type="double"/>
 <element name="reinforcement_area_unsymmetrical_top_side" minOccurs="0" type="double"/>
 <element name="reinforcement_area_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
 <element name="reinforcement_area_uniformly_surrounding" minOccurs="0" type="double"/>
 <element name="reinforcement_area_line" minOccurs="0" type="double"/>
 <element name="reinforcement_area_single" minOccurs="0" type="double"/>
 <element name="reinforcement_area_corner" minOccurs="0" type="double"/>
 <element name="reinforcement_area_total" minOccurs="0" type="double"/>
 <element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
 </sequence>
</complexType>

```

```

<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="additional_offset_type" minOccurs="0" type="xsd:additional_offset_type"/>
<element name="additional_offset_type_single_line_type" minOccurs="0" type="xsd:additional_offset_type_single_line_type"/>
<element name="additional_offset_top_side" minOccurs="0" type="double"/>
<element name="additional_offset_bottom_side" minOccurs="0" type="double"/>
<element name="additional_offset_left_side" minOccurs="0" type="double"/>
<element name="additional_offset_right_side" minOccurs="0" type="double"/>
<element name="additional_offset_reference_type" minOccurs="0" type="xsd:additional_offset_reference_type"/>
<element name="additional_offset_reference_type_at_start_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_start_type"/>
<element name="additional_offset_reference_type_at_end_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_end_type"/>
<element name="additional_horizontal_offset" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_end" minOccurs="0" type="double"/>
<element name="additional_vertical_offset" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_end" minOccurs="0" type="double"/>
<element name="anchorage_start_anchor_type" minOccurs="0" type="xsd:anchorage_start_anchor_type"/>
<element name="anchorage_start_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_start_bending_diameter" minOccurs="0" type="double"/>
<element name="anchorage_end_anchor_type" minOccurs="0" type="xsd:anchorage_end_anchor_type"/>
<element name="anchorage_end_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_end_bending_diameter" minOccurs="0" type="double"/>
<element name="one_rebar_length" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_corner_length" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_rebar_weight" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_corner_weight" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="member_concrete_longitudinal_reinforcement_items" type="xsd:member_concrete_longitudinal_reinforcement_items"
minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_concrete_longitudinal_reinforcement_items"
type="xsd:array_of_member_concrete_longitudinal_reinforcement_items"/>
<simpleType name="member_deflection_check_direction">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z_AND_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_RESULTING_AXIS"/>
</restriction>
</simpleType>
<simpleType name="member_deflection_check_displacement_reference">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_SEGMENT_ENDS"/>
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_UNDEFORMED_SYSTEM"/>
</restriction>
</simpleType>

```

```

<complexType name="member_design_supports_on_internal_nodes">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="node" minOccurs="0" type="int"/>
 <element name="design_support" minOccurs="0" type="int"/>
 </sequence>
</complexType>
<complexType name="array_of_member_design_supports_on_internal_nodes">
 <sequence>
 <element name="member_design_supports_on_internal_nodes" type="xsd:member_design_supports_on_internal_nodes"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_design_supports_on_internal_nodes" type="xsd:array_of_member_design_supports_on_internal_nodes"/>
<complexType name="member_deflection_segments_y_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_deflection_segments_y_axis">
 <sequence>
 <element name="member_deflection_segments_y_axis" type="xsd:member_deflection_segments_y_axis" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_deflection_segments_y_axis" type="xsd:array_of_member_deflection_segments_y_axis"/>
<complexType name="member_deflection_segments_z_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_deflection_segments_z_axis">
 <sequence>
 <element name="member_deflection_segments_z_axis" type="xsd:member_deflection_segments_z_axis" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_deflection_segments_z_axis" type="xsd:array_of_member_deflection_segments_z_axis"/>
<complexType name="get_member">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_member" type="xsd:get_member"/>
<complexType name="get_memberResponse">
 <sequence>
 <element name="value" type="xsd:member"/>
 </sequence>
</complexType>
<element name="get_memberResponse" type="xsd:get_memberResponse"/>
<complexType name="set_member">
 <sequence>
 <element name="value" type="xsd:member"/>
 </sequence>
</complexType>
<element name="set_member" type="xsd:set_member"/>
<complexType name="set_memberResponse">
 <sequence/>
</complexType>
<element name="set_memberResponse" type="xsd:set_memberResponse"/>
<complexType name="member">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:member_type"/>
 <element name="is_deactivated_for_calculation" minOccurs="0" type="boolean"/>
 <element name="line" minOccurs="0" type="int"/>
 <element name="section_distribution_type" minOccurs="0" type="xsd:member_section_distribution_type"/>
 <element name="reference_type" minOccurs="0" type="xsd:member_reference_type"/>
 <element name="nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="node_start" minOccurs="0" type="int"/>
 <element name="node_end" minOccurs="0" type="int"/>
 <element name="analytical_length" minOccurs="0" type="double"/>
 </sequence>

```

```
<element name="analytical_volume" minOccurs="0" type="double"/>
<element name="analytical_surface_of_coating" minOccurs="0" type="double"/>
<element name="analytical_mass" minOccurs="0" type="double"/>
<element name="surface_of_coating" minOccurs="0" type="double"/>
<element name="analytical_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="analytical_center_of_gravity_x" minOccurs="0" type="double"/>
<element name="analytical_center_of_gravity_y" minOccurs="0" type="double"/>
<element name="analytical_center_of_gravity_z" minOccurs="0" type="double"/>
<element name="length" minOccurs="0" type="double"/>
<element name="volume" minOccurs="0" type="double"/>
<element name="mass" minOccurs="0" type="double"/>
<element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="center_of_gravity_x" minOccurs="0" type="double"/>
<element name="center_of_gravity_y" minOccurs="0" type="double"/>
<element name="center_of_gravity_z" minOccurs="0" type="double"/>
<element name="member_representative" minOccurs="0" type="int"/>
<element name="design_properties_via_member" minOccurs="0" type="boolean"/>
<element name="design_properties_via_parent_member_set" minOccurs="0" type="boolean"/>
<element name="design_properties_parent_member_set" minOccurs="0" type="int"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="member_type_rib_alignment" minOccurs="0" type="xsd:member_member_type_rib_alignment"/>
<element name="member_rib_first_surface" minOccurs="0" type="int"/>
<element name="member_rib_second_surface" minOccurs="0" type="int"/>
<element name="member_rib_surface_assignment_autodetect" minOccurs="0" type="boolean"/>
<element name="align_local_z_axis_to_local_z_axis_of_surface" minOccurs="0" type="boolean"/>
<element name="flange_dimensions" minOccurs="0" type="xsd:array_of_member_flange_dimensions"/>
<element name="synchronize_width_mode" minOccurs="0" type="boolean"/>
<element name="relative_ordinates_mode" minOccurs="0" type="boolean"/>
<element name="member_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled" minOccurs="0" type="boolean"/>
<element name="member_type_definable_stiffness" minOccurs="0" type="int"/>
<element name="result_beam_integrate_stresses_and_forces" minOccurs="0" type="xsd:member_result_beam_integrate_stresses_and_forces"/>
<element name="result_beam_y_z" minOccurs="0" type="double"/>
<element name="result_beam_y_plus" minOccurs="0" type="double"/>
<element name="result_beam_z_plus" minOccurs="0" type="double"/>
<element name="result_beam_y_minus" minOccurs="0" type="double"/>
<element name="result_beam_z_minus" minOccurs="0" type="double"/>
<element name="result_beam_radius" minOccurs="0" type="double"/>
<element name="result_beam_include_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_surfaces" minOccurs="0" type="boolean"/>
<element name="result_beam_include_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_solids" minOccurs="0" type="boolean"/>
<element name="result_beam_include_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_members" minOccurs="0" type="boolean"/>
<element name="result_beam_exclude_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_exclude_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_exclude_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="projected_length" minOccurs="0" type="double"/>
<element name="section_distance_from_start_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="section_distance_from_start_absolute" minOccurs="0" type="double"/>
<element name="section_distance_from_start_relative" minOccurs="0" type="double"/>
<element name="section_distance_from_end_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="section_distance_from_end_absolute" minOccurs="0" type="double"/>
<element name="section_distance_from_end_relative" minOccurs="0" type="double"/>
<element name="section_alignment" minOccurs="0" type="xsd:member_section_alignment"/>
<element name="section_start" minOccurs="0" type="int"/>
<element name="section_end" minOccurs="0" type="int"/>
<element name="section_internal" minOccurs="0" type="int"/>
<element name="section_material" minOccurs="0" type="int"/>
<element name="curved_member_is_asymmetric_layout" minOccurs="0" type="boolean"/>
<element name="curved_member_is_cantilevers" minOccurs="0" type="boolean"/>
<element name="curved_member_cantilevers_type" minOccurs="0" type="xsd:member_curved_member_cantilevers_type"/>
<element name="rotation_specification_type" minOccurs="0" type="xsd:member_rotation_specification_type"/>
<element name="rotation_angle" minOccurs="0" type="double"/>
<element name="rotation_help_node" minOccurs="0" type="int"/>
<element name="rotation_plane_type" minOccurs="0" type="xsd:member_rotation_plane_type"/>
<element name="rotation_surface" minOccurs="0" type="int"/>
<element name="rotation_surface_plane_type" minOccurs="0" type="xsd:member_rotation_surface_plane_type"/>
<element name="is_rotated" minOccurs="0" type="boolean"/>
<element name="member_hinge_start" minOccurs="0" type="int"/>
<element name="member_hinge_end" minOccurs="0" type="int"/>
<element name="member_eccentricity_start" minOccurs="0" type="int"/>
<element name="member_eccentricity_end" minOccurs="0" type="int"/>
```



```
<element name="support" minOccurs="0" type="int"/>
<element name="member_transverse_stiffener" minOccurs="0" type="int"/>
<element name="member_nonlinearity" minOccurs="0" type="int"/>
<element name="member_result_intermediate_point" minOccurs="0" type="int"/>
<element name="stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="aluminum_effective_lengths" minOccurs="0" type="int"/>
<element name="aluminum_boundary_conditions" minOccurs="0" type="int"/>
<element name="aluminum_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="aluminum_member_transverse_weld" minOccurs="0" type="int"/>
<element name="aluminum_member_shear_panel" minOccurs="0" type="int"/>
<element name="aluminum_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_aluminum_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_aluminum_design_sls_configuration" minOccurs="0" type="int"/>
<element name="concrete_effective_lengths" minOccurs="0" type="int"/>
<element name="concrete_cover_user_defined_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover" minOccurs="0" type="double"/>
<element name="concrete_cover_top" minOccurs="0" type="double"/>
<element name="concrete_cover_left" minOccurs="0" type="double"/>
<element name="concrete_cover_right" minOccurs="0" type="double"/>
<element name="concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="concrete_cover_min" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_top" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_left" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_right" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_bottom" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_different_at_section_sides_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_durability" minOccurs="0" type="int"/>
<element name="concrete_durability_top" minOccurs="0" type="int"/>
<element name="concrete_durability_left" minOccurs="0" type="int"/>
<element name="concrete_durability_right" minOccurs="0" type="int"/>
<element name="concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="concrete_shear_reinforcement_spans" minOccurs="0"
type="xsd:array_of_member_concrete_shear_reinforcement_spans"/>
<element name="concrete_longitudinal_reinforcement_items" minOccurs="0"
type="xsd:array_of_member_concrete_longitudinal_reinforcement_items"/>
<element name="member_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="steel_effective_lengths" minOccurs="0" type="int"/>
<element name="steel_boundary_conditions" minOccurs="0" type="int"/>
<element name="steel_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="steel_member_transverse_weld" minOccurs="0" type="int"/>
<element name="steel_member_shear_panel" minOccurs="0" type="int"/>
<element name="steel_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_steel_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_fr_configuration" minOccurs="0" type="int"/>
<element name="timber_effective_lengths" minOccurs="0" type="int"/>
<element name="timber_service_class" minOccurs="0" type="int"/>
<element name="timber_moisture_class" minOccurs="0" type="int"/>
<element name="timber_service_conditions" minOccurs="0" type="int"/>
<element name="timber_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="timber_member_shear_panel" minOccurs="0" type="int"/>
<element name="timber_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_timber_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_fr_configuration" minOccurs="0" type="int"/>
<element name="end_modifications_member_start_extension" minOccurs="0" type="double"/>
<element name="end_modifications_member_start_slope_y" minOccurs="0" type="double"/>
<element name="end_modifications_member_start_slope_z" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_extension" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_slope_y" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_slope_z" minOccurs="0" type="double"/>
<element name="has_any_end_modifications" minOccurs="0" type="boolean"/>
<element name="deflection_check_direction" minOccurs="0" type="xsd:member_deflection_check_direction"/>
<element name="deflection_check_displacement_reference" minOccurs="0"
type="xsd:member_deflection_check_displacement_reference"/>
<element name="design_support_on_member_start" minOccurs="0" type="int"/>
<element name="design_support_on_member_end" minOccurs="0" type="int"/>
<element name="design_supports_on_internal_nodes" minOccurs="0"
type="xsd:array_of_member_design_supports_on_internal_nodes"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
```

```

<element name="deflection_segments_defined_length_y_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_defined_length_z_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_y_axis" minOccurs="0" type="xsd:array_of_member_deflection_segments_y_axis"/>
<element name="deflection_segments_z_axis" minOccurs="0" type="xsd:array_of_member_deflection_segments_z_axis"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="member_representative_type">
<restriction base="string">
<enumeration value="TYPE_BEAM"/>
<enumeration value="TYPE_BUCKLING"/>
<enumeration value="TYPE_CABLE"/>
<enumeration value="TYPE_COMPRESSION"/>
<enumeration value="TYPE_COUPLING_HINGE_HINGE"/>
<enumeration value="TYPE_COUPLING_HINGE_RIGID"/>
<enumeration value="TYPE_COUPLING_RIGID_HINGE"/>
<enumeration value="TYPE_COUPLING_RIGID_RIGID"/>
<enumeration value="TYPE_DEFINABLE_STIFFNESS"/>
<enumeration value="TYPE_RESULT_BEAM"/>
<enumeration value="TYPE_RIB"/>
<enumeration value="TYPE_RIGID"/>
<enumeration value="TYPE_TENSION"/>
<enumeration value="TYPE_TRUSS"/>
<enumeration value="TYPE_TRUSS_ONLY_N"/>
</restriction>
</simpleType>
<simpleType name="member_representative_section_distribution_type">
<restriction base="string">
<enumeration value=""/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_LINEAR"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_SADDLE"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_UNIFORM"/>
</restriction>
</simpleType>
<simpleType name="member_representative_reference_type">
<restriction base="string">
<enumeration value="REFERENCE_TYPE_L"/>
<enumeration value="REFERENCE_TYPE_XY"/>
<enumeration value="REFERENCE_TYPE_XZ"/>
<enumeration value="REFERENCE_TYPE_YZ"/>
</restriction>
</simpleType>
<simpleType name="member_representative_member_type_rib_alignment">
<restriction base="string">
<enumeration value="ALIGNMENT_CENTRIC"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_NEGATIVE"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_POSITIVE"/>
<enumeration value="ALIGNMENT_USER_DEFINED_VIA_MEMBER_ECCENTRICITY"/>
</restriction>
</simpleType>
<simpleType name="member_representative_result_beam_integrate_stresses_and_forces">
<restriction base="string">
<enumeration value="INTEGRATE_FROM_LISTED_OBJECT"/>
<enumeration value="INTEGRATE_WITHIN_CUBOID_GENERAL"/>
<enumeration value="INTEGRATE_WITHIN_CUBOID_QUADRATIC"/>
<enumeration value="INTEGRATE_WITHIN_CYLINDER"/>
</restriction>
</simpleType>
<complexType name="member_representative_flange_dimensions">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="end_ordinate" minOccurs="0" type="double"/>
<element name="length" minOccurs="0" type="double"/>
<element name="reference_length_definition_type" minOccurs="0" type="xsd:reference_length_definition_type"/>
<element name="reference_length" minOccurs="0" type="double"/>
<element name="reference_length_width_type" minOccurs="0" type="xsd:reference_length_width_type"/>
<element name="width_minus_y_integrative" minOccurs="0" type="double"/>
<element name="width_minus_y_effictive" minOccurs="0" type="double"/>

```

```

<element name="width_minus_y_maximal" minOccurs="0" type="double"/>
<element name="width_plus_y_integrative" minOccurs="0" type="double"/>
<element name="width_plus_y_effective" minOccurs="0" type="double"/>
<element name="width_plus_y_maximal" minOccurs="0" type="double"/>
<element name="distribution_linearity_type" minOccurs="0" type="xsd:distribution_linearity_type"/>
</sequence>
</complexType>
<complexType name="array_of_member_representative_flange_dimensions">
<sequence>
<element name="member_representative_flange_dimensions" type="xsd:member_representative_flange_dimensions" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_representative_flange_dimensions" type="xsd:array_of_member_representative_flange_dimensions"/>
<simpleType name="member_representative_section_alignment">
<restriction base="string">
<enumeration value="SECTION_ALIGNMENT_BOTTOM"/>
<enumeration value="SECTION_ALIGNMENT_CENTRIC"/>
<enumeration value="SECTION_ALIGNMENT_TOP"/>
</restriction>
</simpleType>
<simpleType name="member_representative_curved_member_cantilevers_type">
<restriction base="string">
<enumeration value="CANTILEVERS_TYPE_HORIZONTAL"/>
<enumeration value="CANTILEVERS_TYPE_OFFSET"/>
<enumeration value="CANTILEVERS_TYPE_PARALLEL"/>
<enumeration value="CANTILEVERS_TYPE_TAPER"/>
</restriction>
</simpleType>
<simpleType name="member_representative_rotation_specification_type">
<restriction base="string">
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_ANGLE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_HELP_NODE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_INSIDE_NODE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_SURFACE"/>
</restriction>
</simpleType>
<simpleType name="member_representative_rotation_plane_type">
<restriction base="string">
<enumeration value="ROTATION_PLANE_XY"/>
<enumeration value="ROTATION_PLANE_XZ"/>
</restriction>
</simpleType>
<simpleType name="member_representative_rotation_surface_plane_type">
<restriction base="string">
<enumeration value="ROTATION_PLANE_XY"/>
<enumeration value="ROTATION_PLANE_XZ"/>
</restriction>
</simpleType>
<complexType name="member_representative_concrete_shear_reinforcement_spans">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="stirrup_type" minOccurs="0" type="xsd:stirrup_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="stirrup_count" minOccurs="0" type="int"/>
<element name="stirrup_diameter" minOccurs="0" type="double"/>
<element name="stirrup_bar_size_type" minOccurs="0" type="xsd:stirrup_bar_size_type"/>
<element name="stirrup_distances" minOccurs="0" type="double"/>
<element name="reinforcement_area" minOccurs="0" type="double"/>
<element name="span_position_reference_type" minOccurs="0" type="xsd:span_position_reference_type"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_position_definition_format_type" minOccurs="0" type="xsd:span_position_definition_format_type"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="stirrup_layout_rule_type" minOccurs="0" type="xsd:stirrup_layout_rule_type"/>
<element name="stirrup_start_offset" minOccurs="0" type="double"/>
<element name="stirrup_end_offset" minOccurs="0" type="double"/>
<element name="one_stirrup_length" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>

```

```

<element name="one_stirrup_weight" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_representative_concrete_shear_reinforcement_spans">
<sequence>
<element name="member_representative_concrete_shear_reinforcement_spans"
type="xsd:member_representative_concrete_shear_reinforcement_spans" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_representative_concrete_shear_reinforcement_spans"
type="xsd:array_of_member_representative_concrete_shear_reinforcement_spans"/>
<complexType name="member_representative_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="rebar_type" minOccurs="0" type="xsd:rebar_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="reinforcement_placed_in_bending_corner_enabled" minOccurs="0" type="boolean"/>
<element name="bar_count_symmetrical" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_at_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_top_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_bottom_side" minOccurs="0" type="int"/>
<element name="bar_count_uniformly_surrounding" minOccurs="0" type="int"/>
<element name="bar_count_line" minOccurs="0" type="int"/>
<element name="bar_count_corner" minOccurs="0" type="int"/>
<element name="bar_diameter_symmetrical" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="bar_diameter_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="bar_diameter_line" minOccurs="0" type="double"/>
<element name="bar_diameter_single" minOccurs="0" type="double"/>
<element name="bar_diameter_corner" minOccurs="0" type="double"/>
<element name="bar_size_designation_symmetrical_type" minOccurs="0" type="xsd:bar_size_designation_symmetrical_type"/>
<element name="bar_size_designation_unsymmetrical_at_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_at_side_type"/>
<element name="bar_size_designation_unsymmetrical_top_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_top_side_type"/>
<element name="bar_size_designation_unsymmetrical_bottom_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_bottom_side_type"/>
<element name="bar_size_designation_uniformly_surrounding_type" minOccurs="0"
type="xsd:bar_size_designation_uniformly_surrounding_type"/>
<element name="bar_size_designation_line_type" minOccurs="0" type="xsd:bar_size_designation_line_type"/>
<element name="bar_size_designation_single_type" minOccurs="0" type="xsd:bar_size_designation_single_type"/>
<element name="bar_size_designation_corner_type" minOccurs="0" type="xsd:bar_size_designation_corner_type"/>
<element name="corner_reinforcement_enabled" minOccurs="0" type="boolean"/>
<element name="reinforcement_area_symmetrical" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="reinforcement_area_line" minOccurs="0" type="double"/>
<element name="reinforcement_area_single" minOccurs="0" type="double"/>
<element name="reinforcement_area_corner" minOccurs="0" type="double"/>
<element name="reinforcement_area_total" minOccurs="0" type="double"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="additional_offset_type" minOccurs="0" type="xsd:additional_offset_type"/>
<element name="additional_offset_type_single_line_type" minOccurs="0" type="xsd:additional_offset_type_single_line_type"/>
<element name="additional_offset_top_side" minOccurs="0" type="double"/>
<element name="additional_offset_bottom_side" minOccurs="0" type="double"/>
<element name="additional_offset_left_side" minOccurs="0" type="double"/>
<element name="additional_offset_right_side" minOccurs="0" type="double"/>
<element name="additional_offset_reference_type" minOccurs="0" type="xsd:additional_offset_reference_type"/>
<element name="additional_offset_reference_type_at_start_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_start_type"/>
<element name="additional_offset_reference_type_at_end_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_end_type"/>

```

