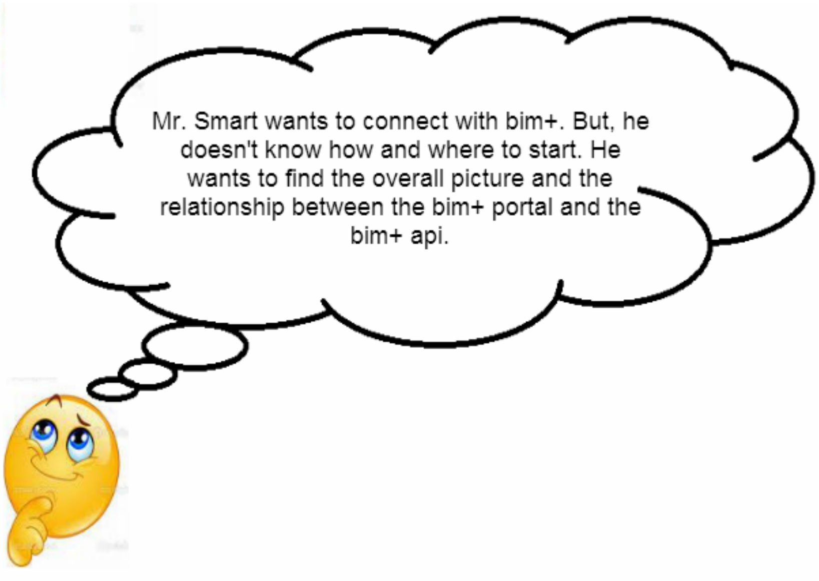
User Stories

> Help us improve the Bimplus API doc by providing your valuable comments. You can find the comment section at the bottom of the page.

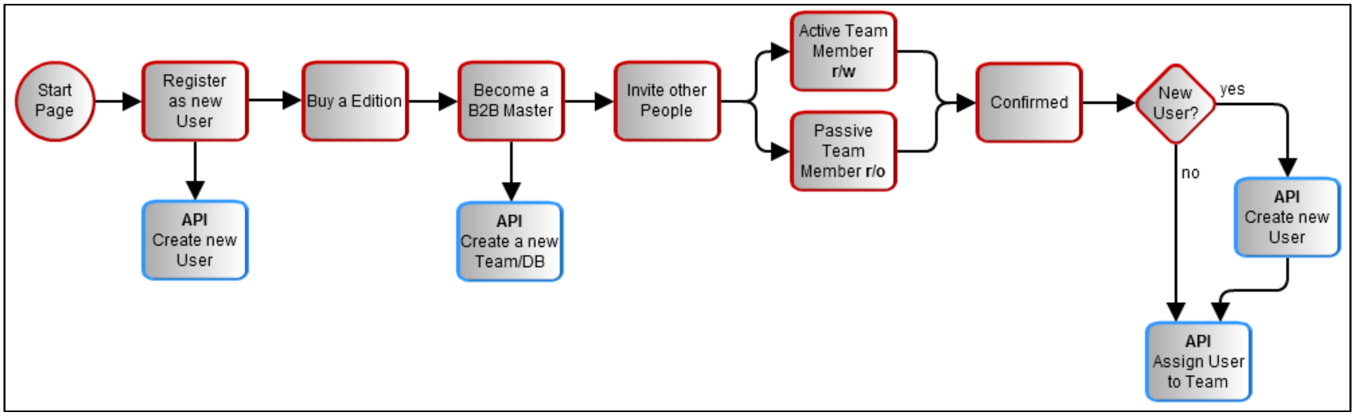
> Developers please use https://api-stage.bimplus.net/v2 (stage version of Bimplus API) and http://portal-stage.bimplus.net/ (stage version of Shop /Portal) for testing purposes. The production version of the portal is located under https://portal.bimplus.net/ and the base url of the API is different for the production version.

Connect with Bimplus. Relation(workflow) between Bimplus portal and Bimplus api


Mr. Smart wants to connect with bim+. But, he doesn't know how and where to start. He wants to find the overall picture and the relationship between the bim+ portal and the bim+ api.

- Bimplus portal is the first place of contact of the user/developer for using the Bimplus platform(stage version). The production version of the portal can be found under shop where you can purchase different editions of Bimplus cloud platform. Here you can also find a few critical foundation apps like the Bimplus Explorer, Bimplus Viewer and Bimplus Problem Spotter. The portal takes care of the user registration(developer registration) as well.
- The Bimplus API(development version url : https://api-stage.bimplus.net/v2) provides an interface for accessing the building information and models. It connects the building project information to a vast number of developers providing innovative Apps / applications that can operate on the building models and information.

The figure below describes the relation between the Bimplus portal and the API.



Become a Bimplus compliant developer/partner. Register and authorize yourself to access the Bimplus resources. Relation between teams and projects

After getting the overall picture of bim+ and its api, Mr. Smart wants to register and start using the bim+ api. He is trying to figure out the things that he needs to do for becoming a bim+ compliant developer and a partner.

- Please register here for accessing the Bimplus resources and become a Bimplus compliant developer and a partner. You will receive an authentication token after the registration.
- A Team is a tenant/company which work on the projects. The team has to be specified and setup through the bimplus portal as so called slugs to present a clean URL . The team slug will be used to know which tenant database to work on.
- Using the authentication token and the team slug, you can start working with the building projects which you are a member of, using our Bimplus api.
- Please find a quick start tutorial here .

Create a team. Create an user. Assign the newly created user to the newly created team. Verify whether he/she is part of the team

Mr. Smart would like to create a team to work on his projects. He wants to create a new user account for his colleague who is an architect and would like to make him member of the newly created team. After that, he wants to verify whether the architect is really part of his team.

The Bimplus provides a few user administration relevant services. They are:

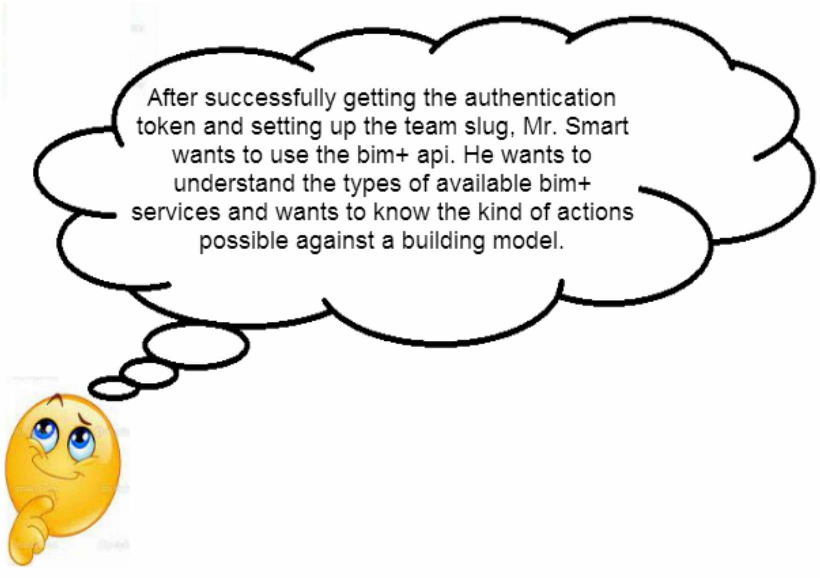
- User Management Service
- Team Management Service
- Membership Management Service
- Please use the Bimplus Team Management Service (POST https://api-stage.bimplus.net/v2/teams) for creating a team. This will basically create a data storage for the team. See in the Request/Response tab where we have created the "best-company" team. Also, please note that the team slug "best-company" will be part of the subsequent requests for accessing the resources and performing the desired actions on them.

- Create a new user using the Bimplus User Management Service(POST https://api-stage.bimplus.net/v2/users).

- Add the user which you have created to the team using the Bimplus Membership Management Service(POST https://api-stage.bimplus.net/v2 //members).

- Verify whether the user has been added to the team by using the Bimplus Membership Management Service(GET https://api-stage.bimplus.net/v2 //members). In the response tab, you can see the user(test@bimplus.net) who is part of the "best-company" team.
- For authorization, user should perform OIDC authentication and receive token (preferrably within browser UI)

Create, modify and access different resources/objects of the building model using the Bimplus api


After successfully getting the authentication token and setting up the team slug, Mr. Smart wants to use the bim+ api. He wants to understand the types of available bim+ services and wants to know the kind of actions possible against a building model.

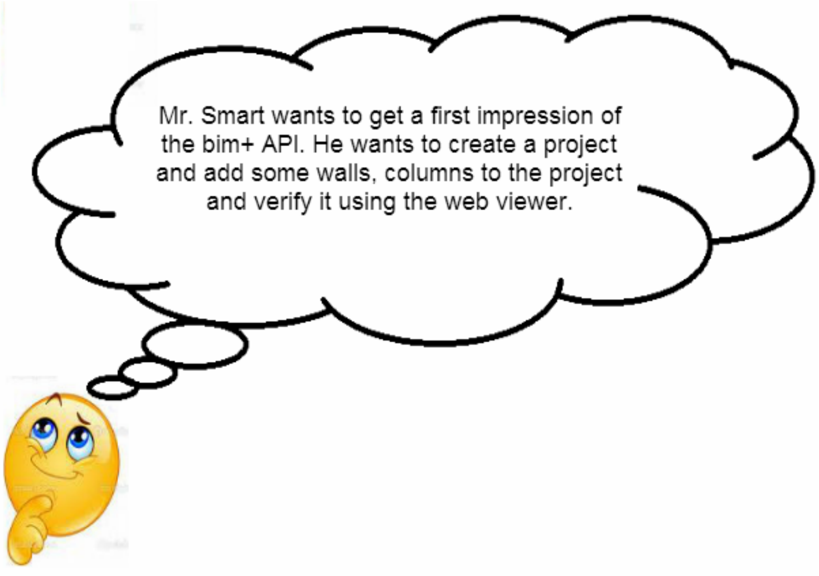
The **project slug** (which should be provided during the creation of a project) should be provided as part of the URL for accessing all the project relevant resources. This is also necessary to verify the user’s access rights on the project in an early stage of processing the API call (before any business data will be touched and any business logic will be executed)

These are the project relevant services available at the moment:

- Project Service
- Object Service
- ElementType Service
- Division Service
- Attachment Service
- Issue Service
- Pin Service
- Import Service

For further details see the Bimplus api reference

Create a project, add an object(wall) and view it using the web viewer


Mr. Smart wants to get a first impression of the bim+ API. He wants to create a project and add some walls, columns to the project and verify it using the web viewer.

1. Create a new project

Use the Bimplus Project Service for creating a new project.

