

Api description

Bimplus WebSDK version : 0.0.98

Table of Contents

- [calculateCutSection](#)
- [api](#)
 - [applicationSettings](#)
 - [postSettings](#)
 - [getSettings](#)
 - [putSettings](#)
 - [deleteSettings](#)
 - [attachments](#)
 - [get](#)
 - [download](#)
 - [put](#)
 - [post](#)
 - [delete](#)
 - [postVersion](#)
 - [getAllVersions](#)
 - [attributeTemplates](#)
 - [post](#)
 - [get](#)
 - [put](#)
 - [delete](#)
 - [authorize \(obsolete\)](#)
 - [post](#)
 - [forgotPassword](#)
 - [checkAccessToken](#)
 - [requestCrossToken](#)
 - [clashDetection](#)
 - [detectAllClashes](#)
 - [detectDisciplinesClashes](#)
 - [getFilters](#)
 - [calculateClashes](#)
 - [getFilterResults](#)
 - [createClashRule](#)
 - [deleteClashRule](#)
 - [getAllClashRules](#)
 - [comments](#)
 - [get](#)
 - [delete](#)
 - [put](#)
 - [customTables](#)
 - [get](#)
 - [post](#)
 - [dataProtection](#)
 - [get](#)
 - [elementTypes](#)
 - [get](#)
 - [getDisciplines](#)
 - [getAttributeTemplates](#)
 - [getTopology](#)
 - [getProjectTopology](#)
 - [exportService](#)
 - [exportObjects](#)
 - [exportModel](#)
 - [freeAttributeGroups](#)
 - [_getUrl](#)
 - [post](#)
 - [get](#)
 - [put](#)
 - [delete](#)
 - [getFreeAttributes](#)
 - [freeAttributes](#)
 - [_getUrl](#)
 - [post](#)
 - [get](#)
 - [put](#)
 - [delete](#)
 - [getTopology](#)
 - [getFilteredAttributes](#)
 - [groupActionsRights](#)
 - [get](#)
 - [hiddenCalculation](#)

- calculate
- hyperlinks
 - get
 - delete
 - put
- importService
 - postModel
 - postModelAndXML
- imprints
 - get
- invitations
 - get
 - getProjectInvitations
 - post
 - delete
 - put
 - accept
 - getInvitation
- issues
 - post
 - get
 - postPin
 - getPins
 - deletePins
 - postComment
 - getComments
 - deleteComments
 - postAttachment
 - getAttachments
 - deleteAttachments
 - put
 - delete
 - getRights
 - getIssueTypes
 - importBcf
 - updateImportBcf
 - exportBcf
 - exportExcel
- locks
 - lockObject
 - checkLocked
 - deleteLock
- members
 - post
 - get
 - putUser
 - deleteUser
- messages
 - post
 - getUserMessages
 - getUserDashboardMessages
 - getSentUserMessages
 - getSentGroupMessages
 - get
 - getFiltered
 - deleteMessage
 - markAsRead
 - updateDashboardAdditionalData
 - addDashboardMessageAttachment
 - getMessagingGroupsDetails
 - getMessagingGroupsDetails
- models
 - get
 - head
 - getDisciplines
 - getRevisions
 - postRevision
 - put
 - download
 - delete
 - move
- objectFilters
 - get
 - getFilteredObjects
 - applyFilter
 - post
 - put
 - delete
 - getFilter

objects

- post
- get
- getInternal
- head
- getTopology
- getGeometryThreeJS
- getGeometryTemplatesThreeJS
- getGeometryCompressed
- getProperties
- put
- postComment
- getComments
- deleteComments
- postAttachment
- getAttachments
- deleteAttachments
- postHyperlink
- getHyperlinks
- deleteHyperlinks
- delete
- getAttributes
- getAttributeValues
- getComplexPropertiesTopology
- postThumbnail
- addChildObject
- getReports

pins

- get
- delete
- put

projectActionsRights

- get

projectAttributeTemplates

- post
- get
- put
- delete
- copyFrom

projects

- post
- put
- get
- delete
- getModels
- getModelTypes
- getFreeAttrModelTypes
- postModel
- getMembers
- postMember
- putMember
- deleteMember
- getPins
- putProject
- postAttachment
- getAttachments
- deleteAttachments
- getTopology
- getSpots
- postThumbnail
- getThumbnail
- getComments
- getHyperlinks
- getIssues
- getIssuesShortInfo
- postSlideshow
- getSlideshows
- deleteSlideshows
- getProjectInfo

revisions

- postRevision
- getRevisionList
- getRevisionInfo
- putRevisionInfo
- getRevisionsComparisonResult
- compareRevisions