```

<element name="additional_horizontal_offset" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_end" minOccurs="0" type="double"/>
<element name="additional_vertical_offset" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_end" minOccurs="0" type="double"/>
<element name="anchorage_start_anchor_type" minOccurs="0" type="xsd:anchorage_start_anchor_type"/>
<element name="anchorage_start_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_start_bending_diameter" minOccurs="0" type="double"/>
<element name="anchorage_end_anchor_type" minOccurs="0" type="xsd:anchorage_end_anchor_type"/>
<element name="anchorage_end_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_end_bending_diameter" minOccurs="0" type="double"/>
<element name="one_rebar_length" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_corner_length" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_rebar_weight" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_corner_weight" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_representative_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="member_representative_concrete_longitudinal_reinforcement_items"
type="xsd:member_representative_concrete_longitudinal_reinforcement_items" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_representative_concrete_longitudinal_reinforcement_items"
type="xsd:array_of_member_representative_concrete_longitudinal_reinforcement_items"/>
<simpleType name="member_representative_deflection_check_direction">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z_AND_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_RESULTING_AXIS"/>
</restriction>
</simpleType>
<simpleType name="member_representative_deflection_check_displacement_reference">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_SEGMENT_ENDS"/>
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_UNDEFORMED_SYSTEM"/>
</restriction>
</simpleType>
<complexType name="member_representative_design_supports_on_internal_nodes">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="node" minOccurs="0" type="int"/>
<element name="design_support" minOccurs="0" type="int"/>
</sequence>
</complexType>
<complexType name="array_of_member_representative_design_supports_on_internal_nodes">
<sequence>
<element name="member_representative_design_supports_on_internal_nodes"
type="xsd:member_representative_design_supports_on_internal_nodes" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_representative_design_supports_on_internal_nodes"
type="xsd:array_of_member_representative_design_supports_on_internal_nodes"/>
<complexType name="member_representative_deflection_segments_y_axis">
<sequence>
<element name="no" minOccurs="0" type="int"/>

```

```

 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_representative_deflection_segments_y_axis">
 <sequence>
 <element name="member_representative_deflection_segments_y_axis" type="xsd:member_representative_deflection_segments_y_axis"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_representative_deflection_segments_y_axis"
 type="xsd:array_of_member_representative_deflection_segments_y_axis"/>
<complexType name="member_representative_deflection_segments_z_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_representative_deflection_segments_z_axis">
 <sequence>
 <element name="member_representative_deflection_segments_z_axis" type="xsd:member_representative_deflection_segments_z_axis"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_representative_deflection_segments_z_axis"
 type="xsd:array_of_member_representative_deflection_segments_z_axis"/>
<complexType name="get_member_representative">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_member_representative" type="xsd:get_member_representative"/>
<complexType name="get_member_representativeResponse">
 <sequence>
 <element name="value" type="xsd:member_representative"/>
 </sequence>
</complexType>
<element name="get_member_representativeResponse" type="xsd:get_member_representativeResponse"/>
<complexType name="set_member_representative">
 <sequence>
 <element name="value" type="xsd:member_representative"/>
 </sequence>
</complexType>
<element name="set_member_representative" type="xsd:set_member_representative"/>
<complexType name="set_member_representativeResponse">
 <sequence/>
</complexType>
<element name="set_member_representativeResponse" type="xsd:set_member_representativeResponse"/>
<complexType name="member_representative">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:member_representative_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="number_of_members" minOccurs="0" type="int"/>
 <element name="total_length" minOccurs="0" type="double"/>
 <element name="total_volume" minOccurs="0" type="double"/>
 <element name="total_mass" minOccurs="0" type="double"/>
 <element name="total_surface_of_coating" minOccurs="0" type="double"/>
 <element name="nodes_on_member_from_start" minOccurs="0" type="xsd:array_of_int"/>
 <element name="is_deactivated_for_calculation" minOccurs="0" type="boolean"/>
 <element name="line" minOccurs="0" type="int"/>
 <element name="section_distribution_type" minOccurs="0" type="xsd:member_representative_section_distribution_type"/>
 <element name="reference_type" minOccurs="0" type="xsd:member_representative_reference_type"/>
 <element name="nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="node_start" minOccurs="0" type="int"/>
 <element name="node_end" minOccurs="0" type="int"/>
 <element name="analytical_length" minOccurs="0" type="double"/>
 <element name="analytical_volume" minOccurs="0" type="double"/>
 <element name="analytical_mass" minOccurs="0" type="double"/>
 <element name="analytical_surface_of_coating" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="analytical_center_of_gravity_x" minOccurs="0" type="double"/>
 </sequence>

```

```
<element name="analytical_center_of_gravity_y" minOccurs="0" type="double"/>
<element name="analytical_center_of_gravity_z" minOccurs="0" type="double"/>
<element name="length" minOccurs="0" type="double"/>
<element name="volume" minOccurs="0" type="double"/>
<element name="mass" minOccurs="0" type="double"/>
<element name="surface_of_coating" minOccurs="0" type="double"/>
<element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="center_of_gravity_x" minOccurs="0" type="double"/>
<element name="center_of_gravity_y" minOccurs="0" type="double"/>
<element name="center_of_gravity_z" minOccurs="0" type="double"/>
<element name="member_representative" minOccurs="0" type="int"/>
<element name="design_properties_via_member" minOccurs="0" type="boolean"/>
<element name="design_properties_via_parent_member_set" minOccurs="0" type="boolean"/>
<element name="design_properties_parent_member_set" minOccurs="0" type="int"/>
<element name="member_type_rib_alignment" minOccurs="0" type="xsd:member_representative_member_type_rib_alignment"/>
<element name="member_rib_first_surface" minOccurs="0" type="int"/>
<element name="member_rib_second_surface" minOccurs="0" type="int"/>
<element name="member_rib_surface_assignment_autodetect" minOccurs="0" type="boolean"/>
<element name="member_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled" minOccurs="0" type="boolean"/>
<element name="align_local_z_axis_to_local_z_axis_of_surface" minOccurs="0" type="boolean"/>
<element name="result_beam_integrate_stresses_and_forces" minOccurs="0"
type="xsd:member_representative_result_beam_integrate_stresses_and_forces"/>
<element name="flange_dimensions" minOccurs="0" type="xsd:array_of_member_representative_flange_dimensions"/>
<element name="synchronize_width_mode" minOccurs="0" type="boolean"/>
<element name="relative_ordinates_mode" minOccurs="0" type="boolean"/>
<element name="member_type_definable_stiffness" minOccurs="0" type="int"/>
<element name="result_beam_y_z" minOccurs="0" type="double"/>
<element name="result_beam_y_plus" minOccurs="0" type="double"/>
<element name="result_beam_z_plus" minOccurs="0" type="double"/>
<element name="result_beam_y_minus" minOccurs="0" type="double"/>
<element name="result_beam_z_minus" minOccurs="0" type="double"/>
<element name="result_beam_radius" minOccurs="0" type="double"/>
<element name="result_beam_include_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_surfaces" minOccurs="0" type="boolean"/>
<element name="result_beam_include_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_solids" minOccurs="0" type="boolean"/>
<element name="result_beam_include_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_include_all_members" minOccurs="0" type="boolean"/>
<element name="result_beam_exclude_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_exclude_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="result_beam_exclude_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="projected_length" minOccurs="0" type="double"/>
<element name="section_distance_from_start_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="section_distance_from_start_absolute" minOccurs="0" type="double"/>
<element name="section_distance_from_start_relative" minOccurs="0" type="double"/>
<element name="section_distance_from_end_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="section_distance_from_end_absolute" minOccurs="0" type="double"/>
<element name="section_distance_from_end_relative" minOccurs="0" type="double"/>
<element name="section_alignment" minOccurs="0" type="xsd:member_representative_section_alignment"/>
<element name="curved_member_is_asymmetric_layout" minOccurs="0" type="boolean"/>
<element name="curved_member_is_cantilevers" minOccurs="0" type="boolean"/>
<element name="curved_member_cantilevers_type" minOccurs="0"
type="xsd:member_representative_curved_member_cantilevers_type"/>
<element name="rotation_specification_type" minOccurs="0" type="xsd:member_representative_rotation_specification_type"/>
<element name="rotation_angle" minOccurs="0" type="double"/>
<element name="rotation_help_node" minOccurs="0" type="int"/>
<element name="rotation_plane_type" minOccurs="0" type="xsd:member_representative_rotation_plane_type"/>
<element name="rotation_surface" minOccurs="0" type="int"/>
<element name="rotation_surface_plane_type" minOccurs="0" type="xsd:member_representative_rotation_surface_plane_type"/>
<element name="is_rotated" minOccurs="0" type="boolean"/>
<element name="section_start" minOccurs="0" type="int"/>
<element name="section_end" minOccurs="0" type="int"/>
<element name="section_internal" minOccurs="0" type="int"/>
<element name="section_material" minOccurs="0" type="int"/>
<element name="members" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_hinge_start" minOccurs="0" type="int"/>
<element name="member_hinge_end" minOccurs="0" type="int"/>
<element name="member_eccentricity_start" minOccurs="0" type="int"/>
<element name="member_eccentricity_end" minOccurs="0" type="int"/>
<element name="support" minOccurs="0" type="int"/>
<element name="member_transverse_stiffener" minOccurs="0" type="int"/>
<element name="member_nonlinearity" minOccurs="0" type="int"/>
<element name="member_result_intermediate_point" minOccurs="0" type="int"/>
<element name="concrete_effective_lengths" minOccurs="0" type="int"/>
```

```
<element name="steel_effective_lengths" minOccurs="0" type="int"/>
<element name="timber_effective_lengths" minOccurs="0" type="int"/>
<element name="aluminum_effective_lengths" minOccurs="0" type="int"/>
<element name="concrete_shear_reinforcement_spans" minOccurs="0"
type="xsd:array_of_member_representative_concrete_shear_reinforcement_spans"/>
<element name="concrete_longitudinal_reinforcement_items" minOccurs="0"
type="xsd:array_of_member_representative_concrete_longitudinal_reinforcement_items"/>
<element name="concrete_cover_user_defined_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover_different_at_section_sides_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover" minOccurs="0" type="double"/>
<element name="concrete_cover_top" minOccurs="0" type="double"/>
<element name="concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="concrete_cover_left" minOccurs="0" type="double"/>
<element name="concrete_cover_right" minOccurs="0" type="double"/>
<element name="concrete_cover_min" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_top" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_bottom" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_left" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_right" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_durability" minOccurs="0" type="int"/>
<element name="concrete_durability_top" minOccurs="0" type="int"/>
<element name="concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="concrete_durability_left" minOccurs="0" type="int"/>
<element name="concrete_durability_right" minOccurs="0" type="int"/>
<element name="steel_boundary_conditions" minOccurs="0" type="int"/>
<element name="steel_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="steel_member_transverse_weld" minOccurs="0" type="int"/>
<element name="steel_member_shear_panel" minOccurs="0" type="int"/>
<element name="steel_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="timber_service_class" minOccurs="0" type="int"/>
<element name="timber_moisture_class" minOccurs="0" type="int"/>
<element name="timber_service_conditions" minOccurs="0" type="int"/>
<element name="timber_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="timber_member_shear_panel" minOccurs="0" type="int"/>
<element name="timber_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="aluminum_boundary_conditions" minOccurs="0" type="int"/>
<element name="aluminum_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="aluminum_member_transverse_weld" minOccurs="0" type="int"/>
<element name="aluminum_member_shear_panel" minOccurs="0" type="int"/>
<element name="aluminum_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="member_aluminum_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_aluminum_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_fr_configuration" minOccurs="0" type="int"/>
<element name="end_modifications_member_start_extension" minOccurs="0" type="double"/>
<element name="end_modifications_member_start_slope_y" minOccurs="0" type="double"/>
<element name="end_modifications_member_start_slope_z" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_extension" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_slope_y" minOccurs="0" type="double"/>
<element name="end_modifications_member_end_slope_z" minOccurs="0" type="double"/>
<element name="has_any_end_modifications" minOccurs="0" type="boolean"/>
<element name="deflection_check_direction" minOccurs="0" type="xsd:member_representative_deflection_check_direction"/>
<element name="deflection_check_displacement_reference" minOccurs="0"
type="xsd:member_representative_deflection_check_displacement_reference"/>
<element name="design_support_on_member_start" minOccurs="0" type="int"/>
<element name="design_support_on_member_end" minOccurs="0" type="int"/>
<element name="design_supports_on_internal_nodes" minOccurs="0"
type="xsd:array_of_member_representative_design_supports_on_internal_nodes"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="deflection_segments_defined_length_y_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_defined_length_z_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_y_axis" minOccurs="0"
type="xsd:array_of_member_representative_deflection_segments_y_axis"/>
<element name="deflection_segments_z_axis" minOccurs="0"
type="xsd:array_of_member_representative_deflection_segments_z_axis"/>
```



```

 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="surface_geometry">
 <restriction base="string">
 <enumeration value="GEOMETRY_NURBS"/>
 <enumeration value="GEOMETRY_PIPE"/>
 <enumeration value="GEOMETRY_PLANE"/>
 <enumeration value="GEOMETRY_QUADRANGLE"/>
 <enumeration value="GEOMETRY_ROTATED"/>
 </restriction>
</simpleType>
<simpleType name="surface_type">
 <restriction base="string">
 <enumeration value="TYPE_GROUNDWATER"/>
 <enumeration value="TYPE_LOAD_TRANSFER"/>
 <enumeration value="TYPE_MEMBRANE"/>
 <enumeration value="TYPE_RIGID"/>
 <enumeration value="TYPE_STANDARD"/>
 <enumeration value="TYPE_WITHOUT_MEMBRANE_TENSION"/>
 <enumeration value="TYPE_WITHOUT_THICKNESS"/>
 </restriction>
</simpleType>
<simpleType name="surface_load_transfer_direction">
 <restriction base="string">
 <enumeration value="LOAD_TRANSFER_DIRECTION_IN_BOTH"/>
 <enumeration value="LOAD_TRANSFER_DIRECTION_IN_X"/>
 <enumeration value="LOAD_TRANSFER_DIRECTION_IN_Y"/>
 </restriction>
</simpleType>
<complexType name="surface_nurbs_control_points">
 <sequence>
 <element name="u" minOccurs="0" type="int"/>
 <element name="v" minOccurs="0" type="int"/>
 <element name="global_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="weight" type="double"/>
 </sequence>
</complexType>
<complexType name="surface_nurbs_control_points_row">
 <sequence>
 <element name="surface_nurbs_control_points_row" type="xsd:surface_nurbs_control_points" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="surface_nurbs_control_points_row" type="xsd:surface_nurbs_control_points_row"/>
<complexType name="array_of_surface_nurbs_control_points_rows">
 <sequence>
 <element name="surface_nurbs_control_points_rows" type="xsd:surface_nurbs_control_points_row" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_surface_nurbs_control_points_rows" type="xsd:array_of_surface_nurbs_control_points_rows"/>
<simpleType name="surface_meshing_type">
 <restriction base="string">
 <enumeration value="MESHING_TYPE_FREE"/>
 <enumeration value="MESHING_TYPE_MAPPED"/>
 <enumeration value="MESHING_TYPE_USE_GLOBAL_SETTINGS"/>
 </restriction>
</simpleType>
<simpleType name="surface_input_axes_rotation_specification_type">
 <restriction base="string">
 <enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_ANGULAR_ROTATION"/>
 <enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_DIRECT_TO_POINT"/>
 <enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_COORDINATE_SYSTEM"/>
 <enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_LINES"/>
 </restriction>
</simpleType>
<simpleType name="surface_input_axes_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 </restriction>
</simpleType>
<simpleType name="surface_result_axes_rotation_specification_type">
 <restriction base="string">

```

```

 <enumeration value="RESULT_AXES_ROTATION_SPECIFICATION_TYPE_IDENTICAL_TO_INPUT_AXES"/>
 </restriction>
</simpleType>
<simpleType name="surface_grid_type">
 <restriction base="string">
 <enumeration value="GRID_TYPE_CARTESIAN"/>
 <enumeration value="GRID_TYPE_POLAR"/>
 </restriction>
</simpleType>
<simpleType name="surface_deflection_check_surface_type">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_SURFACE_TYPE_CANTILEVER"/>
 <enumeration value="DEFLECTION_CHECK_SURFACE_TYPE_DOUBLE_SUPPORTED"/>
 </restriction>
</simpleType>
<simpleType name="surface_deflection_check_displacement_reference">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_USER_DEFINED_REFERENCE_PLANE"/>
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_PARALLEL_SURFACE"/>
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_UNDEFORMED_SYSTEM"/>
 </restriction>
</simpleType>
<simpleType name="surface_deflection_check_reference_length_z_definition_type">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_REFERENCE_LENGTH_DEFINITION_TYPE_BY_MAXIMUM_BOUNDARY_LINE"/>
 <enumeration value="DEFLECTION_CHECK_REFERENCE_LENGTH_DEFINITION_TYPE_BY_MINIMUM_BOUNDARY_LINE"/>
 <enumeration value="DEFLECTION_CHECK_REFERENCE_LENGTH_DEFINITION_TYPE_MANUALLY"/>
 </restriction>
</simpleType>
<complexType name="get_surface">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_surface" type="xsd:get_surface"/>
<complexType name="get_surfaceResponse">
 <sequence>
 <element name="value" type="xsd:surface"/>
 </sequence>
</complexType>
<element name="get_surfaceResponse" type="xsd:get_surfaceResponse"/>
<complexType name="set_surface">
 <sequence>
 <element name="value" type="xsd:surface"/>
 </sequence>
</complexType>
<element name="set_surface" type="xsd:set_surface"/>
<complexType name="set_surfaceResponse">
 <sequence/>
</complexType>
<element name="set_surfaceResponse" type="xsd:set_surfaceResponse"/>
<complexType name="surface">
 <sequence>
 <element name="no" type="int"/>
 <element name="geometry" minOccurs="0" type="xsd:surface_geometry"/>
 <element name="type" minOccurs="0" type="xsd:surface_type"/>
 <element name="boundary_lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="thickness" minOccurs="0" type="int"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="analytical_area" minOccurs="0" type="double"/>
 <element name="analytical_volume" minOccurs="0" type="double"/>
 <element name="analytical_mass" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="analytical_center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="area" minOccurs="0" type="double"/>
 <element name="volume" minOccurs="0" type="double"/>
 <element name="mass" minOccurs="0" type="double"/>
 <element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="position" minOccurs="0" type="string"/>
 <element name="position_short" minOccurs="0" type="string"/>
 <element name="grid_enabled" minOccurs="0" type="boolean"/>
 </sequence>

```

```
<element name="is_deactivated_for_calculation" minOccurs="0" type="boolean"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="design_properties_via_surface" minOccurs="0" type="boolean"/>
<element name="load_transfer_direction" minOccurs="0" type="xsd:surface_load_transfer_direction"/>
<element name="is_surface_weight_enabled" minOccurs="0" type="boolean"/>
<element name="surface_weight" minOccurs="0" type="double"/>
<element name="consider_member_eccentricity" minOccurs="0" type="boolean"/>
<element name="consider_section_distribution" minOccurs="0" type="boolean"/>
<element name="excluded_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="excluded_parallel_to_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="excluded_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="excluded_parallel_to_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="loaded_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="loaded_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="nurbs_control_point_count_in_direction_u" minOccurs="0" type="int"/>
<element name="nurbs_control_point_count_in_direction_v" minOccurs="0" type="int"/>
<element name="nurbs_order_in_direction_u" minOccurs="0" type="int"/>
<element name="nurbs_order_in_direction_v" minOccurs="0" type="int"/>
<element name="nurbs_control_points" minOccurs="0" type="xsd:array_of_surface_nurbs_control_points_rows"/>
<element name="quadrangle_corner_nodes" minOccurs="0" type="xsd:array_of_int"/>
<element name="quadrangle_corner_node_1" minOccurs="0" type="int"/>
<element name="quadrangle_corner_node_2" minOccurs="0" type="int"/>
<element name="quadrangle_corner_node_3" minOccurs="0" type="int"/>
<element name="quadrangle_corner_node_4" minOccurs="0" type="int"/>
<element name="pipe_radius" minOccurs="0" type="double"/>
<element name="pipe_center_line" minOccurs="0" type="int"/>
<element name="pipe_generated_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="has_line_hinges" minOccurs="0" type="boolean"/>
<element name="support" minOccurs="0" type="int"/>
<element name="eccentricity" minOccurs="0" type="int"/>
<element name="mesh_refinement" minOccurs="0" type="int"/>
<element name="meshing_type" minOccurs="0" type="xsd:surface_meshing_type"/>
<element name="input_axes_rotation_specification_type" minOccurs="0" type="xsd:surface_input_axes_rotation_specification_type"/>
<element name="input_axes_angular_rotation" minOccurs="0" type="double"/>
<element name="input_axes_axis" minOccurs="0" type="xsd:surface_input_axes_axis"/>
<element name="input_axes_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="input_axes_point_1" minOccurs="0" type="xsd:vector_3d"/>
<element name="input_axes_point_1_x" minOccurs="0" type="double"/>
<element name="input_axes_point_1_y" minOccurs="0" type="double"/>
<element name="input_axes_point_1_z" minOccurs="0" type="double"/>
<element name="input_axes_point_2" minOccurs="0" type="xsd:vector_3d"/>
<element name="input_axes_point_2_x" minOccurs="0" type="double"/>
<element name="input_axes_point_2_y" minOccurs="0" type="double"/>
<element name="input_axes_point_2_z" minOccurs="0" type="double"/>
<element name="input_axes_coordinate_system" minOccurs="0" type="int"/>
<element name="result_axes_rotation_specification_type" minOccurs="0" type="xsd:surface_result_axes_rotation_specification_type"/>
<element name="reversed_normal" minOccurs="0" type="boolean"/>
<element name="grid_type" minOccurs="0" type="xsd:surface_grid_type"/>
<element name="grid_origin" minOccurs="0" type="xsd:vector_3d"/>
<element name="grid_origin_x" minOccurs="0" type="double"/>
<element name="grid_origin_y" minOccurs="0" type="double"/>
<element name="grid_origin_z" minOccurs="0" type="double"/>
<element name="grid_point_count_negative_x" minOccurs="0" type="int"/>
<element name="grid_point_count_positive_x" minOccurs="0" type="int"/>
<element name="grid_point_count_negative_y" minOccurs="0" type="int"/>
<element name="grid_point_count_positive_y" minOccurs="0" type="int"/>
<element name="grid_numbering_increment" minOccurs="0" type="int"/>
<element name="grid_point_count_r" minOccurs="0" type="int"/>
<element name="grid_distance_x" minOccurs="0" type="double"/>
<element name="grid_distance_y" minOccurs="0" type="double"/>
<element name="grid_distance_r" minOccurs="0" type="double"/>
<element name="grid_rotation_alpha" minOccurs="0" type="double"/>
<element name="grid_rotation_beta" minOccurs="0" type="double"/>
<element name="grid_angle_gamma" minOccurs="0" type="double"/>
<element name="grid_adapt_automatically" minOccurs="0" type="boolean"/>
<element name="auto_detection_of_integrated_objects" minOccurs="0" type="boolean"/>
<element name="integrated_nodes" minOccurs="0" type="xsd:array_of_int"/>
<element name="integrated_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="integrated_openings" minOccurs="0" type="xsd:array_of_int"/>
<element name="has_integrated_objects" minOccurs="0" type="boolean"/>
<element name="has_input_axes_rotation" minOccurs="0" type="boolean"/>
<element name="has_result_axes_rotation" minOccurs="0" type="boolean"/>
<element name="timber_service_class" minOccurs="0" type="int"/>
```