HTTP Method

POST

URL / Resource / JSON Structure

Resource: projects

URL: https://api-stage.bimplus.net/v2/<team>/projects

Example: https://api-stage.bimplus.net/v2/bimplus-gmbh/projects

JSON Structure:

Name	Mandatory / Optional	Type	Description
name	optional	string	Name of the project
shortDescr	optional	string	Short description of the project
created	will be ignored	string (date)	Creation date of the project

Description

Creates a new project.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

JSON

```
{
"name" : "Your Wonderful Model",
"shortDescr" : "The best building model in the world"
}
```

Response

Status

```
Status: 201 Created
```

JSON

```
{
    "name": "Your Wonderful Model",
    "shortDescr": "The best building model in the world",
    "teamName": "Bimplus GmbH",
    "teamSlug": "bimplus-gmbh",
    "hasWriteAccess": true,
    "thumbnail": null,
    "created": "2014-01-30T15:43:55.1302187Z",
    "changed": "2014-01-30T15:43:55.1302187Z",
    "disciplines": null,
    "id": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5"
}
```

2. Create a model under the project

HTTP Method

POST

URL / Resource / JSON Structure

Resource: divisions

URL: https://api-stage.bimplus.net/v2/<team_slug>/projects/<project_id>/divisions

Example: https://api-stage.bimplus.net/v2/bimplus/projects/69e48f92-a8dc-4af5-8f71-187b40a8b8f5/divisions

JSON Structure:

Name	Mandatory / Optional	Type	Description
name	mandatory	string	Name of the model
description	optional	string	Short description of the model
projectId	will be ignored	string (guid)	Id of the project to which the model belongs
topologyId	optional	string (guid)	Id of the topology to which the model belongs
disciplines	optional	object	The discipline list of the model

Description

Creates a new model.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

JSON

```
{
"name" : "Nemetschek Building",
"shortDescr" : "Model created via API"
}
```

Response

Status

```
Status: 201 Created
```

JSON

```
{
    "name": "Nemetschek Building",
    "description": null,
    "projectId": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5",
    "id": "4496b8ad-ba5a-483c-9912-2cb837178708"
}
```

3. Create a topology node

A Topology node has to be created for adding the objects (Column, Wall). As a topology node is also an object, the Bimplus Object Service has to be used for creating the topology object. Please see the request tab. The id of the project is provided as the parent of the topology node object. The model id has to be provided as the "divisionid" attribute.

HTTP Method

POST

URL / Resource / JSON Structure

Resource: objects

URL: https://api-stage.bimplus.net/v2/<team>/objects

Example: https://api-stage.bimplus.net/v2/bimplus/objects

JSON Structure:

Name	Mandatory / Optional	Type	Description
parent	optional	string (guid)	The id of the parent to which the object can be associated
type	mandatory	string	Element type of the object
attributes	optional	string	Attributes of the object
children	optional	Object	Children of the object

Description

Creates a new object

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

JSON

```
{
    "parent": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5",
    "type": "TopologyDivision",
    "attributes": {
        "element": {
            "isparent": true,
            "elementtyp": "e003e080-f21b-4f8a-8c4c-4c9c5026cf50",
            "divisionid": "4496b8ad-ba5a-483c-9912-2cb837178708"
        },
        "general": {
            "name": "created by objects post",
            "isobjectvalid": true
        }
    }
}
```

Response

Status

```
Status: 201 Created
```

JSON

```
{
    "id": "b793c860-d3b8-48a5-aa42-9df9e24a0d71",
    "parent": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5",
    "type": "TopologyDivision",
    "attributes": {
        "element": {
            "isparent": true,
            "elementtyp": "e003e080-f21b-4f8a-8c4c-4c9c5026cf50",
            "divisionid": "4496b8ad-ba5a-483c-9912-2cb837178708",
            "logparentid": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5"
        },
        "general": {
            "name": "created by objects post",
            "isobjectvalid": true
        }
    }
}
```

4. Add a wall to the project

Again we use the Bimplus Object Service for creating the wall object in the project. Please note that the wall is added to the topology node and not the project (i.e See the request tab. The id of the topology node is provided for the parent attribute of the json object(wall))

HTTP Method

POST

URL / Resource / JSON Structure

Resource: objects

URL: https://api-stage.bimplus.net/v2/<team>/objects

Example: https://api-stage.bimplus.net/v2/bimplus/objects

JSON Structure:

Name	Mandatory / Optional	Type	Description
parent	optional	string (guid)	The id of the parent to which the object can be associated
type	mandatory	string	Element type of the object
attributes	optional	string	Attributes of the object
children	optional	Object	Children of the object

Description

Creates a new object

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

JSON

```
{
    "parent": "b793c860-d3b8-48a5-aa42-9df9e24a0d71",
    "type": "Wall",
    "attributes": {
        "element": {
            "elementtyp": "10074eef-9418-4d64-9c6d-23932835a7f1",
            "divisionid": "4496b8ad-ba5a-483c-9912-2cb837178708"
        },
        "general": {
            "name": "a new test wall without walllayer"
        },
        "quantity": {
            "width": 100,
            "volume": 200.4
        },
        "geometry": {
            "mesh": {
                "color": 4291407461,
                "vertices": [
                    -3486.667,
                    0,
                    -3000,
                    8993.333,
                    0,
                    -3000,
                    8993.333,
                    300,
                    -3000,
                    -3486.667,
                    300,
                    -3000,
                    -3486.667,
                    0,
                    -400,
                    8993.333,
                    0,
                    -400,
                    8993.333,
                    300,
                    -400,
                    -3486.667,
                    300,
                    -400
                ],
                "faces": [
                    4,
                    3,
                    0,
                    1,
                    2,
                    4,
                    4,
                    7,
                    6,
                    5,
                    4,
                    0,
                    3,
                    7,
                    4,
                    4,
                    1,
                    0,
                    4,
                    5,
                    4,
                    2,
                    1,
                    5,
                    6,
                    4,
                    3,
                    2,
                    6,
                    7
                ]
            }
        }
    }
}
```

Response

Status

```
Status: 201 Created
```

JSON

```
{
    "id": "b2a92252-ae51-4c72-ab1e-fafa141ac54c",
    "parent": "b793c860-d3b8-48a5-aa42-9df9e24a0d71",
    "type": "Wall",
    "attributes": {
        "element": {
            "elementtyp": "10074eef-9418-4d64-9c6d-23932835a7f1",
            "divisionid": "4496b8ad-ba5a-483c-9912-2cb837178708",
            "logparentid": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5"
        },
        "general": {
            "name": "a new test wall without walllayer"
        },
        "quantity": {
            "width": 100,
            "volume": 200.4
        },
        "geometry": {
            "mesh": {
                "color": 4291407461,
                "vertices": [
                    -3486.667,
                    0,
                    -3000,
                    8993.333,
                    0,
                    -3000,
                    8993.333,
                    300,
                    -3000,
                    -3486.667,
                    300,
                    -3000,
                    -3486.667,
                    0,
                    -400,
                    8993.333,
                    0,
                    -400,
                    8993.333,
                    300,
                    -400,
                    -3486.667,
                    300,
                    -400
                ],
                "faces": [
                    4,
                    3,
                    0,
                    1,
                    2,
                    4,
                    4,
                    7,
                    6,
                    5,
                    4,
                    0,
                    3,
                    7,
                    4,
                    4,
                    1,
                    0,
                    4,
                    5,
                    4,
                    2,
                    1,
                    5,
                    6,
                    4,
                    3,
                    2,
                    6,
                    7
                ]
            }
        },
        "elementstate": {
            "state": "0aaf5a1c-2453-40cb-9232-315119dd288f"
        }
    }
}
```

5. Verify whether the wall and column is added to the project by seeing the object topology

For verifying whether the wall and the column is added to the project, we have to call GET https://api-stage.bimplus.net/v2/nemetschek/objects//topology. Please note that the API call GET https://api-stage.bimplus.net/v2/nemetschek/projects//topology will only give you its children (i.e topology) details. So, if your project is very big, you can first call GET https://api-stage.bimplus.net/v2/nemetschek/projects//topology and get the topology id(i.e 7f76a497-e322-4b78-b820-1ce85ebc39db in our example). Using the topology id we can get the object(wall, column in our case) details using the call GET https://api-stage.bimplus.net/v2/nemetschek/objects/7f76a497-e322-4b78-b820-1ce85ebc39db/topology.

HTTP Method

GET

URL / Resource / JSON Structure

Resource: objects/<object_id>/topology

URL: https://api-stage.bimplus.net/v2/<team>/objects/<object_id>/topology

Example: https://api-stage.bimplus.net/v2/bimplus-gmbh/objects/69e48f92-a8dc-4af5-8f71-187b40a8b8f5/topology

Description

Get all the project properties of a project using its id.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

Response

Status

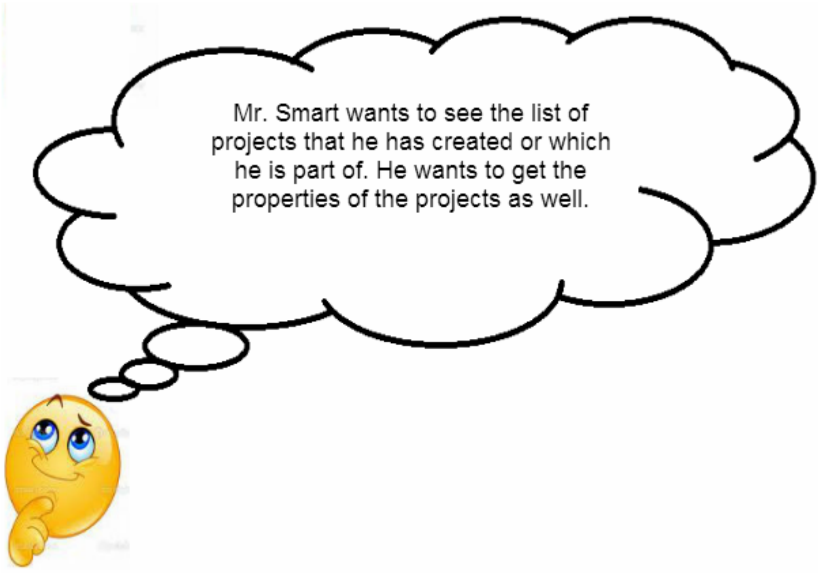
```
Status: 200 OK
```

JSON

```
{
    "parent": null,
    "name": "Your Wonderful Model",
    "type": "ID_Project",
    "children": [
        {
            "parent": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5",
            "name": "created by objects post",
            "type": "ID_TopologyDivision",
            "children": [
                {
                    "parent": "b793c860-d3b8-48a5-aa42-9df9e24a0d71",
                    "name": "a new test wall without walllayer",
                    "type": "ID_ArchWall",
                    "children": [],
                    "id": "b2a92252-ae51-4c72-ab1e-fafa141ac54c"
                }
            ],
            "id": "b793c860-d3b8-48a5-aa42-9df9e24a0d71"
        }
    ],
    "id": "69e48f92-a8dc-4af5-8f71-187b40a8b8f5"
}
```

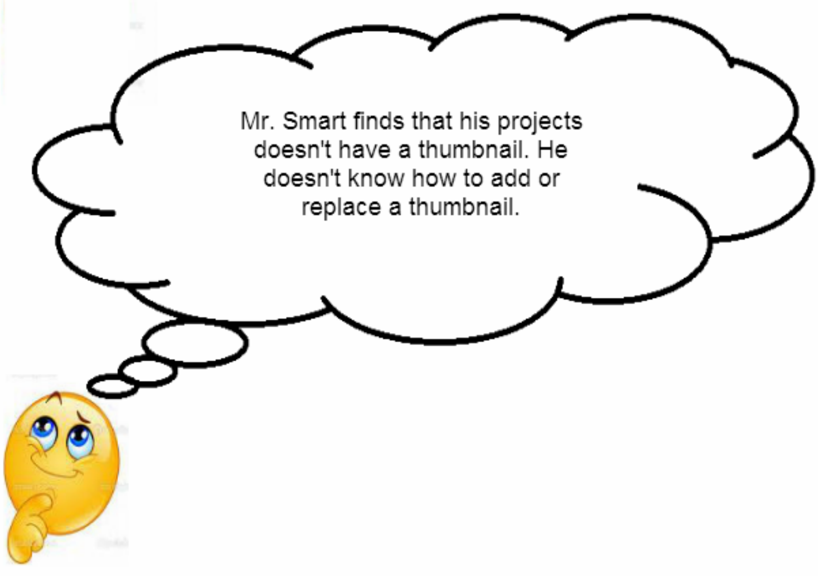
6. See your result visually by using the web viewer.