rightsService

- getRoles
- postUserRole

- putUserRole
 - deleteUser
- roles
 - get
 - getAllRights
 - deleteRole
 - deleteRole
 - getRoleDetails
 - updateRole
- sectionCut
- selections
 - post
 - get
 - getAllSelections
 - delete
- share
 - post
- slides
 - get
 - put
 - delete
 - postThumbnail
- slideshows
 - get
 - put
 - delete
 - postSlide
 - getSlides
 - postThumbnail
 - getThumbnail
- structures
 - postStructure
 - postStructureTopology
 - postSubNode
 - postAssignedObject
 - attributeAssign
 - deleteAssignedObject
 - getStructure
 - getTopology
 - getTopologyStructure
 - resetTopologyStructure
 - copyStructure
 - putStructure
 - reorderStructure
 - deleteStructure
 - deleteAllProjectStructures
 - postVariantType
 - getVariantType
 - deleteVariantType
 - copyElementsToVariant
- teamActionsRights
 - get
- teams
 - get
 - postTeam
 - putTeam
 - deleteTeam
 - getAccountSettings
 - getRestrictions
- terms
 - get
- units
 - post
 - get
 - put
 - delete
- user
 - get
 - put
 - post
 - postPhoto
 - getPhoto
 - putPassword
 - getSubscriptions
- userGroups
 - postGroup
 - getGroups
 - getUsers
 - deleteGroup

- putGroup
- assignUser
- users
 - post
 - exist
 - put
 - delete
 - setSubscriptions
- userSettings
 - postSettings
 - getSettings

calculateCutSection

calculate cut

Parameters

- projectId **string** project id
- cuttingPoint **v3**
- cuttingDirection **v3**
- async **boolean** run async flag (optional, default false)

Returns **promise** promise

api

Parameters

- config **json** api config

applicationSettings

postSettings

post settings

Parameters

- data **json** settins data

Returns **promise** promise

getSettings

get settings

Parameters

- settingId **string** settings id - if not specified return all settings

Returns **promise** promise

putSettings

put settings

Parameters

- settingId **string** settings id
- data **json** settins data

Returns **promise** promise

deleteSettings

delete settings

Parameters

- `settingId` **string** settings id

Returns **promise** promise

attachments

get

get attachment

Parameters

- `id` **string** attachment id

Returns **promise** promise

download

download attachment

Parameters

- `id` **string** attachment id
- `version` **number** version - can be empty

Returns **promise** promise

put

put attachment

Parameters

- `id` **string** attachment id
- `data` **json** data

Returns **promise** promise

post

post attachment

Parameters

- `id` **[type]** [description]
- `data` **[type]** [description]
- `uploadProgress`

Returns **promise** promise

delete

delete attachment

Parameters

- id **string** attachment id
- version **number** version - can be empty

Returns **promise** promise

postVersion

post new version of the attachment

Parameters

- id **[type]** [description]
- data **[type]** [description]
- uploadProgress

Returns **promise** promise

getAllVersions

get all attachment versions

Parameters

- id **string** attachment id

Returns **promise** promise

attributeTemplates

post

post attribute template

Parameters

- data **json** template data

Returns **promise** promise

get

get attribute template - if id is not specified return all attribute templates

Parameters

- id **string** template id

Returns **promise** promise

put

put attribute template

Parameters

- id **string** template id
- data **json** template data

Returns **promise** promise

delete

delete attribute template

Parameters

- id **string** template id

Returns **promise** promise

authorize (**obsolete**)

post

post authorization

Parameters

- username **string** user name
- password **password** password
- applicationId **string** application id - Please contact Allplan company to obtain application id !

Returns **promise** promise

forgotPassword

forgot password

Parameters

- data **json** data

Returns **promise** promise

checkAccessToken

check access token validity

Returns **promise** promise

requestCrossToken

request cross token

Returns **promise** promise

clashDetection

detectAllClashes

detect all clashes

Parameters

- data **json** calculation data

Returns **promise** promise

detectDisciplinesClashes

detect disciplines clashes

Parameters

- data **json** calculation data

Returns **promise** promise

getFilters

get clash filters

Parameters

- filterId **string** filter id - if not specified return all filters

Returns **promise** promise

calculateClashes

calculate clashes

Parameters

- data **json** calculation data
- topology

Returns **promise** promise

getFilterResults

get filter results

Parameters

- projectId **string** project id
- filterId **string** filter id
- topology

Returns **promise** promise

createClashRule

createClashRule

Parameters

- projectId **string** project id
- data **JSON** filter definition data

Returns **promise** promise

deleteClashRule

deleteClashRule

Parameters

- ruleId **string**

Returns **promise** promise

getAllClashRules

get all clash filters (new method)

Parameters

- `projectId` **string** project id

Returns **promise** promise

comments

get

get comments

Parameters

- `id` **string** comments id

Returns **promise** promise

delete

delete comment

Parameters

- `id` **string** comments id

Returns **promise** promise

put

put comment

Parameters

- `id` **string** comments id
- `data` **json** comment data

Returns **promise** promise

Returns **[type]** [description]

customTables

get

get custom tables

Parameters

- `objectId` **string** object id
- `variantTypeId` **string?** variant type id
- `revision` **nubmer** revision number