```

<element name="timber_moisture_class" minOccurs="0" type="int"/>
<element name="timber_service_conditions" minOccurs="0" type="int"/>
<element name="surface_reinforcements" minOccurs="0" type="xsd:array_of_int"/>
<element name="is_user_defined_concrete_cover_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover_top" minOccurs="0" type="double"/>
<element name="concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="user_defined_concrete_cover_top" minOccurs="0" type="double"/>
<element name="user_defined_concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="concrete_durability_top" minOccurs="0" type="int"/>
<element name="concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="reinforcement_direction_top" minOccurs="0" type="int"/>
<element name="reinforcement_direction_bottom" minOccurs="0" type="int"/>
<element name="deflection_check_surface_type" minOccurs="0" type="xsd:surface_deflection_check_surface_type"/>
<element name="deflection_check_displacement_reference" minOccurs="0"
type="xsd:surface_deflection_check_displacement_reference"/>
<element name="deflection_check_reference_length_z" minOccurs="0" type="double"/>
<element name="deflection_check_reference_length_z_definition_type" minOccurs="0"
type="xsd:surface_deflection_check_reference_length_z_definition_type"/>
<element name="deflection_check_reference_plane_point_1" minOccurs="0" type="xsd:vector_3d"/>
<element name="deflection_check_reference_plane_point_1_x" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_1_y" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_1_z" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_2" minOccurs="0" type="xsd:vector_3d"/>
<element name="deflection_check_reference_plane_point_2_x" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_2_y" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_2_z" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_3" minOccurs="0" type="xsd:vector_3d"/>
<element name="deflection_check_reference_plane_point_3_x" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_3_y" minOccurs="0" type="double"/>
<element name="deflection_check_reference_plane_point_3_z" minOccurs="0" type="double"/>
<element name="surface_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="rotated_boundary_line" minOccurs="0" type="int"/>
<element name="rotated_angle_of_rotation" minOccurs="0" type="double"/>
<element name="rotated_point_p" minOccurs="0" type="xsd:vector_3d"/>
<element name="rotated_point_p_x" minOccurs="0" type="double"/>
<element name="rotated_point_p_y" minOccurs="0" type="double"/>
<element name="rotated_point_p_z" minOccurs="0" type="double"/>
<element name="rotated_point_r" minOccurs="0" type="xsd:vector_3d"/>
<element name="rotated_point_r_x" minOccurs="0" type="double"/>
<element name="rotated_point_r_y" minOccurs="0" type="double"/>
<element name="rotated_point_r_z" minOccurs="0" type="double"/>
<element name="rotated_generated_lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<complexType name="get_opening">
<sequence>
<element name="no" type="int"/>
</sequence>
</complexType>
<element name="get_opening" type="xsd:get_opening"/>
<complexType name="get_openingResponse">
<sequence>
<element name="value" type="xsd:opening"/>
</sequence>
</complexType>
<element name="get_openingResponse" type="xsd:get_openingResponse"/>
<complexType name="set_opening">
<sequence>
<element name="value" type="xsd:opening"/>
</sequence>
</complexType>
<element name="set_opening" type="xsd:set_opening"/>
<complexType name="set_openingResponse">
<sequence/>
</complexType>
<element name="set_openingResponse" type="xsd:set_openingResponse"/>
<complexType name="opening">

```

```

<sequence>
 <element name="no" type="int"/>
 <element name="area" minOccurs="0" type="double"/>
 <element name="boundary_lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="center_of_opening" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_of_opening_x" minOccurs="0" type="double"/>
 <element name="center_of_opening_y" minOccurs="0" type="double"/>
 <element name="center_of_opening_z" minOccurs="0" type="double"/>
 <element name="position_full_description" minOccurs="0" type="string"/>
 <element name="position_short_description" minOccurs="0" type="string"/>
 <element name="surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="solid_type">
 <restriction base="string">
 <enumeration value="TYPE_CONTACT"/>
 <enumeration value="TYPE_GAS"/>
 <enumeration value="TYPE_HOLE"/>
 <enumeration value="TYPE_INTERSECTION"/>
 <enumeration value="TYPE_SOIL"/>
 <enumeration value="TYPE_STANDARD"/>
 </restriction>
</simpleType>
<simpleType name="solid_specific_direction_type">
 <restriction base="string">
 <enumeration value="DIRECTION_TYPE_DIRECTED_TO_NODE"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_CS_OF_LINE"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_CS_OF_MEMBER"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_TWO_NODES"/>
 <enumeration value="DIRECTION_TYPE_ROTATED_VIA_3_ANGLES"/>
 </restriction>
</simpleType>
<simpleType name="solid_axes_sequence">
 <restriction base="string">
 <enumeration value="SEQUENCE_XYZ"/>
 <enumeration value="SEQUENCE_XZY"/>
 <enumeration value="SEQUENCE_YXZ"/>
 <enumeration value="SEQUENCE_YZX"/>
 <enumeration value="SEQUENCE_ZXY"/>
 <enumeration value="SEQUENCE_ZYX"/>
 </restriction>
</simpleType>
<simpleType name="solid_directed_to_node_first_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="solid_directed_to_node_second_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="solid_parallel_to_two_nodes_first_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="solid_parallel_to_two_nodes_second_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<complexType name="get_solid">

```

```

 <sequence>
 <element name="no" type="int"/>
 </sequence>
 </complexType>
 <element name="get_solid" type="xsd:get_solid"/>
 <complexType name="get_solidResponse">
 <sequence>
 <element name="value" type="xsd:solid"/>
 </sequence>
 </complexType>
 <element name="get_solidResponse" type="xsd:get_solidResponse"/>
 <complexType name="set_solid">
 <sequence>
 <element name="value" type="xsd:solid"/>
 </sequence>
 </complexType>
 <element name="set_solid" type="xsd:set_solid"/>
 <complexType name="set_solidResponse">
 <sequence/>
 </complexType>
 <element name="set_solidResponse" type="xsd:set_solidResponse"/>
 <complexType name="solid">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:solid_type"/>
 <element name="analytical_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="analytical_center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="analytical_center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="analytical_mass" minOccurs="0" type="double"/>
 <element name="analytical_surface_area" minOccurs="0" type="double"/>
 <element name="analytical_volume" minOccurs="0" type="double"/>
 <element name="boundary_surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="gas" minOccurs="0" type="int"/>
 <element name="is_deactivated_for_calculation" minOccurs="0" type="boolean"/>
 <element name="mass" minOccurs="0" type="double"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="mesh_refinement" minOccurs="0" type="int"/>
 <element name="solid_contact" minOccurs="0" type="int"/>
 <element name="solid_contact_first_surface" minOccurs="0" type="int"/>
 <element name="solid_contact_second_surface" minOccurs="0" type="int"/>
 <element name="stress_analysis_configuration" minOccurs="0" type="int"/>
 <element name="surface_area" minOccurs="0" type="double"/>
 <element name="volume" minOccurs="0" type="double"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="grid_enabled" minOccurs="0" type="boolean"/>
 <element name="grid_inner_points" minOccurs="0" type="boolean"/>
 <element name="grid_boundary_points" minOccurs="0" type="boolean"/>
 <element name="specific_direction_type" minOccurs="0" type="xsd:solid_specific_direction_type"/>
 <element name="grid_distance_x" minOccurs="0" type="double"/>
 <element name="grid_distance_y" minOccurs="0" type="double"/>
 <element name="grid_distance_z" minOccurs="0" type="double"/>
 <element name="use_center_of_gravity_as_grid_origin" minOccurs="0" type="boolean"/>
 <element name="grid_origin" minOccurs="0" type="xsd:vector_3d"/>
 <element name="grid_origin_x" minOccurs="0" type="double"/>
 <element name="grid_origin_y" minOccurs="0" type="double"/>
 <element name="grid_origin_z" minOccurs="0" type="double"/>
 <element name="grid_adapt_automatically" minOccurs="0" type="boolean"/>
 <element name="grid_point_count_negative_u" minOccurs="0" type="int"/>
 <element name="grid_point_count_positive_u" minOccurs="0" type="int"/>
 <element name="grid_point_count_negative_v" minOccurs="0" type="int"/>
 <element name="grid_point_count_positive_v" minOccurs="0" type="int"/>
 <element name="grid_point_count_negative_w" minOccurs="0" type="int"/>
 <element name="grid_point_count_positive_w" minOccurs="0" type="int"/>
 <element name="specific_direction_enabled" minOccurs="0" type="boolean"/>
 <element name="coordinate_system" minOccurs="0" type="int"/>
 <element name="axes_sequence" minOccurs="0" type="xsd:solid_axes_sequence"/>
 <element name="rotated_about_angle_x" minOccurs="0" type="double"/>
 <element name="rotated_about_angle_y" minOccurs="0" type="double"/>
 </sequence>
 </complexType>

```

```

<element name="rotated_about_angle_z" minOccurs="0" type="double"/>
<element name="rotated_about_angle_1" minOccurs="0" type="double"/>
<element name="rotated_about_angle_2" minOccurs="0" type="double"/>
<element name="rotated_about_angle_3" minOccurs="0" type="double"/>
<element name="directed_to_node_direction_node" minOccurs="0" type="int"/>
<element name="directed_to_node_plane_node" minOccurs="0" type="int"/>
<element name="directed_to_node_first_axis" minOccurs="0" type="xsd:solid_directed_to_node_first_axis"/>
<element name="directed_to_node_second_axis" minOccurs="0" type="xsd:solid_directed_to_node_second_axis"/>
<element name="parallel_to_two_nodes_first_node" minOccurs="0" type="int"/>
<element name="parallel_to_two_nodes_second_node" minOccurs="0" type="int"/>
<element name="parallel_to_two_nodes_plane_node" minOccurs="0" type="int"/>
<element name="parallel_to_two_nodes_first_axis" minOccurs="0" type="xsd:solid_parallel_to_two_nodes_first_axis"/>
<element name="parallel_to_two_nodes_second_axis" minOccurs="0" type="xsd:solid_parallel_to_two_nodes_second_axis"/>
<element name="parallel_to_line" minOccurs="0" type="int"/>
<element name="parallel_to_member" minOccurs="0" type="int"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="line_set_set_type">
 <restriction base="string">
 <enumeration value="SET_TYPE_CONTINUOUS"/>
 <enumeration value="SET_TYPE_GROUP"/>
 </restriction>
</simpleType>
<complexType name="get_line_set">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_line_set" type="xsd:get_line_set"/>
<complexType name="get_line_setResponse">
 <sequence>
 <element name="value" type="xsd:line_set"/>
 </sequence>
</complexType>
<element name="get_line_setResponse" type="xsd:get_line_setResponse"/>
<complexType name="set_line_set">
 <sequence>
 <element name="value" type="xsd:line_set"/>
 </sequence>
</complexType>
<element name="set_line_set" type="xsd:set_line_set"/>
<complexType name="set_line_setResponse">
 <sequence/>
</complexType>
<element name="set_line_setResponse" type="xsd:set_line_setResponse"/>
<complexType name="line_set">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="set_type" minOccurs="0" type="xsd:line_set_set_type"/>
 <element name="lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="position" minOccurs="0" type="string"/>
 <element name="position_short" minOccurs="0" type="string"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="member_set_set_type">
 <restriction base="string">
 <enumeration value="SET_TYPE_CONTINUOUS"/>
 <enumeration value="SET_TYPE_GROUP"/>
 </restriction>
</simpleType>
<complexType name="member_set_concrete_shear_reinforcement_spans">
 <sequence>

```

```

<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="stirrup_type" minOccurs="0" type="xsd:stirrup_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="stirrup_count" minOccurs="0" type="int"/>
<element name="stirrup_diameter" minOccurs="0" type="double"/>
<element name="stirrup_bar_size_type" minOccurs="0" type="xsd:stirrup_bar_size_type"/>
<element name="stirrup_distances" minOccurs="0" type="double"/>
<element name="reinforcement_area" minOccurs="0" type="double"/>
<element name="span_position_reference_type" minOccurs="0" type="xsd:span_position_reference_type"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_position_definition_format_type" minOccurs="0" type="xsd:span_position_definition_format_type"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="stirrup_layout_rule_type" minOccurs="0" type="xsd:stirrup_layout_rule_type"/>
<element name="stirrup_start_offset" minOccurs="0" type="double"/>
<element name="stirrup_end_offset" minOccurs="0" type="double"/>
<element name="one_stirrup_length" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_stirrup_weight" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_concrete_shear_reinforcement_spans">
<sequence>
<element name="member_set_concrete_shear_reinforcement_spans" type="xsd:member_set_concrete_shear_reinforcement_spans"
minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_concrete_shear_reinforcement_spans"
type="xsd:array_of_member_set_concrete_shear_reinforcement_spans"/>
<complexType name="member_set_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="rebar_type" minOccurs="0" type="xsd:rebar_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="reinforcement_placed_in_bending_corner_enabled" minOccurs="0" type="boolean"/>
<element name="bar_count_symmetrical" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_at_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_top_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_bottom_side" minOccurs="0" type="int"/>
<element name="bar_count_uniformly_surrounding" minOccurs="0" type="int"/>
<element name="bar_count_line" minOccurs="0" type="int"/>
<element name="bar_count_corner" minOccurs="0" type="int"/>
<element name="bar_diameter_symmetrical" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="bar_diameter_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="bar_diameter_line" minOccurs="0" type="double"/>
<element name="bar_diameter_single" minOccurs="0" type="double"/>
<element name="bar_diameter_corner" minOccurs="0" type="double"/>
<element name="bar_size_designation_symmetrical_type" minOccurs="0" type="xsd:bar_size_designation_symmetrical_type"/>
<element name="bar_size_designation_unsymmetrical_at_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_at_side_type"/>
<element name="bar_size_designation_unsymmetrical_top_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_top_side_type"/>
<element name="bar_size_designation_unsymmetrical_bottom_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_bottom_side_type"/>
<element name="bar_size_designation_uniformly_surrounding_type" minOccurs="0"
type="xsd:bar_size_designation_uniformly_surrounding_type"/>
<element name="bar_size_designation_line_type" minOccurs="0" type="xsd:bar_size_designation_line_type"/>
<element name="bar_size_designation_single_type" minOccurs="0" type="xsd:bar_size_designation_single_type"/>
<element name="bar_size_designation_corner_type" minOccurs="0" type="xsd:bar_size_designation_corner_type"/>
<element name="corner_reinforcement_enabled" minOccurs="0" type="boolean"/>
<element name="reinforcement_area_symmetrical" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_top_side" minOccurs="0" type="double"/>

```



```

<element name="reinforcement_area_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="reinforcement_area_line" minOccurs="0" type="double"/>
<element name="reinforcement_area_single" minOccurs="0" type="double"/>
<element name="reinforcement_area_corner" minOccurs="0" type="double"/>
<element name="reinforcement_area_total" minOccurs="0" type="double"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="additional_offset_type" minOccurs="0" type="xsd:additional_offset_type"/>
<element name="additional_offset_type_single_line_type" minOccurs="0" type="xsd:additional_offset_type_single_line_type"/>
<element name="additional_offset_top_side" minOccurs="0" type="double"/>
<element name="additional_offset_bottom_side" minOccurs="0" type="double"/>
<element name="additional_offset_left_side" minOccurs="0" type="double"/>
<element name="additional_offset_right_side" minOccurs="0" type="double"/>
<element name="additional_offset_reference_type" minOccurs="0" type="xsd:additional_offset_reference_type"/>
<element name="additional_offset_reference_type_at_start_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_start_type"/>
<element name="additional_offset_reference_type_at_end_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_end_type"/>
<element name="additional_horizontal_offset" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_end" minOccurs="0" type="double"/>
<element name="additional_vertical_offset" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_end" minOccurs="0" type="double"/>
<element name="anchorage_start_anchor_type" minOccurs="0" type="xsd:anchorage_start_anchor_type"/>
<element name="anchorage_start_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_start_bending_diameter" minOccurs="0" type="double"/>
<element name="anchorage_end_anchor_type" minOccurs="0" type="xsd:anchorage_end_anchor_type"/>
<element name="anchorage_end_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_end_bending_diameter" minOccurs="0" type="double"/>
<element name="one_rebar_length" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_corner_length" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_rebar_weight" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_corner_weight" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="member_set_concrete_longitudinal_reinforcement_items"
type="xsd:member_set_concrete_longitudinal_reinforcement_items" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_concrete_longitudinal_reinforcement_items"
type="xsd:array_of_member_set_concrete_longitudinal_reinforcement_items"/>
<simpleType name="member_set_deflection_check_direction">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z_AND_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_RESULTING_AXIS"/>

```

```

 </restriction>
</simpleType>
<simpleType name="member_set_deflection_check_displacement_reference">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_SEGMENT_ENDS"/>
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_UNDEFORMED_SYSTEM"/>
 </restriction>
</simpleType>
<complexType name="member_set_design_supports_on_internal_nodes">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="node" minOccurs="0" type="int"/>
 <element name="design_support" minOccurs="0" type="int"/>
 </sequence>
</complexType>
<complexType name="array_of_member_set_design_supports_on_internal_nodes">
 <sequence>
 <element name="member_set_design_supports_on_internal_nodes" type="xsd:member_set_design_supports_on_internal_nodes"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_set_design_supports_on_internal_nodes" type="xsd:array_of_member_set_design_supports_on_internal_nodes"/>
<complexType name="member_set_deflection_segments_y_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_set_deflection_segments_y_axis">
 <sequence>
 <element name="member_set_deflection_segments_y_axis" type="xsd:member_set_deflection_segments_y_axis"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_set_deflection_segments_y_axis" type="xsd:array_of_member_set_deflection_segments_y_axis"/>
<complexType name="member_set_deflection_segments_z_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_set_deflection_segments_z_axis">
 <sequence>
 <element name="member_set_deflection_segments_z_axis" type="xsd:member_set_deflection_segments_z_axis"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_set_deflection_segments_z_axis" type="xsd:array_of_member_set_deflection_segments_z_axis"/>
<complexType name="get_member_set">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_member_set" type="xsd:get_member_set"/>
<complexType name="get_member_setResponse">
 <sequence>
 <element name="value" type="xsd:member_set"/>
 </sequence>
</complexType>
<element name="get_member_setResponse" type="xsd:get_member_setResponse"/>
<complexType name="set_member_set">
 <sequence>
 <element name="value" type="xsd:member_set"/>
 </sequence>
</complexType>
<element name="set_member_set" type="xsd:set_member_set"/>
<complexType name="set_member_setResponse">
 <sequence/>
</complexType>
<element name="set_member_setResponse" type="xsd:set_member_setResponse"/>
<complexType name="member_set">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 </sequence>

```

```
<element name="name" minOccurs="0" type="string"/>
<element name="set_type" minOccurs="0" type="xsd:member_set_set_type"/>
<element name="length" minOccurs="0" type="double"/>
<element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="center_of_gravity_x" minOccurs="0" type="double"/>
<element name="center_of_gravity_y" minOccurs="0" type="double"/>
<element name="center_of_gravity_z" minOccurs="0" type="double"/>
<element name="position" minOccurs="0" type="string"/>
<element name="position_short" minOccurs="0" type="string"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="volume" minOccurs="0" type="double"/>
<element name="mass" minOccurs="0" type="double"/>
<element name="member_set_representative" minOccurs="0" type="int"/>
<element name="discontinuous_torsional_warping" minOccurs="0" type="boolean"/>
<element name="design_properties_activated" minOccurs="0" type="boolean"/>
<element name="steel_effective_lengths" minOccurs="0" type="int"/>
<element name="steel_boundary_conditions" minOccurs="0" type="int"/>
<element name="steel_member_local_section_reductions" minOccurs="0" type="int"/>
<element name="steel_member_shear_panels" minOccurs="0" type="int"/>
<element name="steel_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="member_steel_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_steel_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="aluminum_effective_lengths" minOccurs="0" type="int"/>
<element name="aluminum_boundary_conditions" minOccurs="0" type="int"/>
<element name="aluminum_member_local_section_reductions" minOccurs="0" type="int"/>
<element name="aluminum_member_shear_panels" minOccurs="0" type="int"/>
<element name="aluminum_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="aluminum_member_transverse_weld" minOccurs="0" type="int"/>
<element name="member_aluminum_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_aluminum_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_set_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_shear_reinforcement_spans" minOccurs="0" type="xsd:array_of_member_set_concrete_shear_reinforcement_spans"/>
<element name="concrete_longitudinal_reinforcement_items" minOccurs="0" type="xsd:array_of_member_set_concrete_longitudinal_reinforcement_items"/>
<element name="concrete_cover_user_defined_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover_different_at_section_sides_enabled" minOccurs="0" type="boolean"/>
<element name="concrete_cover" minOccurs="0" type="double"/>
<element name="concrete_cover_top" minOccurs="0" type="double"/>
<element name="concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="concrete_cover_left" minOccurs="0" type="double"/>
<element name="concrete_cover_right" minOccurs="0" type="double"/>
<element name="concrete_cover_min" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_top" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_bottom" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_left" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_cover_min_right" minOccurs="0" type="xsd:array_of_int"/>
<element name="concrete_durability" minOccurs="0" type="int"/>
<element name="concrete_durability_top" minOccurs="0" type="int"/>
<element name="concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="concrete_durability_left" minOccurs="0" type="int"/>
<element name="concrete_durability_right" minOccurs="0" type="int"/>
<element name="concrete_effective_lengths" minOccurs="0" type="int"/>
<element name="member_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="timber_effective_lengths" minOccurs="0" type="int"/>
<element name="service_class_timber_design" minOccurs="0" type="int"/>
<element name="moisture_class_timber_design" minOccurs="0" type="int"/>
<element name="service_conditions_timber_design" minOccurs="0" type="int"/>
<element name="timber_member_shear_panels" minOccurs="0" type="int"/>
<element name="timber_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="timber_local_section_reductions" minOccurs="0" type="int"/>
<element name="member_timber_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_timber_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_transverse_stiffener" minOccurs="0" type="int"/>
<element name="stress_analysis_configuration" minOccurs="0" type="int"/>
```