Get the list of available projects and its properties


Mr. Smart wants to see the list of projects that he has created or which he is part of. He wants to get the properties of the projects as well.

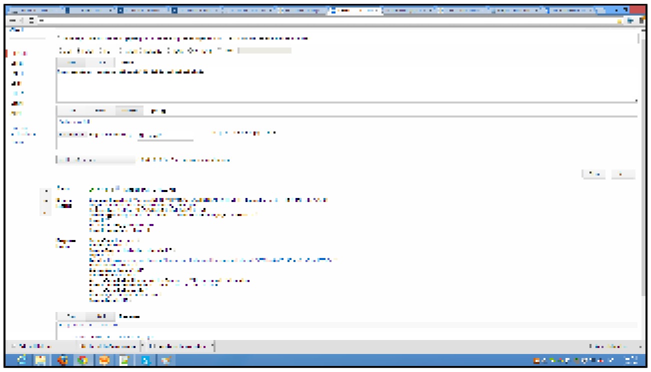
- This API call(GET https://api-stage.bimplus.net/v2//projects) should retrieve the list of projects available to the team. In this call, only a few properties of the projects will be visible.
- Using the project id got from the previous call, the full property list can be accessed using this API call(GET https://api-stage.bimplus.net/v2//projects/)

Add a thumbnail to the project

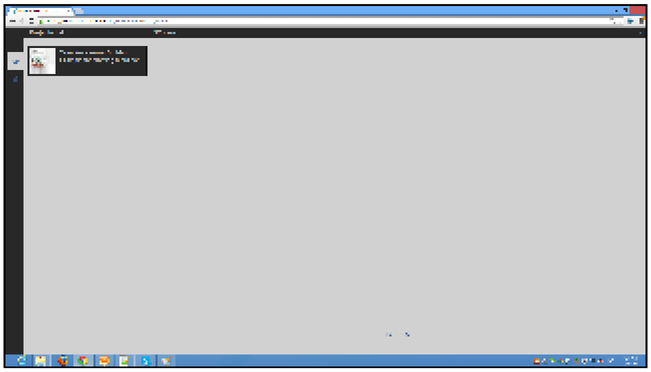

Mr. Smart finds that his projects doesn't have a thumbnail. He doesn't know how to add or replace a thumbnail.

- Thumbnail can be added to a project by using the Bimplus Project Service(POST https://api-stage.bimplus.net/v2//projects//thumbnail). There is also a service available for downloading the thumbnail.

Thumbnail API call:

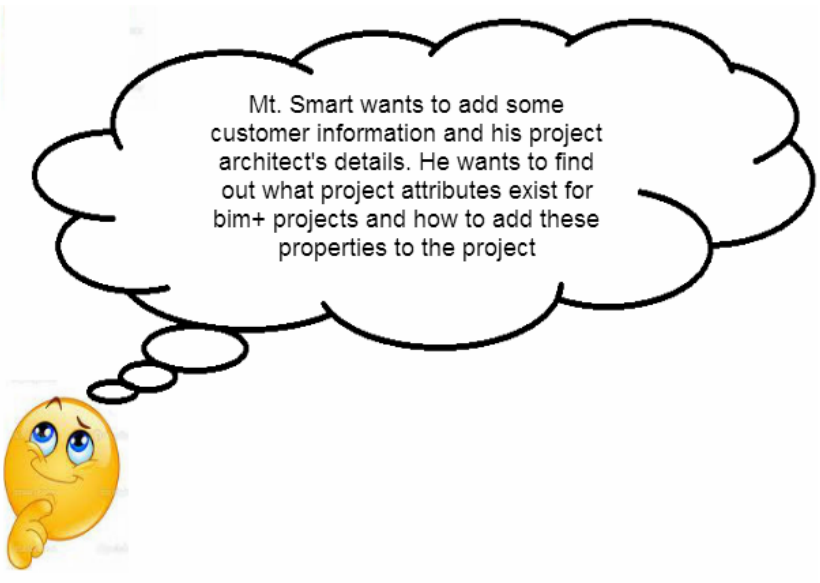


Web Viewer:



- Please, note that there is a seperate attachment service available for uploading attachments other than thumbnail.

Add properties to the project


Mt. Smart wants to add some customer information and his project architect's details. He wants to find out what project attributes exist for bim+ projects and how to add these properties to the project

- The project can be updated with the necessary attributes. The list of available Bimplus attributes can be seen here
- Update the project with the project properties

HTTP Method

PUT

URL / Resource / JSON Structure

Resource: projects/<project_id>

URL: https://api-stage.bimplus.net/v2/<team>/projects/<project_id>

Example: https://api-stage.bimplus.net/v2/bimplus/projects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e

JSON Structure:

Name	Mandatory / Optional	Type	Description
name	optional	string	Name of the project
shortDescr	optional	string	Short description of the project
created	will be ignored	string (date)	Creation date of the project
changed	will be ignored	string (date)	Changed date of the project

Description

Update or replace a specified existing project with a new one.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

JSON

```
{
    "properties": {
        "MainFloorArea": "456m²",
        "Architect": {
            "name": "NemetschekTech.GmbH",
            "phone1": "089/12793-1115",
            "phone2": "0171/71384483",
            "email": "MyName@Nemetschek.com",
            "city": "Munich",
            "zip": "81829",
            "street": "Konrad-Zuse-Platz1"
        },
        "Customer": {
            "name": "Bimplus customer",
            "phone1": "001-201-1279345",
            "email": "MyName@Nemetschek.com",
            "city": "New Jersy",
            "zip": "08201"
        }
    }
}
```

Response

Status

```
Status: 200 OK
```

- See the updated project properties

HTTP Method

GET

URL / Resource / JSON Structure

Resource: projects/<project_id>

URL: https://api-stage.bimplus.net/v2/<team>/projects/<project_id>

Example: https://api-stage.bimplus.net/v2/bimplus/projects/586b02be-43b8-4e27-b698-e067e85e38e2

Description

Get all the project properties of a project using its id.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

Response

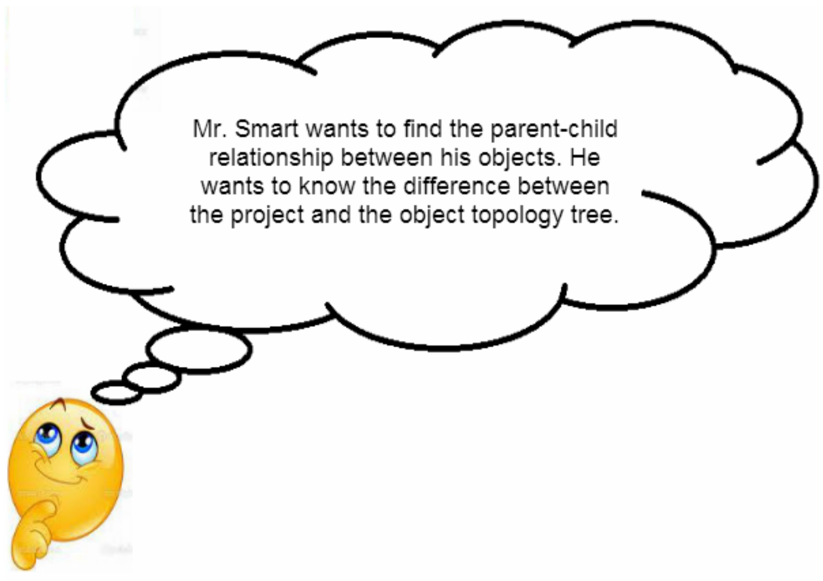
Status

```
Status: 200 OK
```

JSON

```
{
    "name": "Your Awesome Building",
    "shortDescr": "The example project",
    "thumbnail": "/nemetschek/projects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e/thumbnail/download",
    "created": "0001-01-01T00:00:00",
    "changed": "0001-01-01T00:00:00",
    "disciplines": [
        {
            "id": "0f106af0-a919-44c5-b211-15bd5ef620b6",
            "name": "ID_BuildingModel",
            "divisionId": "ac21b9a1-8853-4658-9fbf-0cf62340bd18",
            "divisionName": null,
            "revisions": []
        }
    ],
    "properties": {
        "mainfloorarea": "456m²",
        "architect": {
            "name": "NemetschekTech.GmbH",
            "street": "Konrad-Zuse-Platz1",
            "phone1": "089/12793-1115",
            "phone2": "0171/71384483",
            "email": "MyName@Nemetschek.com",
            "zip": "81829",
            "city": "Munich"
        },
        "customer": {
            "name": "Bimplus customer",
            "phone1": "001-201-1279345",
            "email": "MyName@Nemetschek.com",
            "zip": "08201",
            "city": "New Jersy"
        }
    },
    "id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

Project topology and object topology tree


Mr. Smart wants to find the parent-child relationship between his objects. He wants to know the difference between the project and the object topology tree.

Topology is a hierarchical tree representing all its child nodes. A topology tree consist of components which can have a graphical representation(objects or elements) or components which does not have a graphical representation(nodes). A topology tree can be obtained both on the project as well as on the object level.

- Please use the Bimplus Project Service for obtaining/filtering the project topology tree. This will yield the topology nodes filtered till the sub-project level.

Project Topology Tree (GET api.bimplus.net/v2/<team>/projects/<project_id>/topology)

```
{
"parent": null,
"name": null,
"type": "Project",
"children": \[{
            "parent": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
            "name": null,
            "type": "ID_TopologyDivision",
            "children": [],
            "id": "195db084-3715-4ebc-b164-3ee47a495b2e"
        }
\],
"id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

Project Tree (GET api.bimplus.net/v2/<team>/projects/<project_id>)

```
{
"name": "Your Awesome Building",
"shortDescr": "Updating the example project",
"created": "0001-01-01T00:00:00",
"changed": "0001-01-01T00:00:00",
"disciplines": \[{
            "id": "0f106af0-a919-44c5-b211-15bd5ef620b6",
            "name": "ID_BuildingModel",
            "divisionId": "ac21b9a1-8853-4658-9fbf-0cf62340bd18",
            "divisionName": null,
            "revisions": []
        }
\],
"id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

- Please use the Bimplus Object Service for obtaining/filtering the object topology tree. This will yield the topology nodes with all the properties, attributes, geometry and children.

Object Topology Tree (GET api.bimplus.net/v2/<team>/objects/<object_id>/topology)