Returns **promise** promise

post

post custom table

Parameters

- objectId **string** object id
- data **json** data

Returns **promise** promise

dataProtection

get

get data protection

Parameters

- language **string** data protection language

Returns **promise** promise

elementTypes

get

get element types - if id is not specified it returns all element types

Parameters

- id **string** element type id

Returns **promise** promise

getDisciplines

get element type disciplines - if id is not specified it returns all disciplines for element types

Parameters

- id **string** discipline id

Returns **promise** promise

getAttributeTemplates

get attribute templates

Parameters

- id **string** element type id
- queryParams **json** query params e.g. { attributes : true, projectAttributeTemplate: projectAttributeTemplateId }

Returns **promise** promise

getTopology

get element types topology

Returns **promise** promise

getProjectTopology

get element types for project

Parameters

- `projectId` **string** project id

Returns **promise** promise

exportService

exportObjects

export objects

Parameters

- `projectId` **string** project id
- `elementTypeId` **string** element type id

Returns **promise** promise

exportModel

export model revision

Parameters

- `projectId` **string** project id
- `modelId` **string** model id
- `revisionNr` **string** revision number, if not defined, last revision is exported

Returns **promise** promise

freeAttributeGroups

_getUrl

private helper function to get correct url based on content property

Parameters

- `content` **boolean** flag if it is content attribute group

Returns **promise** promise

post

post free attribute group

Parameters

- `data` **json** group data
- `content` **boolean** flag if it is content attribute group

Returns **promise** promise

get

get free attribute group

Parameters

- freeAttributeGroupId **string** free attribute group id
- content **boolean** flag if it is content attribute group

Returns **promise** promise

put

put free attribute group

Parameters

- freeAttributeGroupId **string** free attribute group id
- data **json** group data
- content **boolean** flag if it is content attribute group

Returns **promise** promise

delete

delete free attribute group

Parameters

- freeAttributeGroupId **string** free attribute group id
- content **boolean** flag if it is content attribute group

Returns **promise** promise

getFreeAttributes

get free attribute group

Parameters

- freeAttributeGroupId **string** free attribute group id
- content **boolean** flag if it is content attribute group

Returns **promise** promise

freeAttributes

_getUrl

private helper function to get correct url based on content property

Parameters

- content **boolean** flag if it is content attribute

Returns **promise** promise

post

post free attribute

Parameters

- data **json** attribute data
- content **boolean** flag if it is content attribute

Returns **promise** promise

get

get free attribute - if freeAttributeId is not specified return all

Parameters

- freeAttributeId **string** attribute id
- content **boolean** flag if it is content attribute
- queryParams **string** filter query

Returns **promise** promise

put

put free attribute

Parameters

- freeAttributeId **string** attribute id
- data **json** attribute data
- content **boolean** flag if it is content attribute

Returns **promise** promise

delete

delete free attribute

Parameters

- freeAttributeId **string** attribute id
- content **boolean** flag if it is content attribute

Returns **promise** promise

getTopology

get free attribute topology

Parameters

- content **boolean** flag if it is content attribute

Returns **promise** promise

getFilteredAttributes

get filtered attributes

Parameters

- projectId **string** project id
- layerId **string** for filtering attributes from requested discipline
- type **string** for filtering attributes from special objects like 'Tasks', 'Attachments', 'Structures'

Returns **promise** promise

groupActionsRights

get

get group actions rights

Returns **promise** promise

hiddenCalculation

calculate

calculate hidden

Parameters

- projectId **string** project id
- view **json** view params
 - view.cameraViewPoint **string** e.g. { X : camPos.x , Y: camPos.y, Z: camPos.z},
 - view.cameraDirection **string** e.g. { X : camDir.x , Y: camDir.y, Z: camDir.z}
 - view.cameraUpVector **string** e.g. { X : 0, Y : 1, Z: 0},
 - view.fieldOfView **number** field of view
- clippingPlanes
- filename **string** file name
- async **boolean** run async flag

Returns **promise** promise

hyperlinks

get

get hyperlink

Parameters

- id **string** hyperlink id

Returns **promise** promise

delete

delete hyperlink

Parameters

- id **string** hyperlink id

Returns **promise** promise

put

put hyperlink

Parameters

- id **string** hyperlink id
- data **json** string

Returns **promise** promise

importService

postModel

post model

Parameters

- projectId **string** project id
- modelId **string** model id
- data **json** data
- progressCallback

Returns **promise** promise

postModelLandXML

post model - land XML

Parameters

- projectId **string** project id
- modelId **string** model id
- data **json** data
- progressCallback