```

<element name="deflection_check_direction" minOccurs="0" type="xsd:member_set_deflection_check_direction"/>
<element name="deflection_check_displacement_reference" minOccurs="0"
type="xsd:member_set_deflection_check_displacement_reference"/>
<element name="design_support_on_member_set_start" minOccurs="0" type="int"/>
<element name="design_support_on_member_set_end" minOccurs="0" type="int"/>
<element name="design_supports_on_internal_nodes" minOccurs="0"
type="xsd:array_of_member_set_design_supports_on_internal_nodes"/>
<element name="members" minOccurs="0" type="xsd:array_of_int"/>
<element name="deflection_segments_defined_length_y_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_defined_length_z_axis_enabled" minOccurs="0" type="boolean"/>
<element name="deflection_segments_y_axis" minOccurs="0" type="xsd:array_of_member_set_deflection_segments_y_axis"/>
<element name="deflection_segments_z_axis" minOccurs="0" type="xsd:array_of_member_set_deflection_segments_z_axis"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="member_set_representative_member_model_type">
<restriction base="string">
<enumeration value="TYPE_BEAM"/>
<enumeration value="TYPE_BUCKLING"/>
<enumeration value="TYPE_CABLE"/>
<enumeration value="TYPE_COMPRESSION"/>
<enumeration value="TYPE_COUPLING_HINGE_HINGE"/>
<enumeration value="TYPE_COUPLING_HINGE_RIGID"/>
<enumeration value="TYPE_COUPLING_RIGID_HINGE"/>
<enumeration value="TYPE_COUPLING_RIGID_RIGID"/>
<enumeration value="TYPE_DEFINABLE_STIFFNESS"/>
<enumeration value="TYPE_RESULT_BEAM"/>
<enumeration value="TYPE_RIB"/>
<enumeration value="TYPE_RIGID"/>
<enumeration value="TYPE_TENSION"/>
<enumeration value="TYPE_TRUSS"/>
<enumeration value="TYPE_TRUSS_ONLY_N"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_section_distribution_type">
<restriction base="string">
<enumeration value=""/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_LINEAR"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_OFFSET_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_SADDLE"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_BOTH_SIDES"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_END_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_TAPERED_AT_START_OF_MEMBER"/>
<enumeration value="SECTION_DISTRIBUTION_TYPE_UNIFORM"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_reference_type">
<restriction base="string">
<enumeration value="REFERENCE_TYPE_L"/>
<enumeration value="REFERENCE_TYPE_XY"/>
<enumeration value="REFERENCE_TYPE_XZ"/>
<enumeration value="REFERENCE_TYPE_YZ"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_member_type_rib_alignment">
<restriction base="string">
<enumeration value="ALIGNMENT_CENTRIC"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_NEGATIVE"/>
<enumeration value="ALIGNMENT_ON_Z_SIDE_POSITIVE"/>
<enumeration value="ALIGNMENT_USER_DEFINED_VIA_MEMBER_ECCENTRICITY"/>
</restriction>
</simpleType>
<complexType name="member_set_representative_member_model_flange_dimensions">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="end_ordinate" minOccurs="0" type="double"/>
<element name="length" minOccurs="0" type="double"/>
<element name="reference_length_definition_type" minOccurs="0" type="xsd:reference_length_definition_type"/>
<element name="reference_length" minOccurs="0" type="double"/>
<element name="reference_length_width_type" minOccurs="0" type="xsd:reference_length_width_type"/>
<element name="width_minus_y_integrative" minOccurs="0" type="double"/>
<element name="width_minus_y_effictive" minOccurs="0" type="double"/>
<element name="width_minus_y_maximal" minOccurs="0" type="double"/>

```

```

<element name="width_plus_y_integrative" minOccurs="0" type="double"/>
<element name="width_plus_y_effective" minOccurs="0" type="double"/>
<element name="width_plus_y_maximal" minOccurs="0" type="double"/>
<element name="distribution_linearity_type" minOccurs="0" type="xsd:distribution_linearity_type"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_model_flange_dimensions">
<sequence>
<element name="member_set_representative_member_model_flange_dimensions"
type="xsd:member_set_representative_member_model_flange_dimensions" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_representative_member_model_flange_dimensions"
type="xsd:array_of_member_set_representative_member_model_flange_dimensions"/>
<simpleType name="member_set_representative_member_model_result_beam_integrate_stresses_and_forces">
<restriction base="string">
<enumeration value="INTEGRATE_FROM_LISTED_OBJECT"/>
<enumeration value="INTEGRATE_WITHIN_CUBOID_GENERAL"/>
<enumeration value="INTEGRATE_WITHIN_CUBOID_QUADRATIC"/>
<enumeration value="INTEGRATE_WITHIN_CYLINDER"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_section_alignment">
<restriction base="string">
<enumeration value="SECTION_ALIGNMENT_BOTTOM"/>
<enumeration value="SECTION_ALIGNMENT_CENTRIC"/>
<enumeration value="SECTION_ALIGNMENT_TOP"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_curved_member_cantilevers_type">
<restriction base="string">
<enumeration value="CANTILEVERS_TYPE_HORIZONTAL"/>
<enumeration value="CANTILEVERS_TYPE_OFFSET"/>
<enumeration value="CANTILEVERS_TYPE_PARALLEL"/>
<enumeration value="CANTILEVERS_TYPE_TAPER"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_rotation_specification_type">
<restriction base="string">
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_ANGLE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_HELP_NODE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_INSIDE_NODE"/>
<enumeration value="COORDINATE_SYSTEM_ROTATION_VIA_SURFACE"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_rotation_plane_type">
<restriction base="string">
<enumeration value="ROTATION_PLANE_XY"/>
<enumeration value="ROTATION_PLANE_XZ"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_rotation_surface_plane_type">
<restriction base="string">
<enumeration value="ROTATION_PLANE_XY"/>
<enumeration value="ROTATION_PLANE_XZ"/>
</restriction>
</simpleType>
<complexType name="member_set_representative_member_model_concrete_shear_reinforcement_spans">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="stirrup_type" minOccurs="0" type="xsd:stirrup_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="stirrup_count" minOccurs="0" type="int"/>
<element name="stirrup_diameter" minOccurs="0" type="double"/>
<element name="stirrup_bar_size_type" minOccurs="0" type="xsd:stirrup_bar_size_type"/>
<element name="stirrup_distances" minOccurs="0" type="double"/>
<element name="reinforcement_area" minOccurs="0" type="double"/>
<element name="span_position_reference_type" minOccurs="0" type="xsd:span_position_reference_type"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_position_definition_format_type" minOccurs="0" type="xsd:span_position_definition_format_type"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>

```

```

<element name="span_length" minOccurs="0" type="double"/>
<element name="stirrup_layout_rule_type" minOccurs="0" type="xsd:stirrup_layout_rule_type"/>
<element name="stirrup_start_offset" minOccurs="0" type="double"/>
<element name="stirrup_end_offset" minOccurs="0" type="double"/>
<element name="one_stirrup_length" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_stirrup_weight" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_model_concrete_shear_reinforcement_spans">
<sequence>
<element name="member_set_representative_member_model_concrete_shear_reinforcement_spans"
type="xsd:member_set_representative_member_model_concrete_shear_reinforcement_spans" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_representative_member_model_concrete_shear_reinforcement_spans"
type="xsd:array_of_member_set_representative_member_model_concrete_shear_reinforcement_spans"/>
<complexType name="member_set_representative_member_model_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="rebar_type" minOccurs="0" type="xsd:rebar_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="reinforcement_placed_in_bending_corner_enabled" minOccurs="0" type="boolean"/>
<element name="bar_count_symmetrical" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_at_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_top_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_bottom_side" minOccurs="0" type="int"/>
<element name="bar_count_uniformly_surrounding" minOccurs="0" type="int"/>
<element name="bar_count_line" minOccurs="0" type="int"/>
<element name="bar_count_corner" minOccurs="0" type="int"/>
<element name="bar_diameter_symmetrical" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="bar_diameter_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="bar_diameter_line" minOccurs="0" type="double"/>
<element name="bar_diameter_single" minOccurs="0" type="double"/>
<element name="bar_diameter_corner" minOccurs="0" type="double"/>
<element name="bar_size_designation_symmetrical_type" minOccurs="0" type="xsd:bar_size_designation_symmetrical_type"/>
<element name="bar_size_designation_unsymmetrical_at_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_at_side_type"/>
<element name="bar_size_designation_unsymmetrical_top_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_top_side_type"/>
<element name="bar_size_designation_unsymmetrical_bottom_side_type" minOccurs="0"
type="xsd:bar_size_designation_unsymmetrical_bottom_side_type"/>
<element name="bar_size_designation_uniformly_surrounding_type" minOccurs="0"
type="xsd:bar_size_designation_uniformly_surrounding_type"/>
<element name="bar_size_designation_line_type" minOccurs="0" type="xsd:bar_size_designation_line_type"/>
<element name="bar_size_designation_single_type" minOccurs="0" type="xsd:bar_size_designation_single_type"/>
<element name="bar_size_designation_corner_type" minOccurs="0" type="xsd:bar_size_designation_corner_type"/>
<element name="corner_reinforcement_enabled" minOccurs="0" type="boolean"/>
<element name="reinforcement_area_symmetrical" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="reinforcement_area_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="reinforcement_area_line" minOccurs="0" type="double"/>
<element name="reinforcement_area_single" minOccurs="0" type="double"/>
<element name="reinforcement_area_corner" minOccurs="0" type="double"/>
<element name="reinforcement_area_total" minOccurs="0" type="double"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="additional_offset_type" minOccurs="0" type="xsd:additional_offset_type"/>
<element name="additional_offset_type_single_line_type" minOccurs="0" type="xsd:additional_offset_type_single_line_type"/>
<element name="additional_offset_top_side" minOccurs="0" type="double"/>

```

```

<element name="additional_offset_bottom_side" minOccurs="0" type="double"/>
<element name="additional_offset_left_side" minOccurs="0" type="double"/>
<element name="additional_offset_right_side" minOccurs="0" type="double"/>
<element name="additional_offset_reference_type" minOccurs="0" type="xsd:additional_offset_reference_type"/>
<element name="additional_offset_reference_type_at_start_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_start_type"/>
<element name="additional_offset_reference_type_at_end_type" minOccurs="0"
type="xsd:additional_offset_reference_type_at_end_type"/>
<element name="additional_horizontal_offset" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_horizontal_offset_at_end" minOccurs="0" type="double"/>
<element name="additional_vertical_offset" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_start" minOccurs="0" type="double"/>
<element name="additional_vertical_offset_at_end" minOccurs="0" type="double"/>
<element name="anchorage_start_anchor_type" minOccurs="0" type="xsd:anchorage_start_anchor_type"/>
<element name="anchorage_start_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_start_bending_diameter" minOccurs="0" type="double"/>
<element name="anchorage_end_anchor_type" minOccurs="0" type="xsd:anchorage_end_anchor_type"/>
<element name="anchorage_end_anchor_length" minOccurs="0" type="double"/>
<element name="anchorage_end_bending_diameter" minOccurs="0" type="double"/>
<element name="one_rebar_length" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_length" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="one_rebar_corner_length" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_rebar_weight" minOccurs="0" type="double"/>
<element name="one_rebar_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_at_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_at_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_top_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_top_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_unsymmetrical_bottom_side_weight" minOccurs="0" type="double"/>
<element name="one_rebar_unsymmetrical_bottom_side_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="one_rebar_corner_weight" minOccurs="0" type="double"/>
<element name="one_rebar_corner_minimal_and_maximal_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_model_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="member_set_representative_member_model_concrete_longitudinal_reinforcement_items"
type="xsd:member_set_representative_member_model_concrete_longitudinal_reinforcement_items" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_representative_member_model_concrete_longitudinal_reinforcement_items"
type="xsd:array_of_member_set_representative_member_model_concrete_longitudinal_reinforcement_items"/>
<simpleType name="member_set_representative_member_model_deflection_check_direction">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z_AND_Y"/>
<enumeration value="DEFLECTION_CHECK_DIRECTION_RESULTING_AXIS"/>
</restriction>
</simpleType>
<simpleType name="member_set_representative_member_model_deflection_check_displacement_reference">
<restriction base="string">
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_SEGMENT_ENDS"/>
<enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_UNDEFORMED_SYSTEM"/>
</restriction>
</simpleType>
<complexType name="member_set_representative_member_model_design_supports_on_internal_nodes">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="node" minOccurs="0" type="int"/>
<element name="design_support" minOccurs="0" type="int"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_model_design_supports_on_internal_nodes">
<sequence>

```

```

<element name="member_set_representative_member_model_design_supports_on_internal_nodes"
type="xsd:member_set_representative_member_model_design_supports_on_internal_nodes" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_representative_member_model_design_supports_on_internal_nodes"
type="xsd:array_of_member_set_representative_member_model_design_supports_on_internal_nodes"/>
<simpleType name="member_set_representative_member_set_model_set_type">
<restriction base="string">
<enumeration value="SET_TYPE_CONTINUOUS"/>
<enumeration value="SET_TYPE_GROUP"/>
</restriction>
</simpleType>
<complexType name="member_set_representative_member_set_model_concrete_shear_reinforcement_spans">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="stirrup_type" minOccurs="0" type="xsd:stirrup_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="stirrup_count" minOccurs="0" type="int"/>
<element name="stirrup_diameter" minOccurs="0" type="double"/>
<element name="stirrup_bar_size_type" minOccurs="0" type="xsd:stirrup_bar_size_type"/>
<element name="stirrup_distances" minOccurs="0" type="double"/>
<element name="reinforcement_area" minOccurs="0" type="double"/>
<element name="span_position_reference_type" minOccurs="0" type="xsd:span_position_reference_type"/>
<element name="span_position_reference_internal_node" minOccurs="0" type="int"/>
<element name="span_position_reference_x_location_relative" minOccurs="0" type="double"/>
<element name="span_position_reference_x_location_absolute" minOccurs="0" type="double"/>
<element name="span_position_definition_format_type" minOccurs="0" type="xsd:span_position_definition_format_type"/>
<element name="span_start_relative" minOccurs="0" type="double"/>
<element name="span_start_absolute" minOccurs="0" type="double"/>
<element name="span_end_relative" minOccurs="0" type="double"/>
<element name="span_end_absolute" minOccurs="0" type="double"/>
<element name="span_length" minOccurs="0" type="double"/>
<element name="stirrup_layout_rule_type" minOccurs="0" type="xsd:stirrup_layout_rule_type"/>
<element name="stirrup_start_offset" minOccurs="0" type="double"/>
<element name="stirrup_end_offset" minOccurs="0" type="double"/>
<element name="one_stirrup_length" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_length" minOccurs="0" type="string"/>
<element name="length" minOccurs="0" type="double"/>
<element name="one_stirrup_weight" minOccurs="0" type="double"/>
<element name="minimal_and_maximal_stirrup_weight" minOccurs="0" type="string"/>
<element name="weight" minOccurs="0" type="double"/>
</sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_set_model_concrete_shear_reinforcement_spans">
<sequence>
<element name="member_set_representative_member_set_model_concrete_shear_reinforcement_spans"
type="xsd:member_set_representative_member_set_model_concrete_shear_reinforcement_spans" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_member_set_representative_member_set_model_concrete_shear_reinforcement_spans"
type="xsd:array_of_member_set_representative_member_set_model_concrete_shear_reinforcement_spans"/>
<complexType name="member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="rebar_type" minOccurs="0" type="xsd:rebar_type"/>
<element name="material" minOccurs="0" type="int"/>
<element name="reinforcement_placed_in_bending_corner_enabled" minOccurs="0" type="boolean"/>
<element name="bar_count_symmetrical" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_at_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_top_side" minOccurs="0" type="int"/>
<element name="bar_count_unsymmetrical_bottom_side" minOccurs="0" type="int"/>
<element name="bar_count_uniformly_surrounding" minOccurs="0" type="int"/>
<element name="bar_count_line" minOccurs="0" type="int"/>
<element name="bar_count_corner" minOccurs="0" type="int"/>
<element name="bar_diameter_symmetrical" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_at_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_top_side" minOccurs="0" type="double"/>
<element name="bar_diameter_unsymmetrical_bottom_side" minOccurs="0" type="double"/>
<element name="bar_diameter_uniformly_surrounding" minOccurs="0" type="double"/>
<element name="bar_diameter_line" minOccurs="0" type="double"/>
<element name="bar_diameter_single" minOccurs="0" type="double"/>
<element name="bar_diameter_corner" minOccurs="0" type="double"/>

```



<element name="bar\_size\_designation\_symmetrical\_type" minOccurs="0" type="xsd:bar\_size\_designation\_symmetrical\_type"/>  
<element name="bar\_size\_designation\_unsymmetrical\_at\_side\_type" minOccurs="0" type="xsd:bar\_size\_designation\_unsymmetrical\_at\_side\_type"/>  
<element name="bar\_size\_designation\_unsymmetrical\_top\_side\_type" minOccurs="0" type="xsd:bar\_size\_designation\_unsymmetrical\_top\_side\_type"/>  
<element name="bar\_size\_designation\_unsymmetrical\_bottom\_side\_type" minOccurs="0" type="xsd:bar\_size\_designation\_unsymmetrical\_bottom\_side\_type"/>  
<element name="bar\_size\_designation\_uniformly\_surrounding\_type" minOccurs="0" type="xsd:bar\_size\_designation\_uniformly\_surrounding\_type"/>  
<element name="bar\_size\_designation\_line\_type" minOccurs="0" type="xsd:bar\_size\_designation\_line\_type"/>  
<element name="bar\_size\_designation\_single\_type" minOccurs="0" type="xsd:bar\_size\_designation\_single\_type"/>  
<element name="bar\_size\_designation\_corner\_type" minOccurs="0" type="xsd:bar\_size\_designation\_corner\_type"/>  
<element name="corner\_reinforcement\_enabled" minOccurs="0" type="boolean"/>  
<element name="reinforcement\_area\_symmetrical" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_unsymmetrical\_at\_side" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_unsymmetrical\_top\_side" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_unsymmetrical\_bottom\_side" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_uniformly\_surrounding" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_line" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_single" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_corner" minOccurs="0" type="double"/>  
<element name="reinforcement\_area\_total" minOccurs="0" type="double"/>  
<element name="span\_position\_reference\_internal\_node" minOccurs="0" type="int"/>  
<element name="span\_position\_reference\_x\_location\_relative" minOccurs="0" type="double"/>  
<element name="span\_position\_reference\_x\_location\_absolute" minOccurs="0" type="double"/>  
<element name="span\_start\_relative" minOccurs="0" type="double"/>  
<element name="span\_start\_absolute" minOccurs="0" type="double"/>  
<element name="span\_end\_relative" minOccurs="0" type="double"/>  
<element name="span\_end\_absolute" minOccurs="0" type="double"/>  
<element name="span\_length" minOccurs="0" type="double"/>  
<element name="additional\_offset\_type" minOccurs="0" type="xsd:additional\_offset\_type"/>  
<element name="additional\_offset\_type\_single\_line\_type" minOccurs="0" type="xsd:additional\_offset\_type\_single\_line\_type"/>  
<element name="additional\_offset\_top\_side" minOccurs="0" type="double"/>  
<element name="additional\_offset\_bottom\_side" minOccurs="0" type="double"/>  
<element name="additional\_offset\_left\_side" minOccurs="0" type="double"/>  
<element name="additional\_offset\_right\_side" minOccurs="0" type="double"/>  
<element name="additional\_offset\_reference\_type" minOccurs="0" type="xsd:additional\_offset\_reference\_type"/>  
<element name="additional\_offset\_reference\_type\_at\_start\_type" minOccurs="0" type="xsd:additional\_offset\_reference\_type\_at\_start\_type"/>  
<element name="additional\_offset\_reference\_type\_at\_end\_type" minOccurs="0" type="xsd:additional\_offset\_reference\_type\_at\_end\_type"/>  
<element name="additional\_horizontal\_offset" minOccurs="0" type="double"/>  
<element name="additional\_horizontal\_offset\_at\_start" minOccurs="0" type="double"/>  
<element name="additional\_horizontal\_offset\_at\_end" minOccurs="0" type="double"/>  
<element name="additional\_vertical\_offset" minOccurs="0" type="double"/>  
<element name="additional\_vertical\_offset\_at\_start" minOccurs="0" type="double"/>  
<element name="additional\_vertical\_offset\_at\_end" minOccurs="0" type="double"/>  
<element name="anchorage\_start\_anchor\_type" minOccurs="0" type="xsd:anchorage\_start\_anchor\_type"/>  
<element name="anchorage\_start\_anchor\_length" minOccurs="0" type="double"/>  
<element name="anchorage\_start\_bending\_diameter" minOccurs="0" type="double"/>  
<element name="anchorage\_end\_anchor\_type" minOccurs="0" type="xsd:anchorage\_end\_anchor\_type"/>  
<element name="anchorage\_end\_anchor\_length" minOccurs="0" type="double"/>  
<element name="anchorage\_end\_bending\_diameter" minOccurs="0" type="double"/>  
<element name="one\_rebar\_length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_minimal\_and\_maximal\_length" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_at\_side\_length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_length" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_top\_side\_length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_length" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_bottom\_side\_length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_length" minOccurs="0" type="string"/>  
<element name="one\_rebar\_corner\_length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_corner\_minimal\_and\_maximal\_length" minOccurs="0" type="string"/>  
<element name="length" minOccurs="0" type="double"/>  
<element name="one\_rebar\_weight" minOccurs="0" type="double"/>  
<element name="one\_rebar\_minimal\_and\_maximal\_weight" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_at\_side\_weight" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_at\_side\_minimal\_and\_maximal\_weight" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_top\_side\_weight" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_top\_side\_minimal\_and\_maximal\_weight" minOccurs="0" type="string"/>  
<element name="one\_rebar\_unsymmetrical\_bottom\_side\_weight" minOccurs="0" type="double"/>  
<element name="one\_rebar\_unsymmetrical\_bottom\_side\_minimal\_and\_maximal\_weight" minOccurs="0" type="string"/>  
<element name="one\_rebar\_corner\_weight" minOccurs="0" type="double"/>  
<element name="one\_rebar\_corner\_minimal\_and\_maximal\_weight" minOccurs="0" type="string"/>  
<element name="weight" minOccurs="0" type="double"/>