```
{
    "parent": null,
    "name": "Updated_Brickwall",
    "type": "ID_ArchWall",
    "children": [],
    "id": "7cf910cb-e357-43f6-aa3a-b45c26380f75"
}
```

Object Tree (GET api.bimplus.net/v2/<team>/objects/<object_id>)

```
{
"id": "7cf910cb-e357-43f6-aa3a-b45c26380f75",
"type": "Wall",
"attributes": {
"element": {
            "isparent": true,
            "nr": 0,
            "matrix": "AAAAAAAA8D8AAAAAAAAAgAAAAAAAAACAINejcP1YsEAAAAAAAAAAgAAAAAAAAPA/AAAAAAAAAIAK16Nw/fy
                       2QAAAAAAAAACAAAAAAAAAAIAAAAAAAAADwPwAArkfheoS
/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8D8=",
            "elementtyp": "10074eef-9418-4d64-9c6d-23932835a7f1",
            "logparentid": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
            "divisionid": "ac21b9a1-8853-4658-9fbf-0cf62340bd18"
        },
"general": {
            "name": "Updated_Brickwall",
            "isobjectvalid": true
        },
"allplan": {
            "allplan_guid": "08b8195a-a2ad-11e2-9993-002215ea7d6b",
            "allplan_id": 2065507287,
            "allplan_text": "<PP>Wall",
            "dateofexport": "2013-05-01T09:32:13"
        },
"geometry": {
"mesh": "{\"C\":0,\"V\":[4450.01,115.01,0.01,3200.01,115.01,0.01,3200.01,0.01,0.01,4450.01,0.01,0.01,2315.01,
          0.01,0.01,2315.01,115.01,0.01,0.01,115.01,0.01,0.01,0.01,0.01,4450.01,115.01,2750.01,4450.01,0.01,
          2750.01,0.01,0.01,2750.01,0.01,115.01,2750.01,2315.01,115.01,2190.01,3200.01,115.01,2190.01,3200.01,
          0.01,2190.01,2315.01,0.01,2190.01],\"F\":[4,0,1,2,3,4,4,5,6,7,4,8,9,10,11,4,0,3,9,8,8,6,5,12,13,1,0,
          8,11,4,7,6,11,10,8,3,2,14,15,4,7,10,9,4,12,15,14,13,4,15,12,5,4,4,13,14,2,1]}"
}
}
}
```

- A project is basically an object. For a better understanding, here the same project is represented using the different API calls.

Project Tree (GET api.bimplus.net/v2/nemetschek/projects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e)

```
{
    "name": "Your Awesome Building",
    "shortDescr": "Updating the example project",
    "thumbnail": "/nemetschek/projects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e/thumbnail/download",
    "created": "0001-01-01T00:00:00",
    "changed": "0001-01-01T00:00:00",
    "disciplines": [
        {
            "id": "0f106af0-a919-44c5-b211-15bd5ef620b6",
            "name": "ID_BuildingModel",
            "divisionId": "ac21b9a1-8853-4658-9fbf-0cf62340bd18",
            "divisionName": null,
            "revisions": []
        }
    ],
    "id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

Project Topology Tree (GET api.bimplus.net/v2/nemetschek/projects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e/topology)

```
{
    "parent": null,
    "name": "Your Awesome Building",
    "type": "ID_Project",
    "children": [
        {
            "parent": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
            "name": "Your Awesome Building",
            "type": "ID_TopologyDivision",
            "children": [],
            "id": "c1c6c2bc-f38e-4c01-ae5a-864baaa0b2d0"
        }
    ],
    "id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

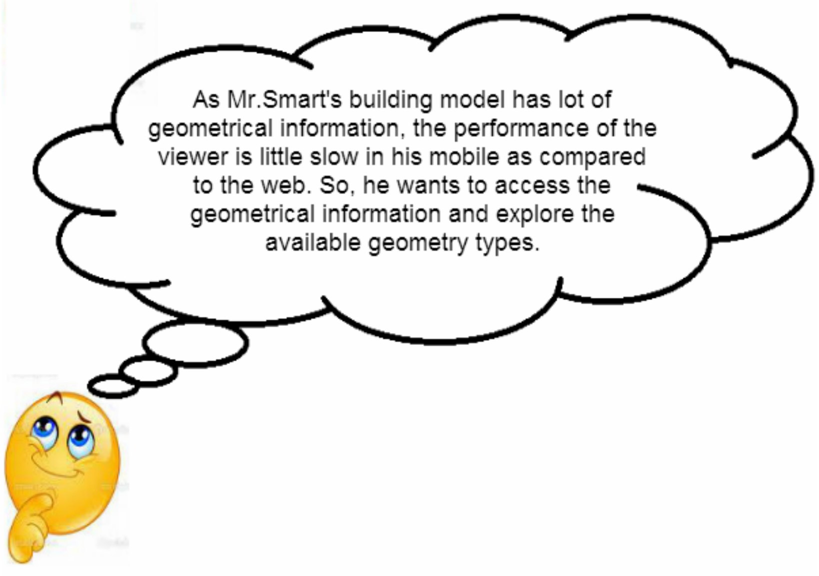
Object Tree (GET api.bimplus.net/v2/nemetschek/objects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e)

```
{
    "id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
    "type": "Project",
    "attributes": {
        "element": {
            "isparent": true,
            "elementtyp": "8d27ae6d-3c9a-4201-8a4d-bf0225861788",
            "materialsurfaceid": "cdbd7fbc-0d02-4be5-adb3-04b7f0b2a638"
        },
        "general": {
            "name": "Your Awesome Building",
            "description": "Updating the example project",
            "isobjectvalid": true
        }
    }
}
```

Object Topology Tree (GET api.bimplus.net/v2/nemetschek/objects/0d2ce2cc-3588-45d6-a59a-f6b89ebed34e/topology)

```
{
    "parent": null,
    "name": "Your Awesome Building",
    "type": "ID_Project",
    "children": [
        {
            "parent": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
            "name": "Your Awesome Building",
            "type": "ID_TopologyDivision",
            "children": [
                {
                    "parent": "c1c6c2bc-f38e-4c01-ae5a-864baaa0b2d0",
                    "name": "Brickwall",
                    "type": "ID_ArchWall",
                    "children": [],
                    "id": "7cf910cb-e357-43f6-aa3a-b45c26380f75"
                }
            ],
            "id": "c1c6c2bc-f38e-4c01-ae5a-864baaa0b2d0"
        }
    ],
    "id": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e"
}
```

Find the geometry of an object. Get to know the types of geometry available


As Mr.Smart's building model has lot of geometrical information, the performance of the viewer is little slow in his mobile as compared to the web. So, he wants to access the geometrical information and explore the available geometry types.

The geometrical information(graphical representation) of a building can be accessed through the Bimplus Object Service. There are different types of geometry available for optimizing and viewing on different devices.

There are three type of Geometry objects.

- **mesh**: It is the default geometry type. It is an uncompressed mesh format where the geometry object will be compressed on the server before storing it into the database. Please use this API call(GET https://api-stage.bimplus.net/v2//objects/) for getting the mesh information of an object.

Get the mesh data from the object tree (GET api.bimplus.net/v2/<team>/objects/<object_id>)

```
{
"id": "7cf910cb-e357-43f6-aa3a-b45c26380f75",
"type": "Wall",
"attributes": {
"element": {
            "isparent": true,
            "nr": 0,
            "matrix": "AAAAAAAA8D8AAAAAAAAAgAAAAAAAAACAINejcP1YsEAAAAAAAAAAAgAAAAAAAPA/AAAAAAAAAIAK16Nw/fy
                       2QAAAAAAAAACAAAAAAAAAAIAAAAAAAAADwPwAArkfheoS
/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8D8=",
            "elementtyp": "10074eef-9418-4d64-9c6d-23932835a7f1",
            "logparentid": "0d2ce2cc-3588-45d6-a59a-f6b89ebed34e",
            "divisionid": "ac21b9a1-8853-4658-9fbf-0cf62340bd18"
        },
"general": {
            "name": "Updated_Brickwall",
            "isobjectvalid": true
        },
"allplan": {
            "allplan_guid": "08b8195a-a2ad-11e2-9993-002215ea7d6b",
            "allplan_id": 2065507287,
            "allplan_text": "<PP>Wall",
            "dateofexport": "2013-05-01T09:32:13"
        },
"geometry": {
"mesh": "{\"C\":0,\"V\":[4450.01,115.01,0.01,3200.01,115.01,0.01,3200.01,0.01,0.01,4450.01,0.01,0.01,2315.01,
          0.01,0.01,2315.01,115.01,0.01,0.01,115.01,0.01,0.01,0.01,0.01,4450.01,115.01,2750.01,4450.01,0.01,
          2750.01,0.01,0.01,2750.01,0.01,115.01,2750.01,2315.01,115.01,2190.01,3200.01,115.01,2190.01,3200.01,
          0.01,2190.01,2315.01,0.01,2190.01],\"F\":[4,0,1,2,3,4,4,5,6,7,4,8,9,10,11,4,0,3,9,8,8,6,5,12,13,1,0,
          8,11,4,7,6,11,10,8,3,2,14,15,4,7,10,9,4,12,15,14,13,4,15,12,5,4,4,13,14,2,1]}"
}
}
}
```

- **meshblob**: It is a compressed mesh format intended to be used in the desktop CAD applications. Internal CAD format of the geometry will be converted into a Bimplus compressed mesh format with the help of a DLL in order to minimize the JSON object. Please use this API call(GET https://api-stage.bimplus.net/v2//objects//geometries/meshblob) for getting the meshblob information of an object.

Meshblob data (GET api.bimplus.net/v2/<team>/objects/<object_id>/geometries/meshblob)

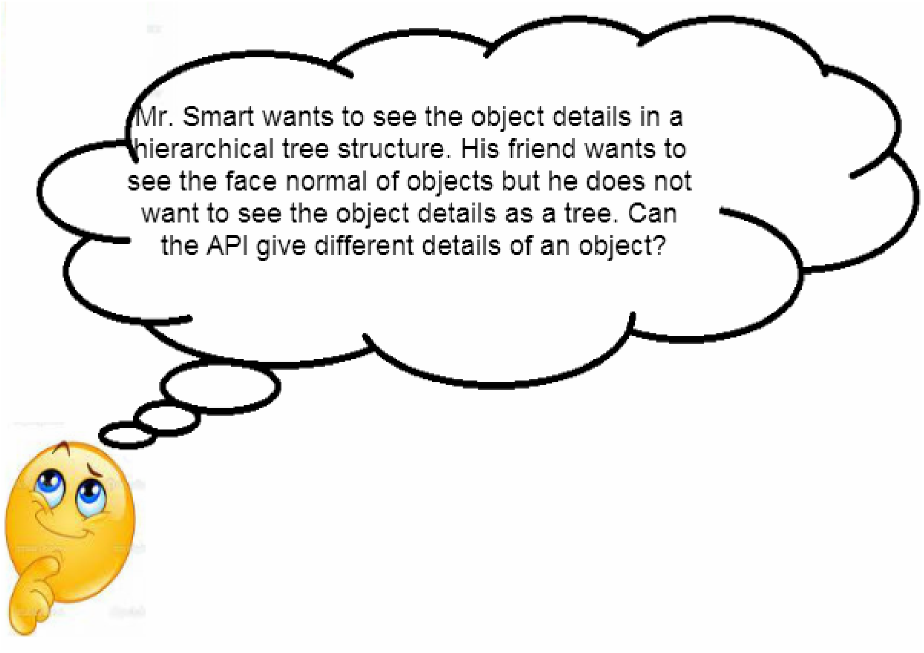
```
{
    "elementsCount": 1,
    "objects": [
        {
            "id": "7cf910cb-e357-43f6-aa3a-b45c26380f75",
            "parent": "00000000-0000-0000-0000-000000000000",
            "type": "Wall",
            "attributes": {
                "geometry": {
                    "compress": true,
                    "picture":
"H4sIAAAAAAAEANVXW3PbRBTWylIsuUmctml6T900t7aua+eetiHOndAQDLS8pBnhyzYVUeQiyR0cGJ545y9QGB
                                54YYY3HvkXPPetQLlDuV/2E7uqiEfUMzRDkWc
/zfl0tHvO2XPOWhKRJOkPduGOq1VmcLVQmPE8xywVzLJXc+iV+k06nE69QB3Xr