Returns **promise** promise

imprints

get

get data protection

Parameters

- language **string** imprints language

Returns **promise** promise

invitations

get

get invitation

Parameters

- id **string** invitation id, if id is null return all invitations

Returns **promise** promise

getProjectInvitations

get all project invitation

Parameters

- prjId
- id **string** invitation id, if id is null return all invitations

Returns **promise** promise

post

post invitation

Parameters

- data **json** stringified json data

Returns **promise** promise

delete

delete invitation

Parameters

- id **string** invitation id

Returns **promise** promise

put

put invitation

Parameters

- id **string** invitation id
- data **json** string

Returns **promise** promise

accept

accept invitation

Parameters

- id **string** invitation id
- data **json** string

Returns **promise** promise

getInvitation

get invitation info

Parameters

- id **string** invitation id
- data **json** string

Returns **promise** promise

issues

post

post issue

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** promise

get

get issue

Parameters

- id **string** issue id

Returns **promise** promise

postPin

post issue's pin

Parameters

- id **string** issue id
- data **json** data

Returns **promise** promise

getPins

get issue's pins

Parameters

- id **string** issue id

Returns **promise** promise

deletePins

delete issues pins

Parameters

- id **string** issue id

Returns **promise** promise

postComment

post issue's comment

Parameters

- id **string** issue id
- data **json** data

Returns **promise** promise

getComments

get issues comments

Parameters

- id **string** issue id

Returns [promise](#) promise

deleteComments

delete issue's comments

Parameters

- id [string](#) issue id

Returns [promise](#) promise

postAttachment

post issue's attachment

Parameters

- id [string](#) issue id
- data [json](#) data

Returns [promise](#) promise

getAttachments

get issue's attachment

Parameters

- id [string](#) issue id

Returns [promise](#) promise

deleteAttachments

delete issue's attachment

Parameters

- id [string](#) issue id

Returns [promise](#) promise

put

put issue

Parameters

- id [string](#) issue id
- data [json](#) data

Returns [promise](#) promise

delete

delete issue

Parameters

- id [string](#) issue id

Returns [promise](#) promise

getRights

get issue rights

Parameters

- id **string** issue id

Returns **promise** promise

getIssueTypes

get issue types

Returns **promise** promise

importBcf

import issue from bcf

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** promise

updateImportBcf

update import bcf

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** promise

exportBcf

export issues in bcf format

Parameters

- projectId **string** project id - if not specified export all team issues
- data **json** data
- version

Returns **promise** promise

exportExcel

export issue in excel format

Parameters

- projectId **string** project id - if not specified export all team issues
- data **json** data

Returns **promise** promise

locks

lockObject

lock object

Parameters

- `objectId` **string** object id
- `data` **json** data

Returns **promise** promise

checkLocked

check if object is locked

Parameters

- `objectId` **string** object id

Returns **promise** promise

deleteLock

delete object lock

Parameters

- `objectId` **string** object id

Returns **promise** promise

members

post

post member

Parameters

- `data` **json** member data

Returns **promise** promise

get

get team members

Returns **promise** promise

putUser

put user

Parameters

- `teamSlug` **string** team slug
- `userId` **string** user id
- `data` **json** member data

Returns **promise** promise

deleteUser

delete user

Parameters

- teamSlug **string** team slug
- userId **string** user id

Returns **promise** promise

messages

post

post message

Parameters

- data **json** message data

Returns **promise** promise

getUserMessages

get user messages

Parameters

- userId **string** user id

Returns **promise** promise

getUserDashboardMessages

get user dashboard messages

Parameters

- userId **string** user id

Returns **promise** promise

getSentUserMessages

get messages sent by user

Parameters

- userId **string** user id

Returns **promise** promise

getSentGroupMessages

get messages sent by user to all groups

Parameters

- `userId` **string** user id

Returns **promise** promise

get

get message

Parameters

- `messageId` **string** message id

Returns **promise** promise

getFiltered

get filtered messages

Parameters

- `filter` **string** filter

Returns **promise** promise

deleteMessage

delete message

Parameters

- `messageId` **string** message id

Returns **promise** promise

markAsRead

mark message as readed

Parameters

- `messageId` **string** message id

Returns **promise** promise

updateDashboardAdditionalData

modify additionaldata part of message (title, content, messagelink and picturelink)

Parameters

- `messageId` **string** message id
- `additionalData` **string** message additionalData

Returns **promise** promise

addDashboardMessageAttachment

adding attachment to dashboard message

Parameters

- `messageId` **string** message id
- `additionalData` **string** message additionalData (formData with set files or images(blobs))
- `config`

Returns **promise** promise

getMessagingGroupsDetails

get all messaging groups

Returns **promise** promise

getMessagingGroupsDetails

Get the messaging group details

Returns **promise** promise

models

get

get model

Parameters

- `modelId` **string** model id

Returns **promise** promise

head

get head

Parameters

- `modelId` **string** model id

Returns **[type]** [description]