```

 </sequence>
 </complexType>
 <complexType name="array_of_member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items">
 <sequence>
 <element name="member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items"
 type="xsd:member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items"
 type="xsd:array_of_member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items"/>
 <simpleType name="member_set_representative_member_set_model_deflection_check_direction">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Y"/>
 <enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z"/>
 <enumeration value="DEFLECTION_CHECK_DIRECTION_LOCAL_AXIS_Z_AND_Y"/>
 <enumeration value="DEFLECTION_CHECK_DIRECTION_RESULTING_AXIS"/>
 </restriction>
 </simpleType>
 <simpleType name="member_set_representative_member_set_model_deflection_check_displacement_reference">
 <restriction base="string">
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_SEGMENT_ENDS"/>
 <enumeration value="DEFLECTION_CHECK_DISPLACEMENT_REFERENCE_DEFORMED_UNDEFORMED_SYSTEM"/>
 </restriction>
 </simpleType>
 <complexType name="member_set_representative_member_set_model_design_supports_on_internal_nodes">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="node" minOccurs="0" type="int"/>
 <element name="design_support" minOccurs="0" type="int"/>
 </sequence>
 </complexType>
 <complexType name="array_of_member_set_representative_member_set_model_design_supports_on_internal_nodes">
 <sequence>
 <element name="member_set_representative_member_set_model_design_supports_on_internal_nodes"
 type="xsd:member_set_representative_member_set_model_design_supports_on_internal_nodes" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_member_set_representative_member_set_model_design_supports_on_internal_nodes"
 type="xsd:array_of_member_set_representative_member_set_model_design_supports_on_internal_nodes"/>
 <complexType name="member_set_representative_member_model_deflection_segments_y_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
 </complexType>
 <complexType name="array_of_member_set_representative_member_model_deflection_segments_y_axis">
 <sequence>
 <element name="member_set_representative_member_model_deflection_segments_y_axis"
 type="xsd:member_set_representative_member_model_deflection_segments_y_axis" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_member_set_representative_member_model_deflection_segments_y_axis"
 type="xsd:array_of_member_set_representative_member_model_deflection_segments_y_axis"/>
 <complexType name="member_set_representative_member_model_deflection_segments_z_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
 </complexType>
 <complexType name="array_of_member_set_representative_member_model_deflection_segments_z_axis">
 <sequence>
 <element name="member_set_representative_member_model_deflection_segments_z_axis"
 type="xsd:member_set_representative_member_model_deflection_segments_z_axis" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_member_set_representative_member_model_deflection_segments_z_axis"
 type="xsd:array_of_member_set_representative_member_model_deflection_segments_z_axis"/>
 <complexType name="member_set_representative_member_set_model_deflection_segments_y_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
 </complexType>

```

```

</complexType>
<complexType name="array_of_member_set_representative_member_set_model_deflection_segments_y_axis">
 <sequence>
 <element name="member_set_representative_member_set_model_deflection_segments_y_axis"
 type="xsd:member_set_representative_member_set_model_deflection_segments_y_axis" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_set_representative_member_set_model_deflection_segments_y_axis"
 type="xsd:array_of_member_set_representative_member_set_model_deflection_segments_y_axis"/>
<complexType name="member_set_representative_member_set_model_deflection_segments_z_axis">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="active" minOccurs="0" type="boolean"/>
 <element name="length" minOccurs="0" type="double"/>
 <element name="precamber" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_member_set_representative_member_set_model_deflection_segments_z_axis">
 <sequence>
 <element name="member_set_representative_member_set_model_deflection_segments_z_axis"
 type="xsd:member_set_representative_member_set_model_deflection_segments_z_axis" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_member_set_representative_member_set_model_deflection_segments_z_axis"
 type="xsd:array_of_member_set_representative_member_set_model_deflection_segments_z_axis"/>
<complexType name="get_member_set_representative">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_member_set_representative" type="xsd:get_member_set_representative"/>
<complexType name="get_member_set_representativeResponse">
 <sequence>
 <element name="value" type="xsd:member_set_representative"/>
 </sequence>
</complexType>
<element name="get_member_set_representativeResponse" type="xsd:get_member_set_representativeResponse"/>
<complexType name="set_member_set_representative">
 <sequence>
 <element name="value" type="xsd:member_set_representative"/>
 </sequence>
</complexType>
<element name="set_member_set_representative" type="xsd:set_member_set_representative"/>
<complexType name="set_member_set_representativeResponse">
 <sequence/>
</complexType>
<element name="set_member_set_representativeResponse" type="xsd:set_member_set_representativeResponse"/>
<complexType name="member_set_representative">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="member_sets" minOccurs="0" type="xsd:array_of_int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="number_of_member_sets" minOccurs="0" type="int"/>
 <element name="number_of_members" minOccurs="0" type="int"/>
 <element name="section_ids" minOccurs="0" type="string"/>
 <element name="total_length" minOccurs="0" type="double"/>
 <element name="total_volume" minOccurs="0" type="double"/>
 <element name="total_mass" minOccurs="0" type="double"/>
 <element name="total_surface_of_coating" minOccurs="0" type="double"/>
 <element name="total_center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="total_center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="total_center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="member_model_no" minOccurs="0" type="int"/>
 <element name="member_model_type" minOccurs="0" type="xsd:member_set_representative_member_model_type"/>
 <element name="member_model_is_deactivated_for_calculation" minOccurs="0" type="boolean"/>
 <element name="member_model_line" minOccurs="0" type="int"/>
 <element name="member_model_section_distribution_type" minOccurs="0"
 type="xsd:member_set_representative_member_model_section_distribution_type"/>
 <element name="member_model_reference_type" minOccurs="0"
 type="xsd:member_set_representative_member_model_reference_type"/>
 <element name="member_model_nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="member_model_node_start" minOccurs="0" type="int"/>
 <element name="member_model_node_end" minOccurs="0" type="int"/>
 <element name="member_model_analytical_length" minOccurs="0" type="double"/>
 <element name="member_model_analytical_volume" minOccurs="0" type="double"/>
 </sequence>

```

```
<element name="member_model_analytical_mass" minOccurs="0" type="double"/>
<element name="member_model_analytical_surface_of_coating" minOccurs="0" type="double"/>
<element name="member_model_analytical_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="member_model_analytical_center_of_gravity_x" minOccurs="0" type="double"/>
<element name="member_model_analytical_center_of_gravity_y" minOccurs="0" type="double"/>
<element name="member_model_analytical_center_of_gravity_z" minOccurs="0" type="double"/>
<element name="member_model_length" minOccurs="0" type="double"/>
<element name="member_model_volume" minOccurs="0" type="double"/>
<element name="member_model_mass" minOccurs="0" type="double"/>
<element name="member_model_surface_of_coating" minOccurs="0" type="double"/>
<element name="member_model_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="member_model_center_of_gravity_x" minOccurs="0" type="double"/>
<element name="member_model_center_of_gravity_y" minOccurs="0" type="double"/>
<element name="member_model_center_of_gravity_z" minOccurs="0" type="double"/>
<element name="member_set_model_position" minOccurs="0" type="string"/>
<element name="member_set_model_position_short" minOccurs="0" type="string"/>
<element name="member_model_member_representative" minOccurs="0" type="int"/>
<element name="member_model_design_properties_via_member" minOccurs="0" type="boolean"/>
<element name="member_model_design_properties_via_parent_member_set" minOccurs="0" type="boolean"/>
<element name="member_model_design_properties_parent_member_set" minOccurs="0" type="int"/>
<element name="member_model_comment" minOccurs="0" type="string"/>
<element name="member_model_member_type_rib_alignment" minOccurs="0"
type="xsd:member_set_representative_member_model_member_type_rib_alignment"/>
<element name="member_model_member_rib_first_surface" minOccurs="0" type="int"/>
<element name="member_model_member_rib_second_surface" minOccurs="0" type="int"/>
<element name="member_model_member_rib_surface_assignment_autodetect" minOccurs="0" type="boolean"/>
<element name="member_model_member_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled" minOccurs="0"
type="boolean"/>
<element name="member_model_align_local_z_axis_to_local_z_axis_of_surface" minOccurs="0" type="boolean"/>
<element name="member_model_flange_dimensions" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_flange_dimensions"/>
<element name="member_model_synchronize_width_mode" minOccurs="0" type="boolean"/>
<element name="member_model_relative Ordinates_mode" minOccurs="0" type="boolean"/>
<element name="member_model_member_type_definable_stiffness" minOccurs="0" type="int"/>
<element name="member_model_result_beam_integrate_stresses_and_forces" minOccurs="0"
type="xsd:member_set_representative_member_model_result_beam_integrate_stresses_and_forces"/>
<element name="member_model_result_beam_y_z" minOccurs="0" type="double"/>
<element name="member_model_result_beam_y_plus" minOccurs="0" type="double"/>
<element name="member_model_result_beam_z_plus" minOccurs="0" type="double"/>
<element name="member_model_result_beam_y_minus" minOccurs="0" type="double"/>
<element name="member_model_result_beam_z_minus" minOccurs="0" type="double"/>
<element name="member_model_result_beam_radius" minOccurs="0" type="double"/>
<element name="member_model_result_beam_include_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_result_beam_include_all_surfaces" minOccurs="0" type="boolean"/>
<element name="member_model_result_beam_include_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_result_beam_include_all_solids" minOccurs="0" type="boolean"/>
<element name="member_model_result_beam_include_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_result_beam_include_all_members" minOccurs="0" type="boolean"/>
<element name="member_model_result_beam_exclude_surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_result_beam_exclude_solids" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_result_beam_exclude_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_projected_length" minOccurs="0" type="double"/>
<element name="member_model_section_distance_from_start_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="member_model_section_distance_from_start_absolute" minOccurs="0" type="double"/>
<element name="member_model_section_distance_from_end_relative" minOccurs="0" type="double"/>
<element name="member_model_section_distance_from_end_is_defined_as_relative" minOccurs="0" type="boolean"/>
<element name="member_model_section_distance_from_end_absolute" minOccurs="0" type="double"/>
<element name="member_model_section_distance_from_start_relative" minOccurs="0" type="double"/>
<element name="member_model_section_alignment" minOccurs="0"
type="xsd:member_set_representative_member_model_section_alignment"/>
<element name="member_model_curved_member_is_asymmetric_layout" minOccurs="0" type="boolean"/>
<element name="member_model_curved_member_is_cantilevers" minOccurs="0" type="boolean"/>
<element name="member_model_curved_member_cantilevers_type" minOccurs="0"
type="xsd:member_set_representative_member_model_curved_member_cantilevers_type"/>
<element name="member_model_rotation_specification_type" minOccurs="0"
type="xsd:member_set_representative_member_model_rotation_specification_type"/>
<element name="member_model_rotation_angle" minOccurs="0" type="double"/>
<element name="member_model_rotation_help_node" minOccurs="0" type="int"/>
<element name="member_model_rotation_plane_type" minOccurs="0"
type="xsd:member_set_representative_member_model_rotation_plane_type"/>
<element name="member_model_rotation_surface" minOccurs="0" type="int"/>
<element name="member_model_rotation_surface_plane_type" minOccurs="0"
type="xsd:member_set_representative_member_model_rotation_surface_plane_type"/>
<element name="member_model_is_rotated" minOccurs="0" type="boolean"/>
<element name="member_model_section_start" minOccurs="0" type="int"/>
<element name="member_model_section_end" minOccurs="0" type="int"/>
```

```
<element name="member_model_section_internal" minOccurs="0" type="int"/>
<element name="member_model_section_material" minOccurs="0" type="int"/>
<element name="member_model_member_hinge_start" minOccurs="0" type="int"/>
<element name="member_model_member_hinge_end" minOccurs="0" type="int"/>
<element name="member_model_member_eccentricity_start" minOccurs="0" type="int"/>
<element name="member_model_member_eccentricity_end" minOccurs="0" type="int"/>
<element name="member_model_support" minOccurs="0" type="int"/>
<element name="member_model_member_transverse_stiffener" minOccurs="0" type="int"/>
<element name="member_model_member_nonlinearity" minOccurs="0" type="int"/>
<element name="member_model_member_result_intermediate_point" minOccurs="0" type="int"/>
<element name="member_model_concrete_effective_lengths" minOccurs="0" type="int"/>
<element name="member_model_steel_effective_lengths" minOccurs="0" type="int"/>
<element name="member_model_timber_effective_lengths" minOccurs="0" type="int"/>
<element name="member_model_aluminum_effective_lengths" minOccurs="0" type="int"/>
<element name="member_model_concrete_shear_reinforcement_spans" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_concrete_shear_reinforcement_spans"/>
<element name="member_model_concrete_longitudinal_reinforcement_items" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_concrete_longitudinal_reinforcement_items"/>
<element name="member_model_concrete_cover_user_defined_enabled" minOccurs="0" type="boolean"/>
<element name="member_model_concrete_cover_different_at_section_sides_enabled" minOccurs="0" type="boolean"/>
<element name="member_model_concrete_cover" minOccurs="0" type="double"/>
<element name="member_model_concrete_cover_top" minOccurs="0" type="double"/>
<element name="member_model_concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="member_model_concrete_cover_left" minOccurs="0" type="double"/>
<element name="member_model_concrete_cover_right" minOccurs="0" type="double"/>
<element name="member_model_concrete_cover_min" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_concrete_cover_min_top" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_concrete_cover_min_bottom" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_concrete_cover_min_left" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_concrete_cover_min_right" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_model_concrete_durability" minOccurs="0" type="int"/>
<element name="member_model_concrete_durability_top" minOccurs="0" type="int"/>
<element name="member_model_concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="member_model_concrete_durability_left" minOccurs="0" type="int"/>
<element name="member_model_concrete_durability_right" minOccurs="0" type="int"/>
<element name="member_model_steel_boundary_conditions" minOccurs="0" type="int"/>
<element name="member_model_steel_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="member_model_steel_member_transverse_weld" minOccurs="0" type="int"/>
<element name="member_model_steel_member_shear_panel" minOccurs="0" type="int"/>
<element name="member_model_steel_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_model_timber_service_class" minOccurs="0" type="int"/>
<element name="member_model_timber_moisture_class" minOccurs="0" type="int"/>
<element name="member_model_timber_service_conditions" minOccurs="0" type="int"/>
<element name="member_model_timber_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="member_model_timber_member_shear_panel" minOccurs="0" type="int"/>
<element name="member_model_timber_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_model_aluminum_boundary_conditions" minOccurs="0" type="int"/>
<element name="member_model_aluminum_member_local_section_reduction" minOccurs="0" type="int"/>
<element name="member_model_aluminum_member_transverse_weld" minOccurs="0" type="int"/>
<element name="member_model_aluminum_member_shear_panel" minOccurs="0" type="int"/>
<element name="member_model_aluminum_member_rotational_restraint" minOccurs="0" type="int"/>
<element name="member_model_stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_aluminum_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_aluminum_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_steel_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_steel_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_steel_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_timber_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_timber_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_model_member_timber_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_model_end_modifications_member_start_extension" minOccurs="0" type="double"/>
<element name="member_model_end_modifications_member_start_slope_y" minOccurs="0" type="double"/>
<element name="member_model_end_modifications_member_start_slope_z" minOccurs="0" type="double"/>
<element name="member_model_end_modifications_member_end_extension" minOccurs="0" type="double"/>
<element name="member_model_end_modifications_member_end_slope_y" minOccurs="0" type="double"/>
<element name="member_model_end_modifications_member_end_slope_z" minOccurs="0" type="double"/>
<element name="member_model_has_any_end_modifications" minOccurs="0" type="boolean"/>
<element name="member_model_deflection_check_direction" minOccurs="0"
type="xsd:member_set_representative_member_model_deflection_check_direction"/>
```

```
<element name="member_model_deflection_check_displacement_reference" minOccurs="0"
type="xsd:member_set_representative_member_model_deflection_check_displacement_reference"/>
<element name="member_model_design_support_on_member_start" minOccurs="0" type="int"/>
<element name="member_model_design_support_on_member_end" minOccurs="0" type="int"/>
<element name="member_model_design_supports_on_internal_nodes" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_design_supports_on_internal_nodes"/>
<element name="member_model_is_generated" minOccurs="0" type="boolean"/>
<element name="member_model_generating_object_info" minOccurs="0" type="string"/>
<element name="member_set_model_no" minOccurs="0" type="int"/>
<element name="member_set_model_user_defined_name_enabled" minOccurs="0" type="boolean"/>
<element name="member_set_model_name" minOccurs="0" type="string"/>
<element name="member_set_model_set_type" minOccurs="0"
type="xsd:member_set_representative_member_set_model_set_type"/>
<element name="member_set_model_members" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_length" minOccurs="0" type="double"/>
<element name="member_set_model_center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="member_set_model_center_of_gravity_x" minOccurs="0" type="double"/>
<element name="member_set_model_center_of_gravity_y" minOccurs="0" type="double"/>
<element name="member_set_model_center_of_gravity_z" minOccurs="0" type="double"/>
<element name="member_set_model_volume" minOccurs="0" type="double"/>
<element name="member_set_model_mass" minOccurs="0" type="double"/>
<element name="member_set_model_member_set_representative" minOccurs="0" type="int"/>
<element name="member_set_model_discontinuous_torsional_warping" minOccurs="0" type="boolean"/>
<element name="member_set_model_design_properties_activated" minOccurs="0" type="boolean"/>
<element name="member_set_model_steel_effective_lengths" minOccurs="0" type="int"/>
<element name="member_set_model_steel_boundary_conditions" minOccurs="0" type="int"/>
<element name="member_set_model_steel_member_local_section_reductions" minOccurs="0" type="int"/>
<element name="member_set_model_steel_member_shear_panels" minOccurs="0" type="int"/>
<element name="member_set_model_steel_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="member_set_model_member_steel_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_steel_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_steel_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_steel_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_effective_lengths" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_boundary_conditions" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_member_local_section_reductions" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_member_shear_panels" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="member_set_model_aluminum_member_transverse_weld" minOccurs="0" type="int"/>
<element name="member_set_model_member_aluminum_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_aluminum_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_shear_reinforcement_spans" minOccurs="0"
type="xsd:array_of_member_set_representative_member_set_model_concrete_shear_reinforcement_spans"/>
<element name="member_set_model_concrete_longitudinal_reinforcement_items" minOccurs="0"
type="xsd:array_of_member_set_representative_member_set_model_concrete_longitudinal_reinforcement_items"/>
<element name="member_set_model_concrete_cover_user_defined_enabled" minOccurs="0" type="boolean"/>
<element name="member_set_model_concrete_cover_different_at_section_sides_enabled" minOccurs="0" type="boolean"/>
<element name="member_set_model_concrete_cover" minOccurs="0" type="double"/>
<element name="member_set_model_concrete_cover_top" minOccurs="0" type="double"/>
<element name="member_set_model_concrete_cover_bottom" minOccurs="0" type="double"/>
<element name="member_set_model_concrete_cover_left" minOccurs="0" type="double"/>
<element name="member_set_model_concrete_cover_right" minOccurs="0" type="double"/>
<element name="member_set_model_concrete_cover_min" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_concrete_cover_min_top" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_concrete_cover_min_bottom" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_concrete_cover_min_left" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_concrete_cover_min_right" minOccurs="0" type="xsd:array_of_int"/>
<element name="member_set_model_concrete_durability" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_durability_top" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_durability_bottom" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_durability_left" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_durability_right" minOccurs="0" type="int"/>
<element name="member_set_model_concrete_effective_lengths" minOccurs="0" type="int"/>
<element name="member_set_model_member_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_service_class_timber_design" minOccurs="0" type="int"/>
<element name="member_set_model_moisture_class_timber_design" minOccurs="0" type="int"/>
<element name="member_set_model_timber_effective_lengths" minOccurs="0" type="int"/>
<element name="member_set_model_service_conditions_timber_design" minOccurs="0" type="int"/>
<element name="member_set_model_timber_member_shear_panels" minOccurs="0" type="int"/>
<element name="member_set_model_timber_member_rotational_restraints" minOccurs="0" type="int"/>
<element name="member_set_model_timber_local_section_reductions" minOccurs="0" type="int"/>
<element name="member_set_model_member_timber_design_uls_configuration" minOccurs="0" type="int"/>
```

```

<element name="member_set_model_member_timber_design_sls_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_timber_design_fr_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_member_transverse_stiffener" minOccurs="0" type="int"/>
<element name="member_set_model_stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="member_set_model_deflection_check_direction" minOccurs="0"
type="xsd:member_set_representative_member_set_model_deflection_check_direction"/>
<element name="member_set_model_deflection_check_displacement_reference" minOccurs="0"
type="xsd:member_set_representative_member_set_model_deflection_check_displacement_reference"/>
<element name="member_set_model_design_support_on_member_set_start" minOccurs="0" type="int"/>
<element name="member_set_model_design_support_on_member_set_end" minOccurs="0" type="int"/>
<element name="member_set_model_design_supports_on_internal_nodes" minOccurs="0"
type="xsd:array_of_member_set_representative_member_set_model_design_supports_on_internal_nodes"/>
<element name="member_set_model_comment" minOccurs="0" type="string"/>
<element name="member_set_model_is_generated" minOccurs="0" type="boolean"/>
<element name="member_set_model_generating_object_info" minOccurs="0" type="string"/>
<element name="member_set_model_member_set_rib_generating_longitudinal_reinforcement_items_from_surfaces_enabled"
minOccurs="0" type="boolean"/>
<element name="member_model_deflection_segments_defined_length_y_axis_enabled" minOccurs="0" type="boolean"/>
<element name="member_model_deflection_segments_defined_length_z_axis_enabled" minOccurs="0" type="boolean"/>
<element name="member_model_deflection_segments_y_axis" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_deflection_segments_y_axis"/>
<element name="member_model_deflection_segments_z_axis" minOccurs="0"
type="xsd:array_of_member_set_representative_member_model_deflection_segments_z_axis"/>
<element name="member_set_model_deflection_segments_defined_length_y_axis_enabled" minOccurs="0" type="boolean"/>
<element name="member_set_model_deflection_segments_defined_length_z_axis_enabled" minOccurs="0" type="boolean"/>
<element name="member_set_model_deflection_segments_y_axis" minOccurs="0"
type="xsd:array_of_member_set_representative_member_set_model_deflection_segments_y_axis"/>
<element name="member_set_model_deflection_segments_z_axis" minOccurs="0"
type="xsd:array_of_member_set_representative_member_set_model_deflection_segments_z_axis"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="surface_set_set_type">
<restriction base="string">
<enumeration value="SET_TYPE_CONTINUOUS"/>
<enumeration value="SET_TYPE_GROUP"/>
</restriction>
</simpleType>
<complexType name="get_surface_set">
<sequence>
<element name="no" type="int"/>
</sequence>
</complexType>
<element name="get_surface_set" type="xsd:get_surface_set"/>
<complexType name="get_surface_setResponse">
<sequence>
<element name="value" type="xsd:surface_set"/>
</sequence>
</complexType>
<element name="get_surface_setResponse" type="xsd:get_surface_setResponse"/>
<complexType name="set_surface_set">
<sequence>
<element name="value" type="xsd:surface_set"/>
</sequence>
</complexType>
<element name="set_surface_set" type="xsd:set_surface_set"/>
<complexType name="set_surface_setResponse">
<sequence/>
</complexType>
<element name="set_surface_setResponse" type="xsd:set_surface_setResponse"/>
<complexType name="surface_set">
<sequence>
<element name="no" type="int"/>
<element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
<element name="name" minOccurs="0" type="string"/>
<element name="set_type" minOccurs="0" type="xsd:surface_set_set_type"/>
<element name="surfaces" minOccurs="0" type="xsd:array_of_int"/>
<element name="surface_area" minOccurs="0" type="double"/>
<element name="volume" minOccurs="0" type="double"/>
<element name="mass" minOccurs="0" type="double"/>
<element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
<element name="center_of_gravity_x" minOccurs="0" type="double"/>
<element name="center_of_gravity_y" minOccurs="0" type="double"/>
<element name="center_of_gravity_z" minOccurs="0" type="double"/>
<element name="position" minOccurs="0" type="string"/>
<element name="position_short" minOccurs="0" type="string"/>

```