NpTQ9lcLpPNjExmxzIjufHRbDo1V7OgOGXTmucUrXSqUCtZZvkyrV+pblJ7yq5Zloo10qt0i3pu+QbdzKxe9UxrxSw5RaeemZstunT

BYk9tr1C16rRCnTh7Qb1ZNW1PoZUNqlwvlqm2ZTxXrJg1t33LmKV2xbQ3nKplUcVjRsrL87FYTFGU2G3yfN316FZmDg/LHrPbzSxRm

zpmObNiut6LubW1SFsCO9jSzJfdiMb6+luPwsQFFpfH28JFtmm7ZmFP5OrVmu3NOE6xjoRuUq03Sk0kJMyG4h4el6WaWYGsxwAKQGWQ
                                SKg/s5pqajoURfxW0apRw5A0TBbrkCTlJ
/Z+eJU97IFsFGWjJBtl2ajIBpWN67KxIRs3ZMOUjZdkY1N6cGnxuMwvacelwNj/QX3AzBb
                                DZEa6quGa21Qzbv01Jwv0QFO2ra3De03TNXbr4EPBPj3+xfeP7vc3Y1rgvc5uh/hQ1Efl/a4W9r
/zHqYF3idQkXzEkQgSEZnQ11QW+R

WOAtRbAW2AdkASgIn0vYB9gP2ATsABQBfgIADB1w8DjsAQXRiSgiGRrSK8n74dRwHHAMcB3YATAMyjnwT0AE4BegF9gH7AAGAQcBpwB

nAWkAacA2QA5wFZQA4wxKBNiyf+FrqHWow98C0eBowARgFjgHHABGAScAFwEXAJa7WoiHZzm4NEIa+QOtlGxrS0YMn7g9felEsf5F+e
                                Snz8zkfX8q92vr90Z/uNaYL9w9T3nv2wW5Xea1Ro26EgHgQK7VxBLNGgkAzP0PV2o0LHDoUGG
/ZyBUE0KOzbodCwxP4dRooZ/CVfezd

POiO8CBQORCwRKHRFGBkoHIxw05eP3c6TQxF7ESgcjtiLQOFIRKiFgopaabqy1BLdMO0Y+0uJriHzXPfvKLejXPDz7TgnIOA0Id2cUPh
                                r5AQn0G1bQKQ4AQH
/a8lJTkBAVyI9nND4a+QUJ9AqUHyklxMJ7jjp4wQE5Dbp50Qrf40MhAzz5xgMTepbejpkGNKfnOEEBKQ7OcuJdhG
                                PdGgV35dzITt8XzKhAKEiyHlOQEAFkCwnOoS3udCyvsZQ6BUsrQ43s51oO
/BLxReDG3u9iYOOqyzb3vBQOrXllquOZZYeHGIj7ARjv4c
                                fXaXx8eJoeXQsNzk8QrMTk+v+EaRPwRs0QL8TPgFpVEjTkMaElIc0LqQZSBNCmoU0KaQ5SBeENA
/popAWIF0S0iJ2Hyb8x6GIPtHjKDh

NX+Jp6tcQogSj9SfD7LRgl8NsXrBPhdkZwV7mSekn5qxgV8K6c4J9Oqw7L9jVsO6CYJ8Js4uCLYTYpHBJE61ENIykcEsTDUK0gaRwTRN
                                lL4o7KdzTfmUfByiTH9gdlZEMXMRnhygxnCP4tP+Gje/YwLGRFG5r+MD4go3v2cB5kQxc
/40xOCBwBqDN/8jkz9m4z8a3mF+EQ0OP/4w
                                xaOVf4YkIifYJk75m4xc2PsUTERbtLpPusfE7G1/iI+dPBQ0gCIMQAAA=",
                    "type": 3,
                    "matrix": "AAAAAAAA8D8AAAAAAAAAgAAAAAAAAACAINejcP1YsEAAAAAAAAAAgAAAAAAAAPA/AAAAAAAAAIAK16Nw
/fy2QAAAAAAAAACAAAAAAAAAAIAAAAAAAA
                              DwPwAArkfheoS/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8D8="
                }
            }
        }
    ]
}
```

- **threejs**: It is a format which is optimized for viewing on mobile devices or in browsers where the native model is filtered and tessellated in order to deliver minimal JSON size and maximal navigation speed (FPS). This model has ThreeJS JSON v 3.1 format and can be directly parsed by the client.Please use this API call(GET https://api-stage.bimplus.net/v2//objects//geometries/threejs) for getting the threejs information of an object.

ThreeJS data (GET api.bimplus.net/v2/<team>/objects/<object_id>/geometries/threejs)

```
{
    "elementsCount": 1,
    "viewbox": {
        "x": -5081.562,
        "y": 5005.8785,
            "z": 1231.1029999999996,
            "width": 5950,
            "height": 5540.835,
            "depth": 7065.71
        },
        "colors": [
            -9868951
        ],
        "objects": [
            {
                "id": "f5cc3c6c-1fdf-499d-a38d-0582adadc216",
                "type": "RoofLayer",
                "attributes": {
                    "geometry": {
                        "threejs": {
                            "faces": [
                                0,
                                2,
                                1,
                                3,
                                0,
                                5,
                                3,
                                9,
                                1,
                                1,
                                14,
                                1,
                                5,
                                6,
                                15,
                                12,
                                1,
                                7,
                                4,
                                13,
                                14
                            ],
                            "vertices": [
                                -2106.562,
                                7588.114,
                                -2301.752,
                                -8056.562,
                                7588.114,
                                -2301.752,
                                -8056.562,
                                -880.999,
                                -4742.212,
                                6700,
                                -880.999,
                                -4742.212,
                                7412.539,
                                -1821.578,
                                -5522.212,
                                7412.539,
                                -1821.578
                            ],
                            "metadata": {
                                "formatVersion": 3,
                                "colorid": 0
                            }
                        }
                    },
                    "quantity": {
                        "length": 7065.71,
                        "height": 150,
                        "volume": 7911354818.818842,
                        "area": 52742365.4587924
                    },
                    "general": {
                        "name": "<PP>RoofLayer"
                }
            }
        }
    ]
}
```

Get the different properties / structure(normal, viewer properties, object in hierarchical tree structure) of the object using different property combinations


Mr. Smart wants to see the object details in a hierarchical tree structure. His friend wants to see the face normal of objects but he does not want to see the object details as a tree. Can the API give different details of an object?

Different properties / structure (normal, viewer properties, object in hierarchical tree structure) of the object can be obtained using different property combinations.

url: /geometries/threejs/properties/{props}

where props can be 1, 2 or 3.
/geometries/threejs/properties/1 : show objects as hierarchical tree (default (/geometries/threejs or /geometries/threejs/properties/0 will give the object details in a flat structure))
/geometries/threejs/properties/2 : show also viewer attributes
/geometries/threejs/properties/3 : include FACE_NORMAL

You can also combine different property numbers

eg: geometries/threejs/properties/4 should give both the show objects as tree(1) + add FACE_NORMAL in threeJs json(3)

HTTP Method

GET

URL / Resource / JSON Structure

Resource: objects/<object_id>/geometries/threejs/properties/{props}

URL: https://api-stage.bimplus.net/v2/<team_slug>/objects/<object_id>/geometries/threejs/properties/{props}

Example: https://api-stage.bimplus.net/v2/bimplus/objects/a17ff78a-8b42-4523-9bf7-283e725e3e95/geometries/threejs/properties/4

Description

Get the different properties / structure (normal, viewer properties, object in hierarchical tree structure) of the object using different property combinations.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

Response

Status