Returns **promise** promise

getDisciplines

get disciplines

Parameters

- `divisionId` **string** division id

Returns **promise** promise

getRevisions

get revisions

Parameters

- `divisionId` **string** division id

Returns **promise** promise

postRevision

post revision

Parameters

- divisionId **string** division id
- data **json** data

Returns **promise** promise

put

put model

Parameters

- divisionId **string** division id
- data **json** data

Returns **promise** promise

download

download model

Parameters

- divisionId **string** division id

Returns **promise** promise

delete

delete model

Parameters

- divisionId **string** division id

Returns **promise** promise

move

move model

Parameters

- divisionId **string** division id
- moveData **json** move data e.g. { *deltax:delta.x,deltay:delta.y,delta_z:delta.z* }

Returns **promise** promise

objectFilters

get

get object filters

Parameters

- disciplineId **string** discipline id
- projectId **string** project id

Returns **promise** promise

getFilteredObjects

get filtered objects

Parameters

- projectId **string** project id
- filterId **string** filter id
- properties **string** properties
- topology **string** topology (string) returns results organized with respect to existing topology

Returns **promise** promise

applyFilter

apply filter

Parameters

- filterId **string** filter id

Returns **promise** promise

post

post object filter

Parameters

- data **json** object filter data
- projectId **string** projectId

Returns **promise** promise

put

put object filter

Parameters

- data **json** object filter data

Returns **promise** promise

delete

delete object filter

Parameters

- id **string** object filter id

Returns **promise** promise

getFilter

ger object filter info

Parameters

- filterId
- id **string** object filter id

Returns **promise** promise

objects

post

post object

Parameters

- data **json** data

Returns **promise** promise

get

get object

Parameters

- objectId **string** object id
- revision **nubmer** revision number
- properties **string** properties
- projectId **string** project id
- shortInfo **boolean** true = return results without attributes and without localizedAttributeGroups

Returns **promise** promise

getInternal

get internal

Parameters

- objectId **string** object id
- revision **number** revision number

Returns **promise** promise

head

get head

Parameters

- objectId **string** object id
- revision **number** revision number

Returns **promise** promise

getTopology

get object topology

Parameters

- objectId **string** object id
- revision **number** revision number
- hideElements **boolean** hide elements flag
- shortInfo **boolean** short info flag
- batch **string** Response batch id or undefined if no batching used

Returns **promise** promise

getGeometryThreeJS

get object geometry in three js format

Parameters

- objectId **string** object id
- disciplineId **string** discipline id
- chunked **boolean** chunked flag
- revision **number** revision number
- lod **string** level of detail
- properties **string** properties
- batch **string** Response batch id or undefined if no batching used

Returns **promise** promise

getGeometryTemplatesThreeJS

get geometry templates in three js format

Parameters

- objectId **string** object id
- disciplineId **string** discipline id
- chunked **boolean** chunked flag
- revision **number** revision number
- lod **string** level of detail
- batch **string** Response batch id or undefined if no batching used

Returns **promise** promise

getGeometryCompressed

get object compressed geometry

Parameters

- objectId **string** object id
- disciplineId **string** discipline id

Returns **promise** promise

getProperties

get object properties

Parameters

- objectId **string** object id
- propId **string** properties id

Returns **promise** promise

put

get object data

Parameters

- objectId **string** object id
- data **json** data
- projectId

Returns **promise** promise

postComment

post comment

Parameters

- objectId **string** object id
- data **json** data

Returns **promise** promise

getComments

get object comments

Parameters

- objectId **string** object id

Returns **promise** promise

deleteComments

delete object comments

Parameters

- objectId **string** object id

Returns **promise** promise

postAttachment

post object attachemnt

Parameters

- objectId **string** object id
- data **json** data
- uploadProgress
- documentFolderId **string** document folder id

Returns **promise** promise

getAttachments

get object attachments

Parameters

- objectId **string** object id
- revision **number** revision number

Returns **promise** promise

deleteAttachments

delete all object's attachment

Parameters

- objectId **string** object id

Returns **promise** promise

postHyperlink

post object hyperlink

Parameters

- objectId **string** object id
- data **json** data

Returns **promise** promise

getHyperlinks

get object hyperlinks

Parameters

- objectId **string** object id

Returns **promise** promise

deleteHyperlinks

delete all object's hyperlinks

Parameters

- objectId **string** object id

Returns **promise** promise

delete

delete object

Parameters

- objectId

Returns **promise** promise

getAttributes

get object attributes

Parameters

- objectId **string** object id
- documentVersion **number** document version
- projectId **string** projectId - required when resource (objectId) represents an user
- revision **number** revision number