```

<element name="stress_analysis_configuration" minOccurs="0" type="int"/>
<element name="timber_service_class" minOccurs="0" type="int"/>
<element name="timber_moisture_class" minOccurs="0" type="int"/>
<element name="timber_service_conditions" minOccurs="0" type="int"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="surface_concrete_design_uls_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_sls_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_fr_configuration" minOccurs="0" type="int"/>
<element name="surface_concrete_design_seismic_configuration" minOccurs="0" type="int"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="solid_set_set_type">
 <restriction base="string">
 <enumeration value="SET_TYPE_CONTINUOUS"/>
 <enumeration value="SET_TYPE_GROUP"/>
 </restriction>
</simpleType>
<complexType name="get_solid_set">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_solid_set" type="xsd:get_solid_set"/>
<complexType name="get_solid_setResponse">
 <sequence>
 <element name="value" type="xsd:solid_set"/>
 </sequence>
</complexType>
<element name="get_solid_setResponse" type="xsd:get_solid_setResponse"/>
<complexType name="set_solid_set">
 <sequence>
 <element name="value" type="xsd:solid_set"/>
 </sequence>
</complexType>
<element name="set_solid_set" type="xsd:set_solid_set"/>
<complexType name="set_solid_setResponse">
 <sequence/>
</complexType>
<element name="set_solid_setResponse" type="xsd:set_solid_setResponse"/>
<complexType name="solid_set">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="set_type" minOccurs="0" type="xsd:solid_set_set_type"/>
 <element name="solids" minOccurs="0" type="xsd:array_of_int"/>
 <element name="surface_area" minOccurs="0" type="double"/>
 <element name="volume" minOccurs="0" type="double"/>
 <element name="mass" minOccurs="0" type="double"/>
 <element name="center_of_gravity" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_of_gravity_x" minOccurs="0" type="double"/>
 <element name="center_of_gravity_y" minOccurs="0" type="double"/>
 <element name="center_of_gravity_z" minOccurs="0" type="double"/>
 <element name="stress_analysis_configuration" minOccurs="0" type="int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="get_intersection">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_intersection" type="xsd:get_intersection"/>
<complexType name="get_intersectionResponse">
 <sequence>
 <element name="value" type="xsd:intersection"/>
 </sequence>
</complexType>

```



```

<element name="get_intersectionResponse" type="xsd:get_intersectionResponse"/>
<complexType name="set_intersection">
 <sequence>
 <element name="value" type="xsd:intersection"/>
 </sequence>
</complexType>
<element name="set_intersection" type="xsd:set_intersection"/>
<complexType name="set_intersectionResponse">
 <sequence/>
</complexType>
<element name="set_intersectionResponse" type="xsd:set_intersectionResponse"/>
<complexType name="intersection">
 <sequence>
 <element name="no" type="int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="generated_lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="generated_nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="surface_a" minOccurs="0" type="int"/>
 <element name="surface_b" minOccurs="0" type="int"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="surface_results_adjustment_shape">
 <restriction base="string">
 <enumeration value="SHAPE_CIRCLE"/>
 <enumeration value="SHAPE_ELLIPSE"/>
 <enumeration value="SHAPE_RECTANGLE"/>
 </restriction>
</simpleType>
<simpleType name="surface_results_adjustment_adjustment_type_in_direction_u">
 <restriction base="string">
 <enumeration value="AVERAGING_OF_MX_MXY_VX_NX_NXY"/>
 <enumeration value="AVERAGING_OF_MY_MXY_VY_NY_NXY"/>
 <enumeration value="NONE"/>
 <enumeration value="USER_DEFINED"/>
 <enumeration value="ZERO"/>
 </restriction>
</simpleType>
<simpleType name="surface_results_adjustment_adjustment_type_in_direction_v">
 <restriction base="string">
 <enumeration value="AVERAGING_OF_MX_MXY_VX_NX_NXY"/>
 <enumeration value="AVERAGING_OF_MY_MXY_VY_NY_NXY"/>
 <enumeration value="NONE"/>
 <enumeration value="USER_DEFINED"/>
 <enumeration value="ZERO"/>
 </restriction>
</simpleType>
<simpleType name="surface_results_adjustment_projection_in_direction_type">
 <restriction base="string">
 <enumeration value="GLOBAL_IN_X"/>
 <enumeration value="GLOBAL_IN_Y"/>
 <enumeration value="GLOBAL_IN_Z"/>
 <enumeration value="PERPENDICULAR"/>
 <enumeration value="VECTOR"/>
 </restriction>
</simpleType>
<complexType name="get_surface_results_adjustment">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_surface_results_adjustment" type="xsd:get_surface_results_adjustment"/>
<complexType name="get_surface_results_adjustmentResponse">
 <sequence>
 <element name="value" type="xsd:surface_results_adjustment"/>
 </sequence>
</complexType>
<element name="get_surface_results_adjustmentResponse" type="xsd:get_surface_results_adjustmentResponse"/>
<complexType name="set_surface_results_adjustment">
 <sequence>
 <element name="value" type="xsd:surface_results_adjustment"/>
 </sequence>
</complexType>
<element name="set_surface_results_adjustment" type="xsd:set_surface_results_adjustment"/>
<complexType name="set_surface_results_adjustmentResponse">
 <sequence/>
</complexType>

```

```

</complexType>
<element name="set_surface_results_adjustmentResponse" type="xsd:set_surface_results_adjustmentResponse"/>
<complexType name="surface_results_adjustment">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="shape" minOccurs="0" type="xsd:surface_results_adjustment_shape"/>
 <element name="dimension_1" minOccurs="0" type="double"/>
 <element name="dimension_2" minOccurs="0" type="double"/>
 <element name="angular_rotation" minOccurs="0" type="double"/>
 <element name="center_position" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_position_x" minOccurs="0" type="double"/>
 <element name="center_position_y" minOccurs="0" type="double"/>
 <element name="center_position_z" minOccurs="0" type="double"/>
 <element name="adjustment_type_in_direction_u" minOccurs="0" type="xsd:surface_results_adjustment_adjustment_type_in_direction_u"/>
 <element name="adjustment_type_in_direction_v" minOccurs="0" type="xsd:surface_results_adjustment_adjustment_type_in_direction_v"/>
 <element name="projection_in_direction_type" minOccurs="0" type="xsd:surface_results_adjustment_projection_in_direction_type"/>
 <element name="vector_of_projection_in_direction_coordinates" minOccurs="0" type="xsd:vector_3d"/>
 <element name="vector_of_projection_in_direction_coordinates_x" minOccurs="0" type="double"/>
 <element name="vector_of_projection_in_direction_coordinates_y" minOccurs="0" type="double"/>
 <element name="vector_of_projection_in_direction_coordinates_z" minOccurs="0" type="double"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="get_surfaces_contact">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_surfaces_contact" type="xsd:get_surfaces_contact"/>
<complexType name="get_surfaces_contactResponse">
 <sequence>
 <element name="value" type="xsd:surfaces_contact"/>
 </sequence>
</complexType>
<element name="get_surfaces_contactResponse" type="xsd:get_surfaces_contactResponse"/>
<complexType name="set_surfaces_contact">
 <sequence>
 <element name="value" type="xsd:surfaces_contact"/>
 </sequence>
</complexType>
<element name="set_surfaces_contact" type="xsd:set_surfaces_contact"/>
<complexType name="set_surfaces_contactResponse">
 <sequence/>
</complexType>
<element name="set_surfaces_contactResponse" type="xsd:set_surfaces_contactResponse"/>
<complexType name="surfaces_contact">
 <sequence>
 <element name="no" type="int"/>
 <element name="surfaces_contact_type" minOccurs="0" type="int"/>
 <element name="surfaces_group1" minOccurs="0" type="xsd:array_of_int"/>
 <element name="surfaces_group2" minOccurs="0" type="xsd:array_of_int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="rigid_link_type">
 <restriction base="string">
 <enumeration value="TYPE_DIAPHRAGM"/>
 <enumeration value="TYPE_LINE_TO_LINE"/>
 <enumeration value="TYPE_LINE_TO_SURFACE"/>
 </restriction>
</simpleType>
<simpleType name="rigid_link_rigid_link_type">
 <restriction base="string">
 <enumeration value="RESILIENT"/>
 <enumeration value="RIGID"/>
 </restriction>
</simpleType>
<complexType name="get_rigid_link">

```

```

 <sequence>
 <element name="no" type="int"/>
 </sequence>
 </complexType>
 <element name="get_rigid_link" type="xsd:get_rigid_link"/>
 <complexType name="get_rigid_linkResponse">
 <sequence>
 <element name="value" type="xsd:rigid_link"/>
 </sequence>
 </complexType>
 <element name="get_rigid_linkResponse" type="xsd:get_rigid_linkResponse"/>
 <complexType name="set_rigid_link">
 <sequence>
 <element name="value" type="xsd:rigid_link"/>
 </sequence>
 </complexType>
 <element name="set_rigid_link" type="xsd:set_rigid_link"/>
 <complexType name="set_rigid_linkResponse">
 <sequence/>
 </complexType>
 <element name="set_rigid_linkResponse" type="xsd:set_rigid_linkResponse"/>
 <complexType name="rigid_link">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:rigid_link_type"/>
 <element name="line1" minOccurs="0" type="int"/>
 <element name="line2" minOccurs="0" type="int"/>
 <element name="surface" minOccurs="0" type="int"/>
 <element name="rigid_link_type" minOccurs="0" type="xsd:rigid_link_rigid_link_type"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="ignore_relative_position" minOccurs="0" type="boolean"/>
 <element name="user_defined_distribution" minOccurs="0" type="boolean"/>
 <element name="line1_start_is_relative" minOccurs="0" type="boolean"/>
 <element name="line1_start_relative" minOccurs="0" type="double"/>
 <element name="line1_start_absolute" minOccurs="0" type="double"/>
 <element name="line1_end_is_relative" minOccurs="0" type="boolean"/>
 <element name="line1_end_relative" minOccurs="0" type="double"/>
 <element name="line1_end_absolute" minOccurs="0" type="double"/>
 <element name="line2_start_is_relative" minOccurs="0" type="boolean"/>
 <element name="line2_start_relative" minOccurs="0" type="double"/>
 <element name="line2_start_absolute" minOccurs="0" type="double"/>
 <element name="line2_end_is_relative" minOccurs="0" type="boolean"/>
 <element name="line2_end_relative" minOccurs="0" type="double"/>
 <element name="line2_end_absolute" minOccurs="0" type="double"/>
 <element name="nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="center_node_no" minOccurs="0" type="int"/>
 <element name="center_user_defined" minOccurs="0" type="boolean"/>
 <element name="center_point" minOccurs="0" type="xsd:vector_3d"/>
 <element name="center_point_x" minOccurs="0" type="double"/>
 <element name="center_point_y" minOccurs="0" type="double"/>
 <element name="center_point_z" minOccurs="0" type="double"/>
 <element name="link_plane_user_defined" minOccurs="0" type="boolean"/>
 <element name="link_plane_node1_no" minOccurs="0" type="int"/>
 <element name="link_plane_node2_no" minOccurs="0" type="int"/>
 <element name="link_plane_node3_no" minOccurs="0" type="int"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
 </complexType>
 <simpleType name="result_section_type">
 <restriction base="string">
 <enumeration value="TYPE_2_POINTS_AND_VECTOR"/>
 <enumeration value="TYPE_LINE"/>
 </restriction>
 </simpleType>
 <simpleType name="result_section_show_results_in_direction">
 <restriction base="string">
 <enumeration value="SHOW_RESULTS_IN_GLOBAL_MINUS_X"/>
 <enumeration value="SHOW_RESULTS_IN_GLOBAL_MINUS_Y"/>
 <enumeration value="SHOW_RESULTS_IN_GLOBAL_MINUS_Z"/>
 <enumeration value="SHOW_RESULTS_IN_GLOBAL_PLUS_X"/>
 <enumeration value="SHOW_RESULTS_IN_GLOBAL_PLUS_Y"/>
 </restriction>
 </simpleType>

```

```

 <enumeration value="SHOW_RESULTS_IN_GLOBAL_PLUS_Z"/>
 <enumeration value="SHOW_RESULTS_IN_LOCAL_MINUS_Z"/>
 <enumeration value="SHOW_RESULTS_IN_LOCAL_PLUS_Y"/>
 <enumeration value="SHOW_RESULTS_IN_LOCAL_PLUS_Z"/>
 </restriction>
</simpleType>
<simpleType name="result_section_projection_in_direction">
 <restriction base="string">
 <enumeration value="PROJECTION_IN_GLOBAL_X"/>
 <enumeration value="PROJECTION_IN_GLOBAL_Y"/>
 <enumeration value="PROJECTION_IN_GLOBAL_Z"/>
 <enumeration value="PROJECTION_IN_USER_DEFINED_U"/>
 <enumeration value="PROJECTION_IN_USER_DEFINED_V"/>
 <enumeration value="PROJECTION_IN_USER_DEFINED_W"/>
 <enumeration value="PROJECTION_IN_VECTOR"/>
 </restriction>
</simpleType>
<complexType name="get_result_section">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_result_section" type="xsd:get_result_section"/>
<complexType name="get_result_sectionResponse">
 <sequence>
 <element name="value" type="xsd:result_section"/>
 </sequence>
</complexType>
<element name="get_result_sectionResponse" type="xsd:get_result_sectionResponse"/>
<complexType name="set_result_section">
 <sequence>
 <element name="value" type="xsd:result_section"/>
 </sequence>
</complexType>
<element name="set_result_section" type="xsd:set_result_section"/>
<complexType name="set_result_sectionResponse">
 <sequence/>
</complexType>
<element name="set_result_sectionResponse" type="xsd:set_result_sectionResponse"/>
<complexType name="result_section">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:result_section_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="assigned_to_surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="assigned_to_solids" minOccurs="0" type="xsd:array_of_int"/>
 <element name="show_results_in_direction" minOccurs="0" type="xsd:result_section_show_results_in_direction"/>
 <element name="coordinate_system" minOccurs="0" type="int"/>
 <element name="show_values_on_isolines_enabled" minOccurs="0" type="boolean"/>
 <element name="lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="first_point" minOccurs="0" type="xsd:vector_3d"/>
 <element name="first_point_coordinate_1" minOccurs="0" type="double"/>
 <element name="first_point_coordinate_2" minOccurs="0" type="double"/>
 <element name="first_point_coordinate_3" minOccurs="0" type="double"/>
 <element name="second_point" minOccurs="0" type="xsd:vector_3d"/>
 <element name="second_point_coordinate_1" minOccurs="0" type="double"/>
 <element name="second_point_coordinate_2" minOccurs="0" type="double"/>
 <element name="second_point_coordinate_3" minOccurs="0" type="double"/>
 <element name="projection_in_direction" minOccurs="0" type="xsd:result_section_projection_in_direction"/>
 <element name="vector" minOccurs="0" type="xsd:vector_3d"/>
 <element name="vector_coordinate_1" minOccurs="0" type="double"/>
 <element name="vector_coordinate_2" minOccurs="0" type="double"/>
 <element name="vector_coordinate_3" minOccurs="0" type="double"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="modification_type">
 <restriction base="string">
 <enumeration value="DIVISION_FACTOR"/>
 <enumeration value="MULTIPLY_FACTOR"/>
 </restriction>
</simpleType>

```

```

<complexType name="structure_modification_modify_stiffnesses_material_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="material_name" minOccurs="0" type="int"/>
 <element name="modification_type" minOccurs="0" type="xsd:modification_type"/>
 <element name="E_and_G" minOccurs="0" type="double"/>
 <element name="comment" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_structure_modification_modify_stiffnesses_material_table">
 <sequence>
 <element name="structure_modification_modify_stiffnesses_material_table"
 type="xsd:structure_modification_modify_stiffnesses_material_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_material_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_material_table"/>
<complexType name="structure_modification_modify_stiffnesses_section_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="section_name" minOccurs="0" type="string"/>
 <element name="A" minOccurs="0" type="double"/>
 <element name="A_y" minOccurs="0" type="double"/>
 <element name="A_z" minOccurs="0" type="double"/>
 <element name="J" minOccurs="0" type="double"/>
 <element name="I_y" minOccurs="0" type="double"/>
 <element name="I_z" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_structure_modification_modify_stiffnesses_section_table">
 <sequence>
 <element name="structure_modification_modify_stiffnesses_section_table"
 type="xsd:structure_modification_modify_stiffnesses_section_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_section_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_section_table"/>
<complexType name="structure_modification_modify_stiffnesses_member_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="member_modification" minOccurs="0" type="int"/>
 <element name="members" minOccurs="0" type="xsd:array_of_int"/>
 <element name="comment" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_structure_modification_modify_stiffnesses_member_table">
 <sequence>
 <element name="structure_modification_modify_stiffnesses_member_table"
 type="xsd:structure_modification_modify_stiffnesses_member_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_member_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_member_table"/>
<complexType name="structure_modification_modify_stiffnesses_surface_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="surface_modification" minOccurs="0" type="int"/>
 <element name="surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="comment" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_structure_modification_modify_stiffnesses_surface_table">
 <sequence>
 <element name="structure_modification_modify_stiffnesses_surface_table"
 type="xsd:structure_modification_modify_stiffnesses_surface_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_surface_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_surface_table"/>
<complexType name="structure_modification_modify_stiffnesses_member_hinges_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="member_side" minOccurs="0" type="string"/>
 <element name="C_u_x" minOccurs="0" type="double"/>
 <element name="C_u_y" minOccurs="0" type="double"/>
 <element name="C_u_z" minOccurs="0" type="double"/>
 <element name="C_phi_x" minOccurs="0" type="double"/>
 <element name="C_phi_y" minOccurs="0" type="double"/>
 <element name="C_phi_z" minOccurs="0" type="double"/>
 </sequence>

```



```

<sequence>
 <element name="structure_modification_modify_stiffnesses_member_supports_table"
 type="xsd:structure_modification_modify_stiffnesses_member_supports_table" minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_member_supports_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_member_supports_table"/>
<complexType name="structure_modification_modify_stiffnesses_surface_supports_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="C_u_X" minOccurs="0" type="double"/>
 <element name="C_u_Y" minOccurs="0" type="double"/>
 <element name="C_u_Z" minOccurs="0" type="double"/>
 <element name="C_v_xz" minOccurs="0" type="double"/>
 <element name="C_v_yz" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_structure_modification_modify_stiffnesses_surface_supports_table">
 <sequence>
 <element name="structure_modification_modify_stiffnesses_surface_supports_table"
 type="xsd:structure_modification_modify_stiffnesses_surface_supports_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_structure_modification_modify_stiffnesses_surface_supports_table"
 type="xsd:array_of_structure_modification_modify_stiffnesses_surface_supports_table"/>
<complexType name="get_structure_modification">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_structure_modification" type="xsd:get_structure_modification"/>
<complexType name="get_structure_modificationResponse">
 <sequence>
 <element name="value" type="xsd:structure_modification"/>
 </sequence>
</complexType>
<element name="get_structure_modificationResponse" type="xsd:get_structure_modificationResponse"/>
<complexType name="set_structure_modification">
 <sequence>
 <element name="value" type="xsd:structure_modification"/>
 </sequence>
</complexType>
<element name="set_structure_modification" type="xsd:set_structure_modification"/>
<complexType name="set_structure_modificationResponse">
 <sequence/>
</complexType>
<element name="set_structure_modificationResponse" type="xsd:set_structure_modificationResponse"/>
<complexType name="structure_modification">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="assigned_to" minOccurs="0" type="string"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="modify_stiffnesses_gamma_m" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_materials" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_sections" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_members" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_surfaces" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_member_hinges" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_line_hinges" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_nodal_supports" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_line_supports" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_member_supports" minOccurs="0" type="boolean"/>
 <element name="modify_stiffnesses_surface_supports" minOccurs="0" type="boolean"/>
 <element name="modify_stiffness_member_reinforcement" minOccurs="0" type="boolean"/>
 <element name="modify_stiffness_surface_reinforcement" minOccurs="0" type="boolean"/>
 <element name="modify_stiffness_timber_members_due_moisture_class" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_material_nonlinearity_models" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_material_temperature_nonlinearities" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_line_hinges" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_member_types" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_member_hinges" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_member_nonlinearities" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_solid_types_contact_or_surfaces_contact" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_nodal_supports" minOccurs="0" type="boolean"/>
 <element name="nonlinearities_disabled_line_supports" minOccurs="0" type="boolean"/>
 </sequence>
</complexType>

```