```
Status: 200 OK
```

JSON

```
{
    "elementsCount": 1,
    "faceCount": 88,
    "viewbox": {
        "x": 21000,
        "y": 2850,
        "z": -19000,
        "width": 133,
        "height": 5700,
        "depth": 140
    },
    "colors": [
        4278190080
    ],
    "objects": [
        {
            "id": "a17ff78a-8b42-4523-9bf7-283e725e3e95",
            "type": "Column",
            "attributes": {
                "geometry": {
                    "threejs": {
                        "faces": [
                            0,
                            2,
                            1,
                            0,
                            0,
                            4,
                            0,
                            3,
                            0,
                            7,
                            6,
                            5,
                            0,
                            8,
                            5,
                            6,
                            0,
                            9,
                            5,
                            8,
                            0,
                            10,
                            9,
                            8,
                            0,
                            13,
                            12,
                            11,
                              0,
                              16,
                              15,
                              14,
                              0,
                              17,
                              14,
                              15,
                              0,
                              18,
                              14,
                              17,
                              0,
                              19,
                              18,
                              17,
                              0,
                              7,
                              15,
                              6,
                              0,
                              16,
                              6,
                              15,
                              0,
                              7,
                              5,
                              15,
                              0,
                              17,
                              15,
                              5,
                              0,
                              17,
                              5,
                              19,
                              0,
                              9,
                              19,
                              5,
                              0,
                              2,
                              11,
                              1,
                              0,
                              2,
                              0,
                              11,
                              0,
                              13,
                              11,
                              0,
                              0,
                              13,
                              0,
                              12,
                              0,
                              4,
                              12,
                              0,
                              0,
                              4,
                              3,
                              12,
                              0,
                              20,
                              12,
                              3,
                              0,
                              18,
                              10,
14,
0,
8,
14,
10,
0,
8,
6,
14,
0,
16,
14,
6,
0,
0,
1,
21,
0,
21,
3,
0,
0,
1,
11,
22,
0,
22,
11,
12,
0,
12,
20,
23,
0,
3,
21,
23,
0,
23,
22,
12,
0,
23,
20,
3,
0,
26,
25,
24,
0,
24,
27,
26,
0,
25,
26,
28,
0,
31,
30,
29,
0,
32,
31,
29,
0,
35,
34,
33,
0,
32,
29,
36,
0,
38,
36,
37,
0,
36,
38,
32,
0,
39,
10,
18,
0,
24,
41,
40,
0,
42,
19,
9,
0,
43,
30,
37,
0,
36,
26,
27,
0,
37,
27,
43,
0,
27,
37,
36,
0,
26,
36,
29,
0,
30,
43,
28,
0,
29,
28,
26,
0,
28,
29,
30,
0,
24,
40,
43,
0,
43,
27,
24,
0,
33,
34,
39,
0,
39,
34,
44,
0,
45,
34,
35,
0,
38,
23,
21,
0,
21,
32,
38,
0,
21,
1,
31,
0,
31,
1,
22,
0,
31,
32,
21,
0,
45,
35,
25,
0,
24,
25,
35,
0,
41,
24,
35,
0,
33,
39,
18,
0,
42,
33,
18,
0,
18,
19,
42,
0,
45,
40,
44,
0,
44,
34,
45,
0,
44,
40,
41,
0,
35,
33,
42,
0,
41,
42,
44,
0,
42,
41,
35,
    0,
    38,
    37,
    22,
    0,
    31,
    22,
    37,
    0,
    22,
    23,
    38,
    0,
    37,
    30,
    31,
    0,
    44,
    42,
    9,
    0,
    39,
    44,
    9,
    0,
    9,
    10,
    39,
    0,
    45,
    25,
    28,
    0,
    40,
    45,
    28,
    0,
    28,
    43,
    40
],
"vertices": [
    21066.5,
    0,
    -18930,
    21058,
    0,
    -18985.2,
    21058,
    0,
    -18930,
    21058,
    0,
    -19070,
    21066.5,
    0,
    -19070,
    20942,
    0,
    -18930,
    20933.5,
    0,
    -19070,
    20933.5,
    0,
    -18930,
    20942,
    0,
    -19070,
    20942,
    0,
-18985.2,
20942,
0,
-19014.8,
21058,
5700,
-18930,
21066.5,
5700,
-19070,
21066.5,
5700,
-18930,
20942,
5700,
-19070,
20933.5,
5700,
-18930,
20933.5,
5700,
-19070,
20942,
5700,
-18930,
20942,
5700,
-19014.8,
20942,
5700,
-18985.2,
21058,
5700,
-19070,
21057.8,
0,
-19012.4,
21057.8,
5700,
-18987.6,
21057.8,
5700,
-19012.4,
20954,
5700,
-18997.2,
20954,
5700,
-19002.8,
21048.3,
5700,
-19003,
21048.3,
5700,
-18997,
21048.3,
0,
-19003,
21052.7,
0,
-19004.8,
21052.7,
0,
-18995.2,
21056,
0,
-18991.9,
21056,
0,
-19008.1,
20944,
                            5700,
                            -19008.1,
                            20945.5,
                            0,
                            -19006.3,
                            20947.3,
                            5700,
                            -19004.8,
                            21052.7,
                            5700,
                            -19004.8,
                            21052.7,
                            5700,
                            -18995.2,
                            21056,
                            5700,
                            -19008.1,
                            20942.9,
                            0,
                            -19010.2,
                            20951.7,
                            0,
                            -18997,
                            20949.4,
                            5700,
                            -18996.3,
                            20942.9,
                            5700,
                            -18989.8,
                            21048.3,
                            0,
                            -18997,
                            20945.5,
                            0,
                            -18993.7,
                            20949.4,
                            0,
                            -19003.7
                        ],
                        "metadata": {
                            "formatVersion": 3,
                            "colorid": 0
                        }
                    }
                },
                "related_objects": {
                    "0": {
                        "id": "4f106618-6110-43c1-93ce-c542417cf5d6",
                        "type": "GeometryObject"
                    }
                }
            }
        }
    ]
}
```

geometries/threejs/properties/7 should give you a combination of objects as tree(1) + viewer attributes(2) + face normal(3)

HTTP Method

GET

URL / Resource / JSON Structure

Resource: objects/<object_id>/geometries/threejs/properties/{props}

URL: https://api-stage.bimplus.net/v2/<team_slug>/objects/<object_id>/geometries/threejs/properties/{props}

Example: https://api-stage.bimplus.net/v2/bimplus/objects/a17ff78a-8b42-4523-9bf7-283e725e3e95/geometries/threejs/properties/7 (i.e 7 gives you a combination of objects as tree(1) + viewer attributes(2) + face normal(3))

Description

Get the different properties / structure (normal, viewer properties, object in hierarchical tree structure) of the object using different property combinations.

Request

Headers

```
Authorization: BimPlus 9c1874a62c974dcfa75e0132c423a088
Content-Type: application/json
```

Response

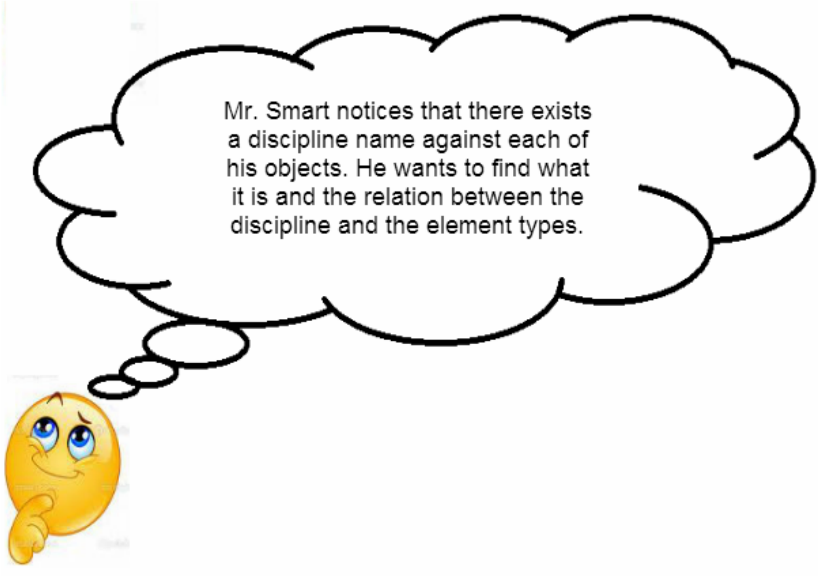
Status