Returns **promise** promise

getAttributeValues

get all objects of a project with existing attribute and its value

Parameters

- projectId **string** project id
- attributeId **string** attribute id

Returns **promise** promise

getComplexPropertiesTopology

get ComplexProperties topology for element

Parameters

- `objectId` **string** object id
- `revision` **string** revision

Returns **promise** promise

postThumbnail

post thumbnail

Parameters

- `objectId` **string** object id
- `data` **json** data
- `uploadProgress`

Returns **promise** [description]

addChildObject

post addChildObject

Parameters

- `data` **json** minimal form is: { "parent": "{{parentId}}", "type": "Proxy", "elementtyp": "30114B52-BC05-47C1-80B5-C7D5485D5831" "attributes": { "general": { "name": "IPE200_Column" }, "element": { "divisionid": "{{divisionId}}", "logparentid": "{{parentId}}" } } }

Returns **promise** [description]

getReports

get object reports

Parameters

- `objectId` **string** object id
- `version` **string** version
- `action`

Returns **promise** promise

pins

get

get pins

Parameters

- `id` **string** pin id

Returns **promise** promise

delete

delete pin

Parameters

- `id` **string** pin id

Returns **promise** promise

put

put pin

Parameters

- id **string** pin id
- data **json** data

Returns **promise** promise

projectActionsRights

get

get project action rights

Returns **promise** promise

projectAttributeTemplates

post

post project attribute template

Parameters

- data **json** template data

Returns **promise** promise

get

get project attribute template - if id is not specified return all templates

Parameters

- id **string** template id

Returns **promise** promise

put

put project template attribute

Parameters

- id **string** template id
- data **json** template data

Returns **promise** promise

delete

delete project template attribute

Parameters

- id **string** template id

Returns **promise** promise

copyFrom

copy project template attribute

Parameters

- fromId **string** source template id
- targetId **string** destination template id

Returns **promise** promise

projects

post

post project

Parameters

- data **json** project data

Returns **promise** promise

put

put project

Parameters

- id **string** project id
- data **json** project data

Returns **promise** promise

get

get project - if no id is specified return list of all projects

Parameters

- id **string** project id
- noDisciplines **boolean** flag to put disciplines in response
- teamSlug **string** team slug, if defined, all projects for this team slug will be loaded

Returns **promise** [description]

delete

delete project

Parameters

- projectId **string** project id

Returns **promise** [description]

getModels

get project's models

Parameters

- `projectId` **string** project id

Returns **promise** [description]

getModelTypes

get all model types for given team

Returns **promise** [description]

getFreeAttrModelTypes

get all model types for given team base on free attr information

Returns **promise** [description]

postModel

post model

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

getMembers

get project members

Parameters

- `projectId` **string** project id

Returns **promise** [description]

postMember

post member

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

putMember

put member

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

deleteMember

delete member

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

getPins

get project pins

Parameters

- `projectId` **string** project id

Returns **promise** [description]

putProject

put project

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

postAttachment

post attachment

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

getAttachments

get attachments

Parameters

- `projectId` **string** project id
- `revision` **number** revision number
- `queryParams` **json** json with params e.g. {projectOnly : true}

Returns **promise** [description]

deleteAttachments

delete attachment

Parameters

- `projectId` **string** project id

Returns **promise** [description]

getTopology

get project's topology

Parameters

- projectId **string** project id

Returns **promise** [description]

getSpots

get spots

Parameters

- projectId **string** project id

Returns **promise** [description]

postThumbnail

post thumbnail

Parameters

- projectId **string** project id
- data **json** data
- uploadProgress

Returns **promise** [description]

getThumbnail

get thumbnail

Parameters

- thumbnailId **string** thumbnail id

Returns **promise** [description]

getComments

get comments

Parameters

- projectId **string** project id

Returns **promise** [description]

getHyperlinks

get hyperlinks

Parameters

- projectId **string** project id

Returns **promise** [description]

getIssues

get issues (tasks)

Parameters

- `projectId` **string** project id

Returns **promise** [description]

getIssuesShortInfo

get issues short info(tasks)

Parameters

- `projectId` **string** project id

Returns **promise** [description]

postSlideshow

post slideshow

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** [description]

getSlideshows

get slideshows

Parameters

- `projectId` **string** project id

Returns **promise** [description]

deleteSlideshows

delete slideshows

Parameters

- `projectId` **string** project id

Returns **promise** [description]

getProjectInfo

get project info

Parameters

- `projectId` **string** project id

Returns **promise** promise

revisions

postRevision

post revision

Parameters

- id **string** division (model) id
- data **json** data

Returns **promise** promise

getRevisionList

get revisions list

Parameters

- id **string** division (model) id

Returns **promise** promise

getRevisionInfo

get revision info

Parameters

- id **string** revision id

Returns **promise** promise

putRevisionInfo

put revision info

Parameters

- id **string** revision id
- data **json** data

Returns **promise** promise

getRevisionsComparisonResult

get revision comparison result

Parameters

- id **string** division (model) id
- revisionNumber1 **number** revision number 1
- revisionNumber2 **number** revision number 2
- queryParams