```

<element name="nonlinearities_disabled_member_supports" minOccurs="0" type="boolean"/>
<element name="nonlinearities_disabled_surface_supports" minOccurs="0" type="boolean"/>
<element name="modify_stiffnesses_material_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_material_table"/>
<element name="modify_stiffnesses_section_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_section_table"/>
<element name="modify_stiffnesses_member_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_member_table"/>
<element name="modify_stiffnesses_surface_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_surface_table"/>
<element name="modify_stiffnesses_member_hinges_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_member_hinges_table"/>
<element name="modify_stiffnesses_line_hinges_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_line_hinges_table"/>
<element name="modify_stiffnesses_nodal_supports_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_nodal_supports_table"/>
<element name="modify_stiffnesses_line_supports_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_line_supports_table"/>
<element name="modify_stiffnesses_member_supports_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_member_supports_table"/>
<element name="modify_stiffnesses_surface_supports_table" minOccurs="0"
type="xsd:array_of_structure_modification_modify_stiffnesses_surface_supports_table"/>
<element name="deactivate_members_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_members" minOccurs="0" type="int"/>
<element name="deactivate_surfaces_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_surfaces" minOccurs="0" type="int"/>
<element name="deactivate_solids_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_solids" minOccurs="0" type="int"/>
<element name="deactivate_support_on_nodes_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_support_on_nodes" minOccurs="0" type="int"/>
<element name="deactivate_support_on_lines_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_support_on_lines" minOccurs="0" type="int"/>
<element name="deactivate_support_on_members_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_support_on_members" minOccurs="0" type="int"/>
<element name="deactivate_support_on_surfaces_enabled" minOccurs="0" type="boolean"/>
<element name="object_selection_for_deactivate_support_on_surfaces" minOccurs="0" type="int"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="cutting_pattern_pattern_orientation_category">
<restriction base="string">
<enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_ANGULAR_ROTATION"/>
<enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_COORDINATE_SYSTEM"/>
<enumeration value="INPUT_AXES_ROTATION_SPECIFICATION_TYPE_PARALLEL_TO_LINES"/>
</restriction>
</simpleType>
<simpleType name="cutting_pattern_axis">
<restriction base="string">
<enumeration value="AXIS_X"/>
<enumeration value="AXIS_Y"/>
</restriction>
</simpleType>
<complexType name="cutting_pattern_cutting_line_settings_table">
<sequence>
<element name="no" minOccurs="0" type="int"/>
<element name="line" minOccurs="0" type="int"/>
<element name="cutting_line_settings" minOccurs="0" type="int"/>
</sequence>
</complexType>
<complexType name="array_of_cutting_pattern_cutting_line_settings_table">
<sequence>
<element name="cutting_pattern_cutting_line_settings_table" type="xsd:cutting_pattern_cutting_line_settings_table" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_cutting_pattern_cutting_line_settings_table" type="xsd:array_of_cutting_pattern_cutting_line_settings_table"/>
<complexType name="get_cutting_pattern">
<sequence>
<element name="no" type="int"/>
</sequence>
</complexType>
<element name="get_cutting_pattern" type="xsd:get_cutting_pattern"/>
<complexType name="get_cutting_patternResponse">
<sequence>
<element name="value" type="xsd:cutting_pattern"/>
</sequence>
</complexType>
<element name="get_cutting_patternResponse" type="xsd:get_cutting_patternResponse"/>

```



```

<complexType name="set_cutting_pattern">
 <sequence>
 <element name="value" type="xsd:cutting_pattern"/>
 </sequence>
</complexType>
<element name="set_cutting_pattern" type="xsd:set_cutting_pattern"/>
<complexType name="set_cutting_patternResponse">
 <sequence/>
</complexType>
<element name="set_cutting_patternResponse" type="xsd:set_cutting_patternResponse"/>
<complexType name="cutting_pattern">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="material_change_enabled" minOccurs="0" type="boolean"/>
 <element name="material" minOccurs="0" type="int"/>
 <element name="pattern_orientation_category" minOccurs="0" type="xsd:cutting_pattern_pattern_orientation_category"/>
 <element name="angular_rotation" minOccurs="0" type="double"/>
 <element name="axis" minOccurs="0" type="xsd:cutting_pattern_axis"/>
 <element name="parralel_to_lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="coordinate_system" minOccurs="0" type="int"/>
 <element name="boundary_lines" minOccurs="0" type="xsd:array_of_int"/>
 <element name="cutting_line_settings_table" minOccurs="0" type="xsd:array_of_cutting_pattern_cutting_line_settings_table"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="soil_sample_type">
 <restriction base="string">
 <enumeration value="TYPE_PHANTOM"/>
 <enumeration value="TYPE_STANDARD"/>
 </restriction>
</simpleType>
<complexType name="soil_sample_layers_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="layer_no" minOccurs="0" type="int"/>
 <element name="soil_material" minOccurs="0" type="int"/>
 <element name="thickness" minOccurs="0" type="double"/>
 <element name="bottom_ordinate" minOccurs="0" type="double"/>
 </sequence>
</complexType>
<complexType name="array_of_soil_sample_layers_table">
 <sequence>
 <element name="soil_sample_layers_table" type="xsd:soil_sample_layers_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_soil_sample_layers_table" type="xsd:array_of_soil_sample_layers_table"/>
<complexType name="get_soil_sample">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_soil_sample" type="xsd:get_soil_sample"/>
<complexType name="get_soil_sampleResponse">
 <sequence>
 <element name="value" type="xsd:soil_sample"/>
 </sequence>
</complexType>
<element name="get_soil_sampleResponse" type="xsd:get_soil_sampleResponse"/>
<complexType name="set_soil_sample">
 <sequence>
 <element name="value" type="xsd:soil_sample"/>
 </sequence>
</complexType>
<element name="set_soil_sample" type="xsd:set_soil_sample"/>
<complexType name="set_soil_sampleResponse">
 <sequence/>
</complexType>
<element name="set_soil_sampleResponse" type="xsd:set_soil_sampleResponse"/>
<complexType name="soil_sample">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:soil_sample_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 </sequence>

```

```

<element name="name" minOccurs="0" type="string"/>
<element name="coordinates" minOccurs="0" type="xsd:vector_3d"/>
<element name="coordinate_0" minOccurs="0" type="double"/>
<element name="coordinate_1" minOccurs="0" type="double"/>
<element name="coordinate_2" minOccurs="0" type="double"/>
<element name="import_coordinate_z_from_terrain" minOccurs="0" type="boolean"/>
<element name="groundwater" minOccurs="0" type="boolean"/>
<element name="groundwater_ordinate" minOccurs="0" type="double"/>
<element name="layers_table" minOccurs="0" type="xsd:array_of_soil_sample_layers_table"/>
<element name="layers_thickness_sum" minOccurs="0" type="double"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="soil_massif_type">
 <restriction base="string">
 <enumeration value="TYPE_PHANTOM"/>
 <enumeration value="TYPE_STANDARD"/>
 </restriction>
</simpleType>
<simpleType name="soil_massif_assigned_to_type">
 <restriction base="string">
 <enumeration value="ASSIGNED_TO_TYPE_SOIL_SAMPLES"/>
 <enumeration value="ASSIGNED_TO_TYPE_SOIL_SOLIDS"/>
 </restriction>
</simpleType>
<simpleType name="soil_massif_topology_type">
 <restriction base="string">
 <enumeration value="TOPOLOGY_TYPE_RECTANGLE"/>
 </restriction>
</simpleType>
<simpleType name="soil_massif_model_soil_block_via">
 <restriction base="string">
 <enumeration value="MODEL_SOIL_BLOCK_VIA_FE_SOLIDS"/>
 <enumeration value="MODEL_SOIL_BLOCK_VIA_SUPPORT_SPRINGS"/>
 </restriction>
</simpleType>
<complexType name="get_soil_massif">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_soil_massif" type="xsd:get_soil_massif"/>
<complexType name="get_soil_massifResponse">
 <sequence>
 <element name="value" type="xsd:soil_massif"/>
 </sequence>
</complexType>
<element name="get_soil_massifResponse" type="xsd:get_soil_massifResponse"/>
<complexType name="set_soil_massif">
 <sequence>
 <element name="value" type="xsd:soil_massif"/>
 </sequence>
</complexType>
<element name="set_soil_massif" type="xsd:set_soil_massif"/>
<complexType name="set_soil_massifResponse">
 <sequence/>
</complexType>
<element name="set_soil_massifResponse" type="xsd:set_soil_massifResponse"/>
<complexType name="soil_massif">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:soil_massif_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="assigned_to_type" minOccurs="0" type="xsd:soil_massif_assigned_to_type"/>
 <element name="assigned_to_soil_samples" minOccurs="0" type="xsd:array_of_int"/>
 <element name="assigned_to_solids" minOccurs="0" type="xsd:array_of_int"/>
 <element name="assigned_to_solid_sets" minOccurs="0" type="xsd:array_of_int"/>
 <element name="assigned_to_solids_and_solid_sets" minOccurs="0" type="string"/>
 <element name="topology_type" minOccurs="0" type="xsd:soil_massif_topology_type"/>
 <element name="depth_according_to_soil_samples" minOccurs="0" type="boolean"/>
 <element name="diameter_for_circle_topology" minOccurs="0" type="double"/>
 <element name="polyline_for_polygon_topology" minOccurs="0" type="int"/>
 <element name="center_x" minOccurs="0" type="double"/>
 </sequence>

```

```

<element name="center_y" minOccurs="0" type="double"/>
<element name="size" minOccurs="0" type="xsd:vector_3d"/>
<element name="size_x" minOccurs="0" type="double"/>
<element name="size_y" minOccurs="0" type="double"/>
<element name="size_z" minOccurs="0" type="double"/>
<element name="rotation_about_z" minOccurs="0" type="double"/>
<element name="groundwater" minOccurs="0" type="boolean"/>
<element name="groundwater_surface" minOccurs="0" type="int"/>
<element name="model_soil_block_via" minOccurs="0" type="xsd:soil_massif_model_soil_block_via"/>
<element name="relevant_permanent_loading" minOccurs="0" type="int"/>
<element name="mesh_nodes_in_position_of_precise_layer_surfaces" minOccurs="0" type="boolean"/>
<element name="rock_beneath_last_layer" minOccurs="0" type="boolean"/>
<element name="change_mesh_refinement" minOccurs="0" type="boolean"/>
<element name="size_increasing_fe_during_growing_depth" minOccurs="0" type="double"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="nodal_support_spring_x_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_spring_y_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_spring_z_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_rotational_restraint_x_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 </restriction>
</simpleType>

```

```

<enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
<enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
<enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
<enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
<enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
<enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
<enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
<enumeration value="NONLINEARITY_TYPE_NONE"/>
<enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
<enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
</restriction>
</simpleType>
<simpleType name="nodal_support_rotational_restraint_y_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_rotational_restraint_z_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_axes_sequence">
 <restriction base="string">
 <enumeration value="SEQUENCE_XYZ"/>
 <enumeration value="SEQUENCE_XZY"/>
 <enumeration value="SEQUENCE_YXZ"/>
 <enumeration value="SEQUENCE_YZX"/>
 <enumeration value="SEQUENCE_ZXY"/>
 <enumeration value="SEQUENCE_ZYX"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_column_base_support_type">
 <restriction base="string">
 <enumeration value="BASE_SUPPORT_TYPE_HINGED"/>
 <enumeration value="BASE_SUPPORT_TYPE_RIGID"/>
 <enumeration value="BASE_SUPPORT_TYPE_SEMI_RIGID"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_column_head_support_type">
 <restriction base="string">
 <enumeration value="HEAD_SUPPORT_TYPE_HINGED"/>
 <enumeration value="HEAD_TYPE_SUPPORT_SEMI_RIGID"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_column_head_type">
 <restriction base="string">
 <enumeration value="HEAD_TYPE_CIRCULAR"/>
 <enumeration value="HEAD_TYPE_RECTANGULAR"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_column_support_type">

```

```

 <restriction base="string">
 <enumeration value="SUPPORT_TYPE_ELASTIC_NODAL_SUPPORT"/>
 <enumeration value="SUPPORT_TYPE_ELASTIC_SURFACE_FOUNDATIONS"/>
 <enumeration value="SUPPORT_TYPE_WITH_ADAPTED_FE_MESH"/>
 </restriction>
 </simpleType>
 <simpleType name="nodal_support_diagram_along_x_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <simpleType name="nodal_support_diagram_along_x_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <complexType name="nodal_support_diagram_along_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
 </complexType>
 <complexType name="array_of_nodal_support_diagram_along_x_table">
 <sequence>
 <element name="nodal_support_diagram_along_x_table" type="xsd:nodal_support_diagram_along_x_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_nodal_support_diagram_along_x_table" type="xsd:array_of_nodal_support_diagram_along_x_table"/>
 <simpleType name="nodal_support_diagram_along_y_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <simpleType name="nodal_support_diagram_along_y_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <complexType name="nodal_support_diagram_along_y_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
 </complexType>
 <complexType name="array_of_nodal_support_diagram_along_y_table">
 <sequence>
 <element name="nodal_support_diagram_along_y_table" type="xsd:nodal_support_diagram_along_y_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_nodal_support_diagram_along_y_table" type="xsd:array_of_nodal_support_diagram_along_y_table"/>
 <simpleType name="nodal_support_diagram_along_z_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>

```

```

</simpleType>
<simpleType name="nodal_support_diagram_along_z_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_diagram_along_z_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_diagram_along_z_table">
 <sequence>
 <element name="nodal_support_diagram_along_z_table" type="xsd:nodal_support_diagram_along_z_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_diagram_along_z_table" type="xsd:array_of_nodal_support_diagram_along_z_table"/>
<simpleType name="nodal_support_diagram_around_x_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_diagram_around_x_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_diagram_around_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_diagram_around_x_table">
 <sequence>
 <element name="nodal_support_diagram_around_x_table" type="xsd:nodal_support_diagram_around_x_table" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_diagram_around_x_table" type="xsd:array_of_nodal_support_diagram_around_x_table"/>
<simpleType name="nodal_support_diagram_around_y_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_diagram_around_y_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_diagram_around_y_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 </sequence>
</complexType>

```

```

 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_diagram_around_y_table">
 <sequence>
 <element name="nodal_support_diagram_around_y_table" type="xsd:nodal_support_diagram_around_y_table" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_diagram_around_y_table" type="xsd:array_of_nodal_support_diagram_around_y_table"/>
<simpleType name="nodal_support_diagram_around_z_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_diagram_around_z_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_diagram_around_z_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_diagram_around_z_table">
 <sequence>
 <element name="nodal_support_diagram_around_z_table" type="xsd:nodal_support_diagram_around_z_table" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_diagram_around_z_table" type="xsd:array_of_nodal_support_diagram_around_z_table"/>
<simpleType name="nodal_support_directed_to_node_first_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_directed_to_node_second_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_parallel_to_two_nodes_first_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_parallel_to_two_nodes_second_axis">
 <restriction base="string">
 <enumeration value="AXIS_X"/>
 <enumeration value="AXIS_Y"/>
 <enumeration value="AXIS_Z"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_partial_activity_along_x_negative_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 </restriction>

```





```

<simpleType name="nodal_support_partial_activity_around_y_positive_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_partial_activity_around_z_negative_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_partial_activity_around_z_positive_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_specific_direction_type">
 <restriction base="string">
 <enumeration value="DIRECTION_TYPE_DIRECTED_TO_NODE"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_CS_OF_LINE"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_CS_OF_MEMBER"/>
 <enumeration value="DIRECTION_TYPE_PARALLEL_TO_TWO_NODES"/>
 <enumeration value="DIRECTION_TYPE_ROTATED_VIA_3_ANGLES"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_x_depends_on">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_P"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PX"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PY"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PZ"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_x_end">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_x_start">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_stiffness_diagram_around_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_stiffness_diagram_around_x_table">
 <sequence>
 <element name="nodal_support_stiffness_diagram_around_x_table" type="xsd:nodal_support_stiffness_diagram_around_x_table"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_stiffness_diagram_around_x_table" type="xsd:array_of_nodal_support_stiffness_diagram_around_x_table"/>
<simpleType name="nodal_support_stiffness_diagram_around_y_depends_on">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_P"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PX"/>
 </restriction>

```

```

 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PY"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PZ"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_y_end">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_y_start">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_stiffness_diagram_around_y_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_stiffness_diagram_around_y_table">
 <sequence>
 <element name="nodal_support_stiffness_diagram_around_y_table" type="xsd:nodal_support_stiffness_diagram_around_y_table"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_stiffness_diagram_around_y_table" type="xsd:array_of_nodal_support_stiffness_diagram_around_y_table"/>
<simpleType name="nodal_support_stiffness_diagram_around_z_depends_on">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_P"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PX"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PY"/>
 <enumeration value="STIFFNESS_DIAGRAM_DEPENDS_ON_PZ"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_z_end">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_stiffness_diagram_around_z_start">
 <restriction base="string">
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="STIFFNESS_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="nodal_support_stiffness_diagram_around_z_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_nodal_support_stiffness_diagram_around_z_table">
 <sequence>
 <element name="nodal_support_stiffness_diagram_around_z_table" type="xsd:nodal_support_stiffness_diagram_around_z_table"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_nodal_support_stiffness_diagram_around_z_table" type="xsd:array_of_nodal_support_stiffness_diagram_around_z_table"/>
<simpleType name="nodal_support_support_dimension_type_on_x">
 <restriction base="string">
 <enumeration value="SUPPORT_DIMENSION_TYPE_CIRCULAR"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_NONE"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_RECTANGULAR"/>
 </restriction>
</simpleType>
<simpleType name="nodal_support_support_dimension_type_on_y">

```

```

 <restriction base="string">
 <enumeration value="SUPPORT_DIMENSION_TYPE_CIRCULAR"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_NONE"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_RECTANGULAR"/>
 </restriction>
 </simpleType>
 <simpleType name="nodal_support_support_dimension_type_on_z">
 <restriction base="string">
 <enumeration value="SUPPORT_DIMENSION_TYPE_CIRCULAR"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_NONE"/>
 <enumeration value="SUPPORT_DIMENSION_TYPE_RECTANGULAR"/>
 </restriction>
 </simpleType>
 <complexType name="get_nodal_support">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
 </complexType>
 <element name="get_nodal_support" type="xsd:get_nodal_support"/>
 <complexType name="get_nodal_supportResponse">
 <sequence>
 <element name="value" type="xsd:nodal_support"/>
 </sequence>
 </complexType>
 <element name="get_nodal_supportResponse" type="xsd:get_nodal_supportResponse"/>
 <complexType name="set_nodal_support">
 <sequence>
 <element name="value" type="xsd:nodal_support"/>
 </sequence>
 </complexType>
 <element name="set_nodal_support" type="xsd:set_nodal_support"/>
 <complexType name="set_nodal_supportResponse">
 <sequence/>
 </complexType>
 <element name="set_nodal_supportResponse" type="xsd:set_nodal_supportResponse"/>
 <complexType name="nodal_support">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="spring" minOccurs="0" type="xsd:vector_3d"/>
 <element name="spring_x" minOccurs="0" type="double"/>
 <element name="spring_y" minOccurs="0" type="double"/>
 <element name="spring_z" minOccurs="0" type="double"/>
 <element name="rotational_restraint" minOccurs="0" type="xsd:vector_3d"/>
 <element name="rotational_restraint_x" minOccurs="0" type="double"/>
 <element name="rotational_restraint_y" minOccurs="0" type="double"/>
 <element name="rotational_restraint_z" minOccurs="0" type="double"/>
 <element name="spring_x_nonlinearity" minOccurs="0" type="xsd:nodal_support_spring_x_nonlinearity"/>
 <element name="spring_y_nonlinearity" minOccurs="0" type="xsd:nodal_support_spring_y_nonlinearity"/>
 <element name="spring_z_nonlinearity" minOccurs="0" type="xsd:nodal_support_spring_z_nonlinearity"/>
 <element name="rotational_restraint_x_nonlinearity" minOccurs="0" type="xsd:nodal_support_rotational_restraint_x_nonlinearity"/>
 <element name="rotational_restraint_y_nonlinearity" minOccurs="0" type="xsd:nodal_support_rotational_restraint_y_nonlinearity"/>
 <element name="rotational_restraint_z_nonlinearity" minOccurs="0" type="xsd:nodal_support_rotational_restraint_z_nonlinearity"/>
 <element name="axes_sequence" minOccurs="0" type="xsd:nodal_support_axes_sequence"/>
 <element name="column_base_semi_rigid" minOccurs="0" type="double"/>
 <element name="column_base_support_type" minOccurs="0" type="xsd:nodal_support_column_base_support_type"/>
 <element name="column_head_support_type" minOccurs="0" type="xsd:nodal_support_column_head_support_type"/>
 <element name="column_head_type" minOccurs="0" type="xsd:nodal_support_column_head_type"/>
 <element name="column_height" minOccurs="0" type="double"/>
 <element name="column_material" minOccurs="0" type="int"/>
 <element name="column_rotation" minOccurs="0" type="double"/>
 <element name="column_rotational_restraint_x" minOccurs="0" type="double"/>
 <element name="column_rotational_restraint_y" minOccurs="0" type="double"/>
 <element name="column_section" minOccurs="0" type="int"/>
 <element name="column_section_same_as_head" minOccurs="0" type="boolean"/>
 <element name="column_shear_stiffness" minOccurs="0" type="boolean"/>
 <element name="column_spring_x" minOccurs="0" type="double"/>
 <element name="column_spring_y" minOccurs="0" type="double"/>
 <element name="column_spring_z" minOccurs="0" type="double"/>
 <element name="column_support_type" minOccurs="0" type="xsd:nodal_support_column_support_type"/>
 <element name="column_width_x" minOccurs="0" type="double"/>
 <element name="column_width_y" minOccurs="0" type="double"/>
 <element name="comment" minOccurs="0" type="string"/>
 </sequence>
 </complexType>

```





```

<element name="support_dimension_type_on_y" minOccurs="0" type="xsd:nodal_support_support_dimension_type_on_y"/>
<element name="support_dimension_type_on_z" minOccurs="0" type="xsd:nodal_support_support_dimension_type_on_z"/>
<element name="support_dimension_width_x" minOccurs="0" type="double"/>
<element name="support_dimension_width_y" minOccurs="0" type="double"/>
<element name="support_dimension_width_z" minOccurs="0" type="double"/>
<element name="support_dimensions_enabled" minOccurs="0" type="boolean"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="nodal_mesh_refinement_type">
 <restriction base="string">
 <enumeration value="TYPE_CIRCULAR"/>
 <enumeration value="TYPE_RECTANGULAR"/>
 </restriction>
</simpleType>
<simpleType name="nodal_mesh_refinement_circular_length_arrangement">
 <restriction base="string">
 <enumeration value="LENGTH_ARRANGEMENT_COMBINED"/>
 <enumeration value="LENGTH_ARRANGEMENT_GRADUALLY"/>
 <enumeration value="LENGTH_ARRANGEMENT_RADIAL"/>
 </restriction>
</simpleType>
<complexType name="get_nodal_mesh_refinement">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_nodal_mesh_refinement" type="xsd:get_nodal_mesh_refinement"/>
<complexType name="get_nodal_mesh_refinementResponse">
 <sequence>
 <element name="value" type="xsd:nodal_mesh_refinement"/>
 </sequence>
</complexType>
<element name="get_nodal_mesh_refinementResponse" type="xsd:get_nodal_mesh_refinementResponse"/>
<complexType name="set_nodal_mesh_refinement">
 <sequence>
 <element name="value" type="xsd:nodal_mesh_refinement"/>
 </sequence>
</complexType>
<element name="set_nodal_mesh_refinement" type="xsd:set_nodal_mesh_refinement"/>
<complexType name="set_nodal_mesh_refinementResponse">
 <sequence/>
</complexType>
<element name="set_nodal_mesh_refinementResponse" type="xsd:set_nodal_mesh_refinementResponse"/>
<complexType name="nodal_mesh_refinement">
 <sequence>
 <element name="no" type="int"/>
 <element name="type" minOccurs="0" type="xsd:nodal_mesh_refinement_type"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="nodes" minOccurs="0" type="xsd:array_of_int"/>
 <element name="circular_radius" minOccurs="0" type="double"/>
 <element name="circular_target_inner_length" minOccurs="0" type="double"/>
 <element name="circular_target_outer_length" minOccurs="0" type="double"/>
 <element name="circular_length_arrangement" minOccurs="0" type="xsd:nodal_mesh_refinement_circular_length_arrangement"/>
 <element name="rectangular_side" minOccurs="0" type="double"/>
 <element name="rectangular_target_inner_length" minOccurs="0" type="double"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="apply_only_on_selected_surfaces" minOccurs="0" type="boolean"/>
 <element name="selected_surfaces" minOccurs="0" type="xsd:array_of_int"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="line_support_coordinate_system">
 <restriction base="string">
 <enumeration value="COORDINATE_SYSTEM_GLOBAL"/>
 <enumeration value="COORDINATE_SYSTEM_LOCAL"/>
 </restriction>
</simpleType>
<simpleType name="line_support_diagram_along_x_end">

```