```
Status: 200 OK
```

JSON

```
{
    "elementsCount": 1,
    "faceCount": 88,
    "viewbox": {
        "x": 21000,
        "y": 2850,
        "z": -19000,
        "width": 133,
        "height": 5700,
        "depth": 140
    },
    "colors": [
        4278190080
    ],
    "objects": [
        {
            "id": "a17ff78a-8b42-4523-9bf7-283e725e3e95",
            "type": "Column",
            "attributes": {
                "geometry": {
                    "threejs": {
                        "faces": [
                            0,
                            2,
                            1,
                            0,
                            0,
                            4,
                            0,
                            3,
                            0,
                            7,
                            6,
                            5,
                            0,
                            8,
                            5,
                            6,
                            0,
                            9,
                            5,
                            8,
                            0,
                            10,
9,
8,
0,
13,
12,
11,
0,
16,
15,
14,
0,
17,
14,
15,
0,
18,
14,
17,
0,
19,
18,
17,
0,
7,
15,
6,
0,
16,
6,
15,
0,
7,
5,
15,
0,
17,
15,
5,
0,
17,
5,
19,
0,
9,
19,
5,
0,
2,
11,
1,
0,
2,
0,
11,
0,
13,
11,
0,
0,
13,
0,
12,
0,
4,
12,
0,
0,
4,
3,
12,
0,
20,
12,
3,
0,
18,
10,
14,
0,
8,
14,
10,
0,
8,
6,
14,
0,
16,
14,
6,
0,
0,
1,
21,
0,
21,
3,
0,
0,
1,
11,
22,
0,
22,
11,
12,
0,
12,
20,
23,
0,
3,
21,
23,
0,
23,
22,
12,
0,
23,
20,
3,
0,
26,
25,
24,
0,
24,
27,
26,
0,
25,
26,
28,
0,
31,
30,
29,
0,
32,
31,
29,
            0,
            35,
            34,
            33,
            0,
            32,
            29,
            36,
            0,
            38,
            36,
            37,
            0,
            36,
            38,
            32,
            0,
            39,
            10,
            18,
            0,
            24,
            41,
            40,
            0,
            42,
            19,
            9,
            0,
            43,
            30,
            37,
            0,
            36,
            26,
            27,
            0,
            37,
            27,
            43,
            0,
            27,
            37,
            36,
            0,
            26,
            36,
            29,
            0,
            30,
            43,
            28,
            0,
            29,
            28,
            26,
            0,
            28,
            29,
            30,
            0,
            24,
            40,
            43,
            0,
            43,
            27,
            24,
            0,
            33,
            34,
                            39,
                            0,
                            39,
                            34,
                            44,
                            0,
                            45,
                            34,
                            35,
                            0,
                            38,
                            23,
                            21,
                            0,
                            21,
                            32,
                            38,
                            0,
                            21,
                            1,
                            31,
                            0,
                            31,
                            1,
                            22,
                            0,
                            31,
                            32,
                            21,
                            0,
                            45,
                            35,
                            25,
                            0,
                            24,
                            25,
                            35,
                            0,
                            41,
                            24,
                            35,
                            0,
                            33,
                            39,
                            18,
                            0,
                            42,
                            33,
                            18,
                            0,
                            18,
                            19,
                            42,
                            0,
                            45,
                            40,
                            44,
                            0,
                            44,
                            34,
                            45,
                            0,
                            44,
                            40,
                            41,
                            0,
                            35,
                            33,
                            42,
                            0,
                            41,
                42,
                44,
                0,
                42,
                41,
                35,
                0,
                38,
                37,
                22,
                0,
                31,
                22,
                37,
                0,
                22,
                23,
                38,
                0,
                37,
                30,
                31,
                0,
                44,
                42,
                9,
                0,
                39,
                44,
                9,
                0,
                9,
                10,
                39,
                0,
                45,
                25,
                28,
                0,
                40,
                45,
                28,
                0,
                28,
                43,
                40
            ],
            "vertices": [
                21066.5,
                0,
                -18930,
                21058,
                0,
                -18985.2,
                21058,
                0,
                -18930,
                21058,
                0,
                -19070,
                21066.5,
                0,
                -19070,
                20942,
                0,
                -18930,
                20933.5,
                0,
                -19070,
                20933.5,
                0,
-18930,
20942,
0,
-19070,
20942,
0,
-18985.2,
20942,
0,
-19014.8,
21058,
5700,
-18930,
21066.5,
5700,
-19070,
21066.5,
5700,
-18930,
20942,
5700,
-19070,
20933.5,
5700,
-18930,
20933.5,
5700,
-19070,
20942,
5700,
-18930,
20942,
5700,
-19014.8,
20942,
5700,
-18985.2,
21058,
5700,
-19070,
21057.8,
0,
-19012.4,
21057.8,
5700,
-18987.6,
21057.8,
5700,
-19012.4,
20954,
5700,
-18997.2,
20954,
5700,
-19002.8,
21048.3,
5700,
-19003,
21048.3,
5700,
-18997,
21048.3,
0,
-19003,
21052.7,
0,
-19004.8,
21052.7,
0,
-18995.2,
21056,
                            0,
                            -18991.9,
                            21056,
                            0,
                            -19008.1,
                            20944,
                            5700,
                            -19008.1,
                            20945.5,
                            0,
                            -19006.3,
                            20947.3,
                            5700,
                            -19004.8,
                            21052.7,
                            5700,
                            -19004.8,
                            21052.7,
                            5700,
                            -18995.2,
                            21056,
                            5700,
                            -19008.1,
                            20942.9,
                            0,
                            -19010.2,
                            20951.7,
                            0,
                            -18997,
                            20949.4,
                            5700,
                            -18996.3,
                            20942.9,
                            5700,
                            -18989.8,
                            21048.3,
                            0,