Returns **promise** promise

compareRevisions

compare revisions

Parameters

- id **string** division(model) id
- revisionNumber1 **number** revision number 1
- revisionNumber2 **number** revision number 2
- queryParams

Returns **promise** promise

rightsService

getRoles

get right roles

Parameters

- `projectId` **string** project id - if not specified return all roles

Returns **promise** promise

postUserRole

post user role

Parameters

- `projectId` **string** project id - if not specified return all roles
- `data` **json** role data

Returns **promise** promise

putUserRole

put user role

Parameters

- `projectId` **string** project id - if not specified return all roles
- `data` **json** role data

Returns **promise** promise

deleteUser

delete user

Parameters

- `projectId` **string** project id - if not specified return all roles

Returns **promise** promise

roles

get

get roles

Parameters

- `rights` **boolean**
- `customrole`
- `customRole` **boolean** if true, returns only editable roles

Returns **promise** promise When query parameter "rights=false" is defined, the API call will returns all roles. i.e. not only roles which contains resources and right definitions. When query parameter is missing, the API call returns only roles which includes resources and rights definitions.

getAllRights

get all rights

Parameters

- filter **JSON** { globalFreeAttributes:true/false global:true/false, project:true/false, document:true/false, layer:true/false }

Returns **promise** promise

deleteRole

create new role

Parameters

- newRole **json**

Returns **promise** promise newRoleJSON Structure: prop-name mandatory/optional Type Description resources mandatory JSON Array Array of the resource definition objects for which the role is valid id optional string (guid) Id of the role name mandatory string The name of the role parent optional string (guid) The Id of the parent role customRole mandatory boolean When this flag is true the role is custom role otherwise will role is 'built in' role> > JSON Structure of the resource definition object: > > resource mandatory string Describe kind of right e.g. "UserRightGlobal" or "UserRightLayer" > > rights mandatory JSON Array Array of the strings of rights e.g.: "FreeAttributeView", "MEP: 2", "MEPEdit" > > <<

deleteRole

delete existing role

Parameters

- roleId **string**

Returns **promise** promise

getRoleDetails

get details of role

Parameters

- roleId **string**

Returns **promise** promise

updateRole

update existing role

Parameters

- roleId **string**
- updatedRole **json** (see create newRole for JSON details)

Returns **promise** promise

sectionCut

selections

post

Creates a new selection object. The structure of the Selection objects is the same like the structure of the Export Object Selection.

Parameters

- `projectId` **string** project id
- `data` **json** selection data

Returns **promise** promise

get

get selection object details

Parameters

- `selectionId` **string** selection id

Returns **promise** [description]

getAllSelections

get all selections of the project

Parameters

- `projectId` **string** selection id

Returns **promise** [description]

delete

delete the selection

Parameters

- `selectionId` **string** selection id

Returns **promise** [description]

share

post

post share

Parameters

- `data` **json** share data

Returns **promise** promise

slides

get

get slide

Parameters

- slideId **string** slide id

Returns **promise** promise

put

put slide

Parameters

- slideId **string** slide id
- data **json** slide data

Returns **promise** promise

delete

delete slide

Parameters

- slideId **string** slide id

Returns **promise** promise

postThumbnail

post thumbnail

Parameters

- slideId **string** slide id
- data **json** thumbnail data

Returns **promise** promise

slideshows

get

get slideshow

Parameters

- slideshowId **string** slideshow id

Returns **promise** promise

put

put slideshow

Parameters

- slideshowId **string** slideshow id
- data **json** data

Returns **promise** promise

delete

delete slideshow

Parameters

- `slideshowId` **string** slideshow id

Returns **promise** promise

postSlide

post slide

Parameters

- `slideshowId` **string** slideshow id
- `data` **json** data

Returns **promise** promise

getSlides

get slides

Parameters

- `slideshowId` **string** slideshow id

Returns **promise** promise

postThumbnail

post slideshow thumbnail

Parameters

- `slideshowId` **string** slideshow id
- `data` **json** data

Returns **promise** promise

getThumbnail

get slideshow thumbnail

Parameters

- `thumbnailId` **string** thumbnail id

Returns **promise** promise

structures

postStructure

post structure

Parameters

- `projectId` **string** project id
- `data` **json** data

Returns **promise** promise

postStructureTopology

post structure topology

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** promise

postSubNode

post sub node

Parameters

- structureId **string** structure id
- data **json** data

Returns **promise** promise

postAssignedObject

post assigned object - assign object to structure

Parameters

- structureId **string** structure id
- objectId **string** object id
- varianttype **string** variant type
- data **json** data

Returns **promise** promise

attributeAssign

assign attribute to structure

Parameters

- projectId **string** project id
- structureId **string** structure id
- attributeId **string** attribute id
- varianttype **string** variant type