```

 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <simpleType name="line_support_diagram_along_x_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <complexType name="line_support_diagram_along_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
 </complexType>
 <complexType name="array_of_line_support_diagram_along_x_table">
 <sequence>
 <element name="line_support_diagram_along_x_table" type="xsd:line_support_diagram_along_x_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_line_support_diagram_along_x_table" type="xsd:array_of_line_support_diagram_along_x_table"/>
 <simpleType name="line_support_diagram_along_y_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <simpleType name="line_support_diagram_along_y_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <complexType name="line_support_diagram_along_y_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
 </complexType>
 <complexType name="array_of_line_support_diagram_along_y_table">
 <sequence>
 <element name="line_support_diagram_along_y_table" type="xsd:line_support_diagram_along_y_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
 </complexType>
 <element name="array_of_line_support_diagram_along_y_table" type="xsd:array_of_line_support_diagram_along_y_table"/>
 <simpleType name="line_support_diagram_along_z_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>
 <simpleType name="line_support_diagram_along_z_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
 </simpleType>

```

```

 </restriction>
</simpleType>
<complexType name="line_support_diagram_along_z_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="displacement" minOccurs="0" type="double"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_line_support_diagram_along_z_table">
 <sequence>
 <u>element name="line_support_diagram_along_z_table" type="xsd:line_support_diagram_along_z_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<u>element name="array_of_line_support_diagram_along_z_table" type="xsd:array_of_line_support_diagram_along_z_table"/>
<simpleType name="line_support_diagram_around_x_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_diagram_around_x_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="line_support_diagram_around_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_line_support_diagram_around_x_table">
 <sequence>
 <u>element name="line_support_diagram_around_x_table" type="xsd:line_support_diagram_around_x_table" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<u>element name="array_of_line_support_diagram_around_x_table" type="xsd:array_of_line_support_diagram_around_x_table"/>
<simpleType name="line_support_diagram_around_y_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_diagram_around_y_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="line_support_diagram_around_y_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_line_support_diagram_around_y_table">
 <sequence>

```



```

<element name="line_support_diagram_around_y_table" type="xsd:line_support_diagram_around_y_table" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
</complexType>
<element name="array_of_line_support_diagram_around_y_table" type="xsd:array_of_line_support_diagram_around_y_table"/>
<simpleType name="line_support_diagram_around_z_end">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_diagram_around_z_start">
 <restriction base="string">
 <enumeration value="DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_STOP"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<complexType name="line_support_diagram_around_z_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="rotation" minOccurs="0" type="double"/>
 <element name="moment" minOccurs="0" type="double"/>
 <element name="spring" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_line_support_diagram_around_z_table">
 <sequence>
 <element name="line_support_diagram_around_z_table" type="xsd:line_support_diagram_around_z_table" minOccurs="0"
maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_line_support_diagram_around_z_table" type="xsd:array_of_line_support_diagram_around_z_table"/>
<simpleType name="line_support_partial_activity_along_x_negative_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_partial_activity_along_x_positive_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_partial_activity_along_y_negative_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_partial_activity_along_y_positive_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FIXED"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_TEARING"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_support_partial_activity_along_z_negative_type">
 <restriction base="string">
 <enumeration value="PARTIAL_ACTIVITY_TYPE_COMPLETE"/>
 <enumeration value="PARTIAL_ACTIVITY_TYPE_FAILURE"/>
 </restriction>
</simpleType>

```





```

 </restriction>
 </simpleType>
 <simpleType name="line_support_spring_z_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
 </simpleType>
 <simpleType name="line_support_fictitious_wall_head_support_type">
 <restriction base="string">
 <enumeration value="HEAD_SUPPORT_TYPE_HINGED"/>
 <enumeration value="HEAD_TYPE_SUPPORT_SEMI_RIGID"/>
 </restriction>
 </simpleType>
 <simpleType name="line_support_fictitious_wall_base_support_type">
 <restriction base="string">
 <enumeration value="BASE_SUPPORT_TYPE_ELASTIC"/>
 <enumeration value="BASE_SUPPORT_TYPE_HINGED"/>
 <enumeration value="BASE_SUPPORT_TYPE_RIGID"/>
 </restriction>
 </simpleType>
 <complexType name="get_line_support">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
 </complexType>
 <element name="get_line_support" type="xsd:get_line_support"/>
 <complexType name="get_line_supportResponse">
 <sequence>
 <element name="value" type="xsd:line_support"/>
 </sequence>
 </complexType>
 <element name="get_line_supportResponse" type="xsd:get_line_supportResponse"/>
 <complexType name="set_line_support">
 <sequence>
 <element name="value" type="xsd:line_support"/>
 </sequence>
 </complexType>
 <element name="set_line_support" type="xsd:set_line_support"/>
 <complexType name="set_line_supportResponse">
 <sequence/>
 </complexType>
 <element name="set_line_supportResponse" type="xsd:set_line_supportResponse"/>
 <complexType name="line_support">
 <sequence>
 <element name="no" type="int"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="coordinate_system" minOccurs="0" type="xsd:line_support_coordinate_system"/>
 <element name="diagram_along_x_end" minOccurs="0" type="xsd:line_support_diagram_along_x_end"/>
 <element name="diagram_along_x_is_sorted" minOccurs="0" type="boolean"/>
 <element name="diagram_along_x_start" minOccurs="0" type="xsd:line_support_diagram_along_x_start"/>
 <element name="diagram_along_x_symmetric" minOccurs="0" type="boolean"/>
 <element name="diagram_along_x_table" minOccurs="0" type="xsd:array_of_line_support_diagram_along_x_table"/>
 <element name="diagram_along_y_end" minOccurs="0" type="xsd:line_support_diagram_along_y_end"/>
 <element name="diagram_along_y_is_sorted" minOccurs="0" type="boolean"/>
 <element name="diagram_along_y_start" minOccurs="0" type="xsd:line_support_diagram_along_y_start"/>
 <element name="diagram_along_y_symmetric" minOccurs="0" type="boolean"/>
 <element name="diagram_along_y_table" minOccurs="0" type="xsd:array_of_line_support_diagram_along_y_table"/>
 <element name="diagram_along_z_end" minOccurs="0" type="xsd:line_support_diagram_along_z_end"/>
 <element name="diagram_along_z_is_sorted" minOccurs="0" type="boolean"/>
 <element name="diagram_along_z_start" minOccurs="0" type="xsd:line_support_diagram_along_z_start"/>
 <element name="diagram_along_z_symmetric" minOccurs="0" type="boolean"/>
 <element name="diagram_along_z_table" minOccurs="0" type="xsd:array_of_line_support_diagram_along_z_table"/>
 <element name="diagram_around_x_end" minOccurs="0" type="xsd:line_support_diagram_around_x_end"/>
 <element name="diagram_around_x_is_sorted" minOccurs="0" type="boolean"/>
 </sequence>
 </complexType>

```



```

<element name="partial_activity_around_z_positive_moment" minOccurs="0" type="double"/>
<element name="partial_activity_around_z_positive_rotation" minOccurs="0" type="double"/>
<element name="partial_activity_around_z_positive_slippage" minOccurs="0" type="double"/>
<element name="partial_activity_around_z_positive_type" minOccurs="0"
type="xsd:line_support_partial_activity_around_z_positive_type"/>
<element name="rotational_restraint" minOccurs="0" type="xsd:vector_3d"/>
<element name="rotational_restraint_x" minOccurs="0" type="double"/>
<element name="rotational_restraint_x_nonlinearity" minOccurs="0" type="xsd:line_support_rotational_restraint_x_nonlinearity"/>
<element name="rotational_restraint_y" minOccurs="0" type="double"/>
<element name="rotational_restraint_y_nonlinearity" minOccurs="0" type="xsd:line_support_rotational_restraint_y_nonlinearity"/>
<element name="rotational_restraint_z" minOccurs="0" type="double"/>
<element name="rotational_restraint_z_nonlinearity" minOccurs="0" type="xsd:line_support_rotational_restraint_z_nonlinearity"/>
<element name="spring" minOccurs="0" type="xsd:vector_3d"/>
<element name="spring_x" minOccurs="0" type="double"/>
<element name="spring_x_nonlinearity" minOccurs="0" type="xsd:line_support_spring_x_nonlinearity"/>
<element name="spring_y" minOccurs="0" type="double"/>
<element name="spring_y_nonlinearity" minOccurs="0" type="xsd:line_support_spring_y_nonlinearity"/>
<element name="spring_z" minOccurs="0" type="double"/>
<element name="spring_z_nonlinearity" minOccurs="0" type="xsd:line_support_spring_z_nonlinearity"/>
<element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
<element name="x_axis_rotation" minOccurs="0" type="double"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<element name="generating_object_info" minOccurs="0" type="string"/>
<element name="fictitious_wall_enabled" minOccurs="0" type="boolean"/>
<element name="fictitious_wall_width" minOccurs="0" type="double"/>
<element name="fictitious_wall_height" minOccurs="0" type="double"/>
<element name="fictitious_wall_head_support_type" minOccurs="0" type="xsd:line_support_fictitious_wall_head_support_type"/>
<element name="fictitious_wall_base_support_type" minOccurs="0" type="xsd:line_support_fictitious_wall_base_support_type"/>
<element name="fictitious_wall_base_elastic" minOccurs="0" type="double"/>
<element name="fictitious_wall_shear_stiffness" minOccurs="0" type="boolean"/>
<element name="fictitious_wall_material" minOccurs="0" type="int"/>
<element name="fictitious_wall_spring_x" minOccurs="0" type="double"/>
<element name="fictitious_wall_spring_y" minOccurs="0" type="double"/>
<element name="fictitious_wall_spring_z" minOccurs="0" type="double"/>
<element name="fictitious_wall_rotational_restraint_about_line_axis" minOccurs="0" type="double"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="line_mesh_refinement_type">
<restriction base="string">
<enumeration value="TYPE_ELEMENTS"/>
<enumeration value="TYPE_GRADUAL"/>
<enumeration value="TYPE_LENGTH"/>
</restriction>
</simpleType>
<complexType name="get_line_mesh_refinement">
<sequence>
<element name="no" type="int"/>
</sequence>
</complexType>
<element name="get_line_mesh_refinement" type="xsd:get_line_mesh_refinement"/>
<complexType name="get_line_mesh_refinementResponse">
<sequence>
<element name="value" type="xsd:line_mesh_refinement"/>
</sequence>
</complexType>
<element name="get_line_mesh_refinementResponse" type="xsd:get_line_mesh_refinementResponse"/>
<complexType name="set_line_mesh_refinement">
<sequence>
<element name="value" type="xsd:line_mesh_refinement"/>
</sequence>
</complexType>
<element name="set_line_mesh_refinement" type="xsd:set_line_mesh_refinement"/>
<complexType name="set_line_mesh_refinementResponse">
<sequence/>
</complexType>
<element name="set_line_mesh_refinementResponse" type="xsd:set_line_mesh_refinementResponse"/>
<complexType name="line_mesh_refinement">
<sequence>
<element name="no" type="int"/>
<element name="type" minOccurs="0" type="xsd:line_mesh_refinement_type"/>
<element name="comment" minOccurs="0" type="string"/>
<element name="elements_finite_elements" minOccurs="0" type="int"/>
<element name="generating_object_info" minOccurs="0" type="string"/>

```

```

<element name="gradual_rows" minOccurs="0" type="int"/>
<element name="is_generated" minOccurs="0" type="boolean"/>
<u>element name="lines" minOccurs="0" type="xsd:array_of_int"/>
<element name="name" minOccurs="0" type="string"/>
<element name="number_of_layers" minOccurs="0" type="int"/>
<element name="target_length" minOccurs="0" type="double"/>
<element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
<element name="id_for_export_import" minOccurs="0" type="string"/>
<element name="metadata_for_export_import" minOccurs="0" type="string"/>
</sequence>
</complexType>
<simpleType name="line_hinge_translational_release_u_x_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="line_hinge_translational_release_u_y_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="line_hinge_translational_release_u_z_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<simpleType name="line_hinge_rotational_release_phi_x_nonlinearity">
 <restriction base="string">
 <enumeration value="NONLINEARITY_TYPE_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_ALL_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_NEGATIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FAILURE_IF_POSITIVE"/>
 <enumeration value="NONLINEARITY_TYPE_FORCE_MOMENT_DIAGRAM"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_1_PLUS_2"/>
 <enumeration value="NONLINEARITY_TYPE_FRICTION_DIRECTION_2"/>
 <enumeration value="NONLINEARITY_TYPE_NONE"/>
 </restriction>

```

```

 <enumeration value="NONLINEARITY_TYPE_PARTIAL_ACTIVITY"/>
 <enumeration value="NONLINEARITY_TYPE_STIFFNESS_DIAGRAM"/>
 </restriction>
</simpleType>
<complexType name="line_hinge_force_moment_diagram_around_x_table">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="force" minOccurs="0" type="double"/>
 <element name="max_moment" minOccurs="0" type="double"/>
 <element name="min_moment" minOccurs="0" type="double"/>
 <element name="note" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<complexType name="array_of_line_hinge_force_moment_diagram_around_x_table">
 <sequence>
 <element name="line_hinge_force_moment_diagram_around_x_table" type="xsd:line_hinge_force_moment_diagram_around_x_table"
 minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_line_hinge_force_moment_diagram_around_x_table"
 type="xsd:array_of_line_hinge_force_moment_diagram_around_x_table"/>
<simpleType name="line_hinge_force_moment_diagram_around_x_start">
 <restriction base="string">
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_hinge_force_moment_diagram_around_x_end">
 <restriction base="string">
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_CONTINUOUS"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_TEARING"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_ENDING_TYPE_YIELDING"/>
 </restriction>
</simpleType>
<simpleType name="line_hinge_force_moment_diagram_around_x_depends_on">
 <restriction base="string">
 <enumeration value="FORCE_MOMENT_DIAGRAM_DEPENDS_ON_N"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_DEPENDS_ON_VY"/>
 <enumeration value="FORCE_MOMENT_DIAGRAM_DEPENDS_ON_VZ"/>
 </restriction>
</simpleType>
<complexType name="line_hinge_generated_line_hinges">
 <sequence>
 <element name="no" minOccurs="0" type="int"/>
 <element name="generated_by" minOccurs="0" type="int"/>
 <element name="generated_line_hinge" minOccurs="0" type="int"/>
 </sequence>
</complexType>
<complexType name="array_of_line_hinge_generated_line_hinges">
 <sequence>
 <element name="line_hinge_generated_line_hinges" type="xsd:line_hinge_generated_line_hinges" minOccurs="0"
 maxOccurs="unbounded"/>
 </sequence>
</complexType>
<element name="array_of_line_hinge_generated_line_hinges" type="xsd:array_of_line_hinge_generated_line_hinges"/>
<complexType name="get_line_hinge">
 <sequence>
 <element name="no" type="int"/>
 </sequence>
</complexType>
<element name="get_line_hinge" type="xsd:get_line_hinge"/>
<complexType name="get_line_hingeResponse">
 <sequence>
 <element name="value" type="xsd:line_hinge"/>
 </sequence>
</complexType>
<element name="get_line_hingeResponse" type="xsd:get_line_hingeResponse"/>
<complexType name="set_line_hinge">
 <sequence>
 <element name="value" type="xsd:line_hinge"/>
 </sequence>
</complexType>
<element name="set_line_hinge" type="xsd:set_line_hinge"/>
<complexType name="set_line_hingeResponse">
 <sequence/>
</complexType>
<element name="set_line_hingeResponse" type="xsd:set_line_hingeResponse"/>

```



```

<complexType name="line_hinge">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="assigned_to" minOccurs="0" type="string"/>
 <element name="translational_release_u_x" minOccurs="0" type="double"/>
 <element name="translational_release_u_x_nonlinearity" minOccurs="0" type="xsd:line_hinge_translational_release_u_x_nonlinearity"/>
 <element name="translational_release_u_y" minOccurs="0" type="double"/>
 <element name="translational_release_u_y_nonlinearity" minOccurs="0" type="xsd:line_hinge_translational_release_u_y_nonlinearity"/>
 <element name="translational_release_u_z" minOccurs="0" type="double"/>
 <element name="translational_release_u_z_nonlinearity" minOccurs="0" type="xsd:line_hinge_translational_release_u_z_nonlinearity"/>
 <element name="rotational_release_phi_x" minOccurs="0" type="double"/>
 <element name="rotational_release_phi_x_nonlinearity" minOccurs="0" type="xsd:line_hinge_rotational_release_phi_x_nonlinearity"/>
 <element name="force_moment_diagram_around_x_table" minOccurs="0" type="xsd:array_of_line_hinge_force_moment_diagram_around_x_table"/>
 <element name="force_moment_diagram_around_x_symmetric" minOccurs="0" type="boolean"/>
 <element name="force_moment_diagram_around_x_is_sorted" minOccurs="0" type="boolean"/>
 <element name="force_moment_diagram_around_x_start" minOccurs="0" type="xsd:line_hinge_force_moment_diagram_around_x_start"/>
 <element name="force_moment_diagram_around_x_end" minOccurs="0" type="xsd:line_hinge_force_moment_diagram_around_x_end"/>
 <element name="force_moment_diagram_around_x_depends_on" minOccurs="0" type="xsd:line_hinge_force_moment_diagram_around_x_depends_on"/>
 <element name="slab_wall_connection" minOccurs="0" type="boolean"/>
 <element name="slab_wall_with_slab_edge_block" minOccurs="0" type="boolean"/>
 <element name="slab_wall_connection_offset" minOccurs="0" type="double"/>
 <element name="slab_edge_block_width" minOccurs="0" type="double"/>
 <element name="generated_line_hinges" minOccurs="0" type="xsd:array_of_line_hinge_generated_line_hinges"/>
 <element name="comment" minOccurs="0" type="string"/>
 <element name="is_generated" minOccurs="0" type="boolean"/>
 <element name="generating_object_info" minOccurs="0" type="string"/>
 <element name="id_for_export_import" minOccurs="0" type="string"/>
 <element name="metadata_for_export_import" minOccurs="0" type="string"/>
 </sequence>
</complexType>
<simpleType name="line_welded_joint_joint_type">
 <restriction base="string">
 <enumeration value="BUTT_JOINT"/>
 <enumeration value="CORNER_JOINT"/>
 <enumeration value="LAP_JOINT"/>
 <enumeration value="TEE_JOINT"/>
 </restriction>
</simpleType>
<simpleType name="line_welded_joint_weld_type">
 <restriction base="string">
 <enumeration value="WELD_BEVEL_AND_FILLET"/>
 <enumeration value="WELD_DOUBLE_BEVEL"/>
 <enumeration value="WELD_DOUBLE_FILLET"/>
 <enumeration value="WELD_DOUBLE_J"/>
 <enumeration value="WELD_DOUBLE_SQUARE"/>
 <enumeration value="WELD_DOUBLE_U"/>
 <enumeration value="WELD_DOUBLE_V"/>
 <enumeration value="WELD_J_AND_FILLET"/>
 <enumeration value="WELD_SINGLE_BEVEL"/>
 <enumeration value="WELD_SINGLE_FILLET"/>
 <enumeration value="WELD_SINGLE_J"/>
 <enumeration value="WELD_SINGLE_SQUARE"/>
 <enumeration value="WELD_SINGLE_U"/>
 <enumeration value="WELD_SINGLE_V"/>
 <enumeration value="WELD_V_AND_FILLET"/>
 </restriction>
</simpleType>
<simpleType name="line_welded_joint_longitudinal_arrangement">
 <restriction base="string">
 <enumeration value="CHAIN_INTERMITTENT"/>
 <enumeration value="CONTINUOUS"/>
 <enumeration value="STAGGERED_INTERMITTENT"/>
 </restriction>
</simpleType>
<complexType name="get_line_welded_joint">
 <sequence>
 <element name="no" type="int"/>
 </sequence>

```

```
</complexType>
<element name="get_line_welded_joint" type="xsd:get_line_welded_joint"/>
<complexType name="get_line_welded_jointResponse">
 <sequence>
 <element name="value" type="xsd:line_welded_joint"/>
 </sequence>
</complexType>
<element name="get_line_welded_jointResponse" type="xsd:get_line_welded_jointResponse"/>
<complexType name="set_line_welded_joint">
 <sequence>
 <element name="value" type="xsd:line_welded_joint"/>
 </sequence>
</complexType>
<element name="set_line_welded_joint" type="xsd:set_line_welded_joint"/>
<complexType name="set_line_welded_jointResponse">
 <sequence/>
</complexType>
<element name="set_line_welded_jointResponse" type="xsd:set_line_welded_jointResponse"/>
<complexType name="line_welded_joint">
 <sequence>
 <element name="no" type="int"/>
 <element name="user_defined_name_enabled" minOccurs="0" type="boolean"/>
 <element name="name" minOccurs="0" type="string"/>
 <element name="joint_type" minOccurs="0" type="xsd:line_welded_joint_joint_type"/>
 <element name="weld_type" minOccurs="0" type="xsd:line_welded_joint_weld_type"/>
 <element name="longitudinal_arrangement" minOccurs="0" type="xsd:line_welded_joint_longitudinal_arrangement"/>
 <element name="weld_size_a1" minOccurs="0" type="double"/>
 <element name="weld_size_a2" minOccurs="0" type="double"/>
 <element name="weld_length"/>
 </sequence>
</complexType>
```