Returns **promise** promise

deleteAssignedObject

delete assigned object from structure

Parameters

- structureId **string** structure id
- objectId **string** object id
- varianttype **string** variant type

Returns **promise** promise

getStructure

get structure - is structure id is empty return all structures with specific type

Parameters

- projectId **[type]** [description]
- structureId **string** structure id
- type **string** requested structure type

Returns **promise** promise

getTopology

get structure topology

Parameters

- structureId **string** structure id
- varianttype **string** variant type
- attributes **boolean** flag if result should contain attributes

Returns **promise** promise

getTopologyStructure

get topology structure

Parameters

- projectId **[type]** [description]

Returns **promise** promise

resetTopologyStructure

create or reset topology structure

Parameters

- projectId **[type]** [description]

Returns **promise** promise

copyStructure

copy structure

Parameters

- structureId **string** structure id
- data **json** data
- copyElements **boolean** copy elements

Returns **promise** promise

putStructure

put structure

Parameters

- structureId **string** structure id
- data **json** data

Returns **promise** promise

reorderStructure

reorder structure

Parameters

- structureId **string** structure id
- varianttype **string** variant type

Returns **promise** promise

deleteStructure

delete structure

Parameters

- structureId **string** structure id

Returns **promise** promise

deleteAllProjectStructures

delete all project structures

Parameters

- projectId **string** project id

Returns **promise** promise

postVariantType

post variant type

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** promise

getVariantType

get variant type

Parameters

- projectId **string** project id
- varianttype **string** variant type

Returns **promise** promise

deleteVariantType

delete variant type

Parameters

- projectId **string** project id
- varianttype **string** variant type

Returns **promise** promise

copyElementsToVariant

copy elements to variant

Parameters

- `projectId` **string** project id
- `varianttype` **string** variant type
- `variant` **json** json string with variant data

Returns **promise** promise

teamActionsRights

get

get team action rights

Returns **promise** promise

teams

get

get team data if id is not specified return list of teams

Parameters

- `id` **string** team id - if not specified return list of teams

Returns **promise** promise

postTeam

post team

Parameters

- `data` **json** team data

Returns **promise** promise

putTeam

put team

Parameters

- `teamId` **string** team id
- `data` **json** team data

Returns **promise** promise

deleteTeam

delete team

Parameters

- `teamId` **string** team id

Returns **promise** promise

getAccountSettings

get team account settings

Parameters

- teamId **string** team id

Returns **promise** promise

getRestrictions

get team restrictions

Parameters

- teamId **string** team id

Returns **promise** promise

terms

get

get terms

Parameters

- language **string** terms language

Returns **promise** promise

units

post

post units

Parameters

- data **json** units data

Returns **promise** promise

get

get unit - if id is not specified return all units

Parameters

- unitId **string** unit id

Returns **promise** promise

put

put unit

Parameters

- unitId **string** unit id
- data **json** unit data

Returns **promise** promise

delete

delete unit

Parameters

- unitId **string** unit id

Returns **promise** promise

user

get

get user

Parameters

- token **[type]** [description]

Returns **promise** [description]

put

put

Parameters

- userId **[type]** [description]
- data **[type]** [description]
- token **[type]** [description]

Returns **promise** promise

post

post user

Parameters

- data **[type]** [description]

Returns **promise** [description]

postPhoto

post user photo

Parameters

- data **[type]** image data

Returns **promise** [description]

getPhoto

get user photo

Returns **[type]** [description]

putPassword

put (change) password

Parameters

- data **[type]** [description]

Returns **promise** promise

getSubscriptions

get user subscriptions

Parameters

- id **string** user id

Returns **promise** promise

userGroups

postGroup

post group

Parameters

- projectId **string** project id
- data **json** data

Returns **promise** [description]

getGroups

get groups

Parameters

- projectId **string** project id

Returns **promise** promise

getUsers

get users of a group

Parameters

- groupId **string** group id

Returns **promise** promise

deleteGroup

delete group

Parameters

- groupId **string** group id

Returns **promise** promise

putGroup

put group

Parameters

- groupId **string** group id
- data **json** data

Returns **promise** promise

assignUser

assign user to a group or change group role

Parameters

- groupId **string** group id
- data **json** data

Returns **promise** promise

users

post

post users

Parameters

- data **json** [description]

Returns **promise** promise

exist

check if user exists

Parameters

- email **string** user email

Returns **promise** promise

put

put user

Parameters

- userId **string** user id
- data

Returns **promise** promise

delete

delete user

Parameters

- `userId` **string** user id

Returns **promise** promise

setSubscriptions

create one or more trial account setting(s)

Parameters

- `userId` **string** user id
- `data` **json** [description]

Returns **promise** promise

userSettings

postSettings

post settings

Parameters

- `category` **string** key of user settings
- `data` **json** settins data

Returns **promise** promise

getSettings

get settings

Parameters

- `category` **string** key of user settings

Returns **promise** promise