                            -18997,
                            20945.5,
                            0,
                            -18993.7,
                            20949.4,
                            0,
                            -19003.7
                        ],
                        "metadata": {
                            "formatVersion": 3,
                            "colorid": 0
                        }
                    }
                },
                "general": {
                    "name": "B132",
                    "description": "Ifc Imported Element"
                },
                "element": {
                    "elementtyp": "Column",
                    "layer": "BuildingModel",
                    "model": "Structural model",
                    "material": "S 235"
                },
                "elementstate": {
                    "state": "Nothing Defined"
                },
                "allfreeattrib": {
                    "globalid": "26gojRHan8_9KtPcvJkUQf",
                    "skipparentlocations": "False",
                    "extrudeprofilename": "HEA140",
                    "extrudeprofiletype": "IShapeProfile",
                    "filletradius": "12",
                    "flangethickness": "8,5",
                    "overalldepth": "133",
                    "overallwidth": "140",
                    "webthickness": "5,5"
                }
            }
        }
    ]
}
```

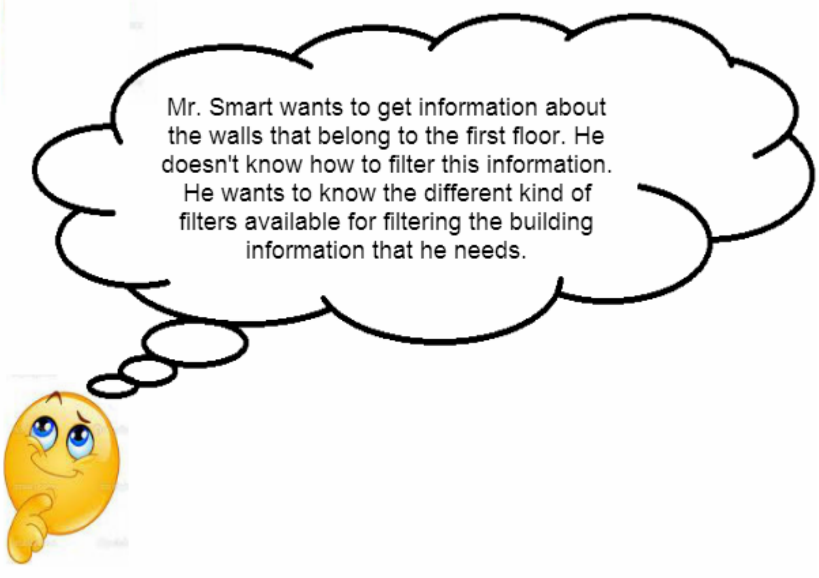
Get the project layers and element types. Find the relation between them


Mr. Smart notices that there exists a discipline name against each of his objects. He wants to find what it is and the relation between the discipline and the element types.

A group of element types constitute a layer. But an element type cannot belong to two layers. Both the element types and layers are predefined.

- Please see the list of element types here
- The list of element types can be accessed through this API call(GET https://api-stage.bimplus.net/v2//element-types)
- The list of layers can be accessed through this API call(GET https://api-stage.bimplus.net/v2//element-types/disciplines)

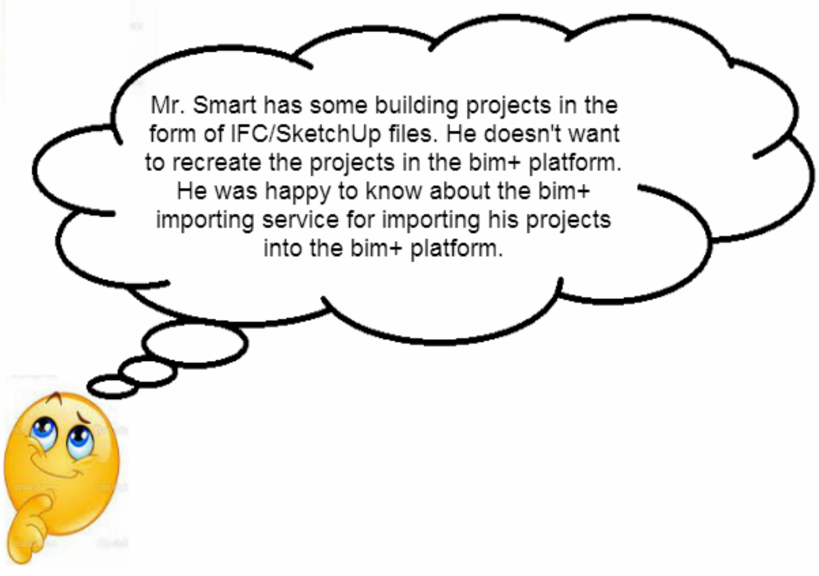
Filter the building information using different filter criteria


Mr. Smart wants to get information about the walls that belong to the first floor. He doesn't know how to filter this information. He wants to know the different kind of filters available for filtering the building information that he needs.

The "ThreeJS" or the "meshblob" information can be filtered by the layer.

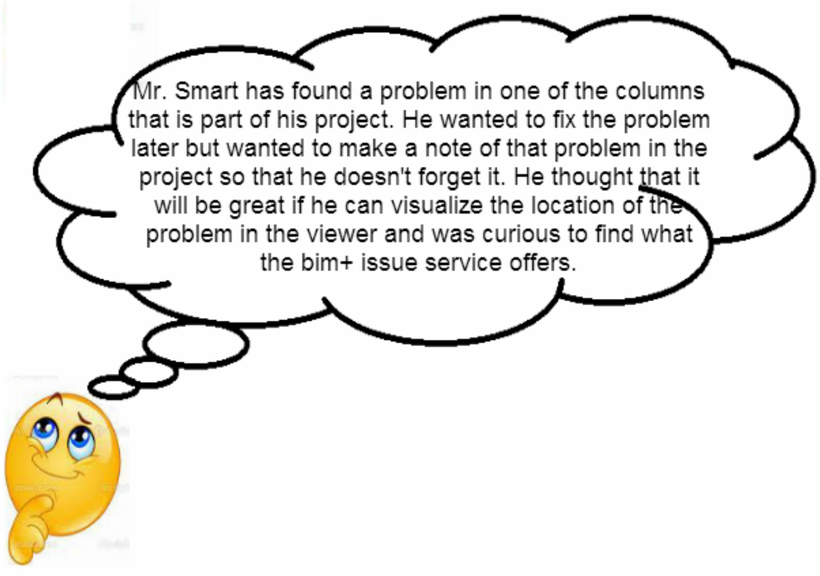
- Please use this API call (GET https://api-stage.bimplus.net/v2/<team>/objects/<object_id>/disciplines/<discipline_id>/geometries/threejs) to get the object tree with selected property list whose geometry type is "ThreeJS" filtered by layer
- Please use this API call (GET https://api-stage.bimplus.net/v2/<team>/objects/<object_id>/disciplines/<discipline_id>/geometries/meshblob) to get the object tree whose geometry type is "compressed geometry mesh (meshblob)" filtered by layer

Import an existing model in the form of IFC/SketchUp file


Mr. Smart has some building projects in the form of IFC/SketchUp files. He doesn't want to recreate the projects in the bim+ platform. He was happy to know about the bim+ importing service for importing his projects into the bim+ platform.

Projects in the form of IFC/SketchUp files can be imported into the Bimplus platform by using the Bimplus Import Service

Create and assign issues to projects. Locate/visualize the issues in the project


Mr. Smart has found a problem in one of the columns that is part of his project. He wanted to fix the problem later but wanted to make a note of that problem in the project so that he doesn't forget it. He thought that it will be great if he can visualize the location of the problem in the viewer and was curious to find what the bim+ issue service offers.

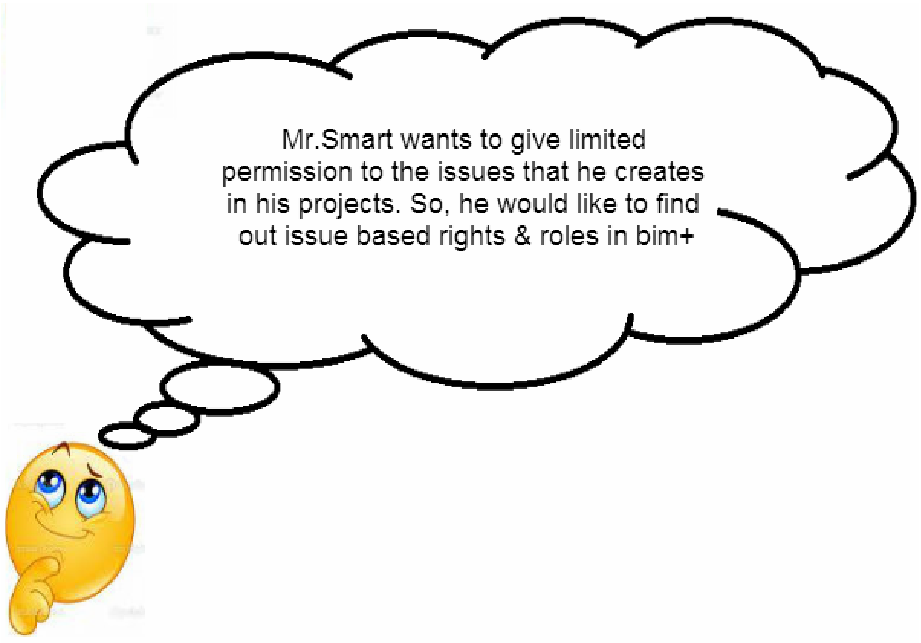
Any problems or remarks can be reported and assigned to a project in the form of issues. To understand the problem better, comments or attachments can be added to the issue.

- Please use the Bimplus Project Service for creating/deleting an issue or for getting all the issues assigned to a project
- Please use the Bimplus Issue Service for doing all the specific issue related actions

Pins are used to visualize the issues. Pins are relation between an issue and an object. It defines position of an issue in the 3D space.

- A Pin can be created by using the Bimplus Issue Service (POST https://api-stage.bimplus.net/v2//issues//pins)
- There is also a Bimplus PIN Service available for doing all the specific pin related actions.

Learn about the issue based rights & roles

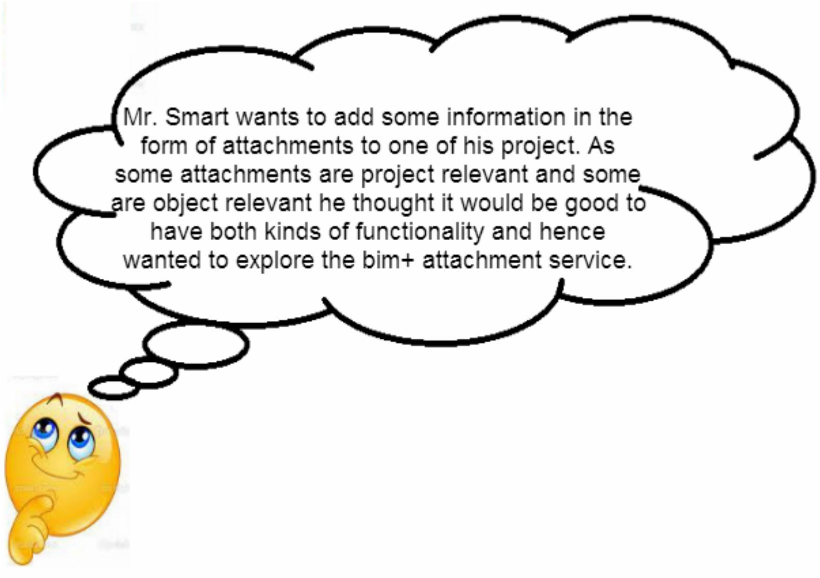

Mr.Smart wants to give limited permission to the issues that he creates in his projects. So, he would like to find out issue based rights & roles in bim+

Currently Bimplus has a couple of issue based roles (i.e Author, Responsible). Author is the user who creates an issue and the responsible is the one who solves it. Here are some of the conditions that apply for the Bimplus issue tracking.

- An user can create an issue only if he has the rights for editing a project. See the project based rights & roles here
- Issue details can be changed only by the author. An exception to it is the "solution" property of the issue where both the author and responsible can change this property.
- Only the author of the issue can set the issue "Status" to be "Closed". Currently there are three defined issue status: Status, Open & Solved
- Issues can be deleted only by the author of the respective issue
- An email will be sent to the author/responsible/cc when an issue is created/updated/deleted (not implemented)

- Please use the Bimplus Project Service for creating/deleting an issue or for getting all the issues assigned to a project
- Please use the Bimplus Issue Service for doing all the specific issue related actions

Add attachments to the whole building as well as to the individual objects


Mr. Smart wants to add some information in the form of attachments to one of his project. As some attachments are project relevant and some are object relevant he thought it would be good to have both kinds of functionality and hence wanted to explore the bim+ attachment service.

The pictures or any other documents(pdf, ifc etc) can be attached to both the project as well as the individual objects.

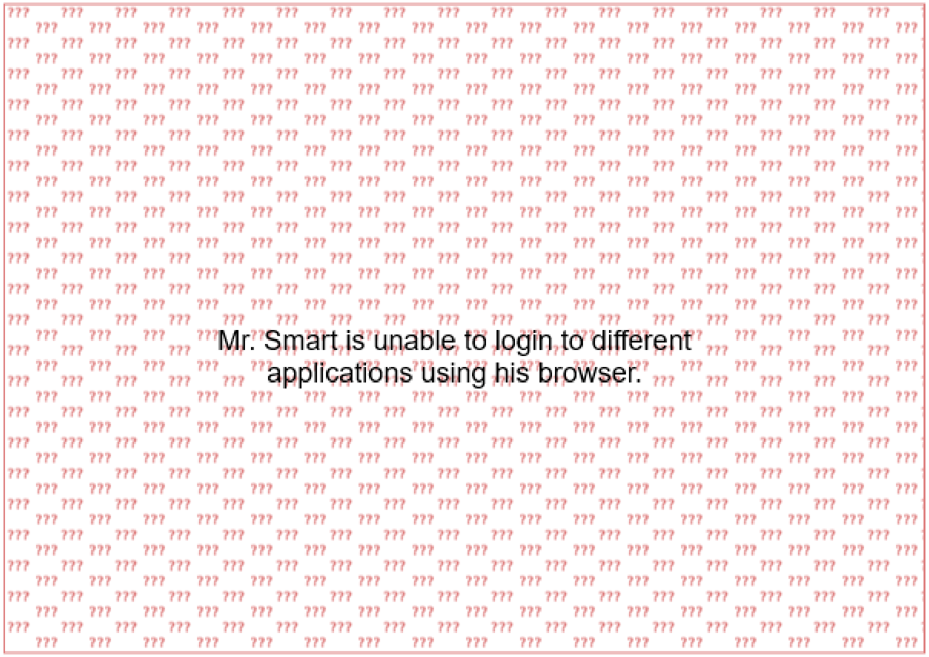
- Please use the Bimplus Project Service (POST https://api-stage.bimplus.net/v2/projects/attachments) for adding an attachment to the project.
- Verify whether the picture has been attached to the project using the Bimplus Project Service (GET https://api-stage.bimplus.net/v2/projects /attachments)
- Please use the Bimplus Object Service (POST https://api-stage.bimplus.net/v2/objects/attachments) for adding an attachment to the object.
- Verify whether the picture has been attached to the object using the Bimplus Object Service (GET https://api-stage.bimplus.net/v2/objects /attachments)

Find out how many teams I am part of. Remove myself from some of the teams

Mr. Smart has successfully completed the tasks of some projects. Now, he has got some new assignments and hence part of some new teams. But, in the viewer he sees the old projects and hence wants to remove himself from some of the teams.

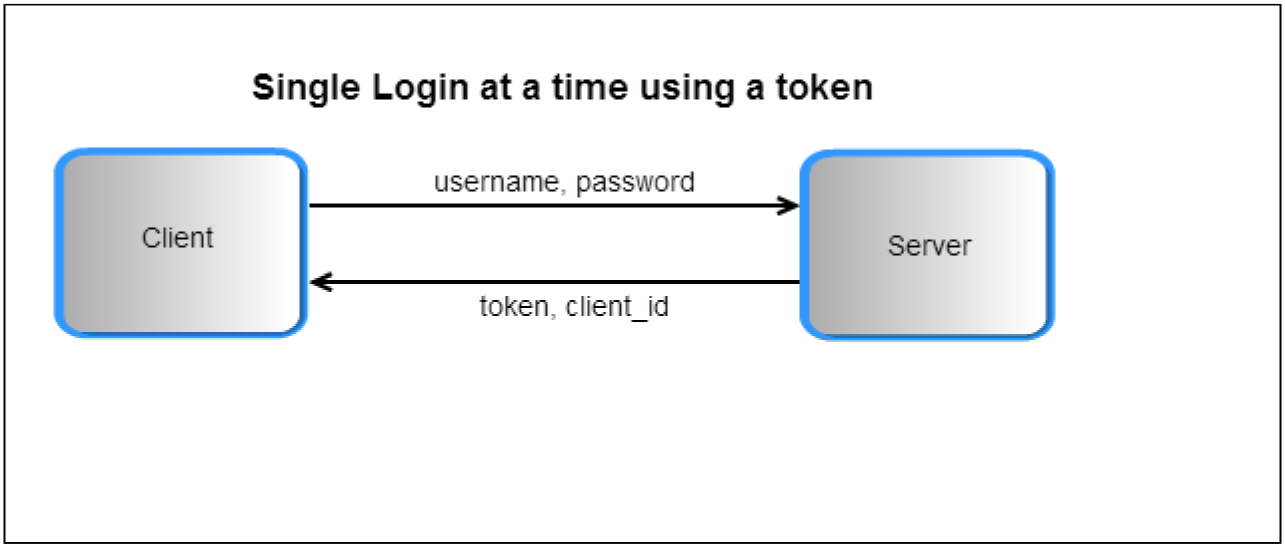
- Please use the Bimplus Team Management Service (GET https://api-stage.bimplus.net/v2/teams) to get the team list in which you are member of.
- After getting the team list, use the Bimplus Membership Management Service (DELETE https://api-stage.bimplus.net/v2//members/) to delete yourself from some of the teams that you don't want to be part of.

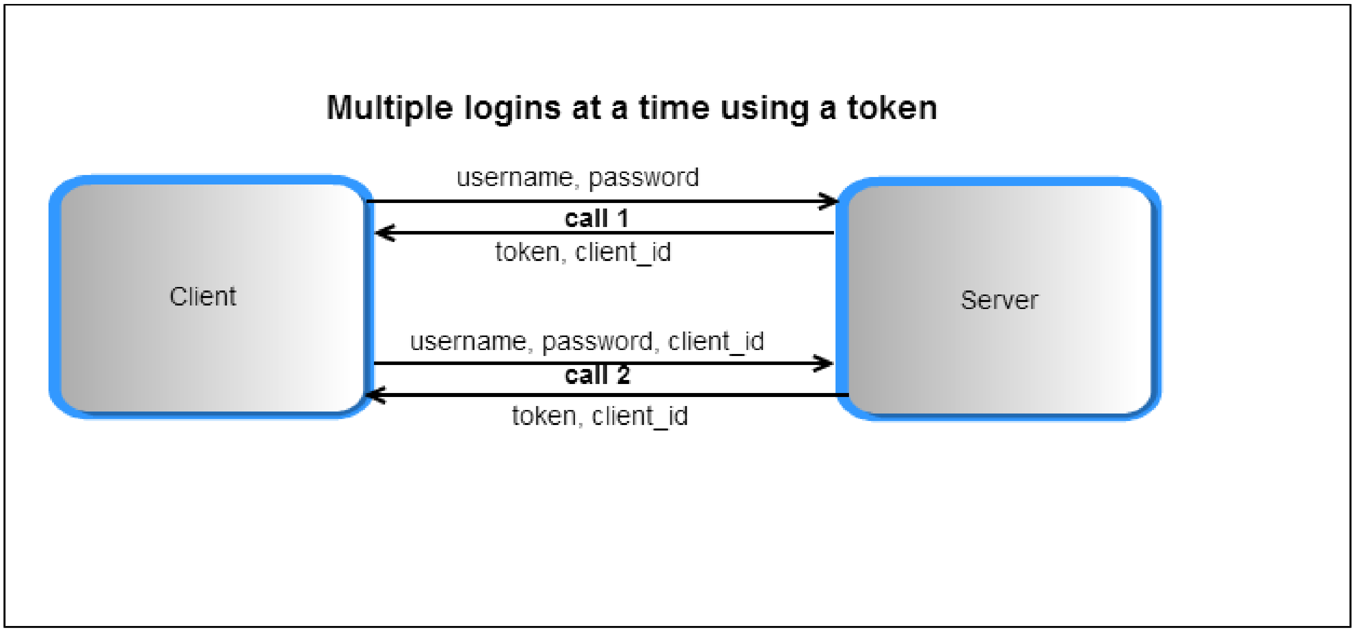
Authentication token for multiple services


Mr. Smart is unable to login to different applications using his browser.

- Please use the client_id as part of the authentication request. The client_id is a name of another application ,that user wants to use additionally. When user authenticate to another service using SSO, it opens browser UI login page again, it uses cookies and issues token without re-entering credentials. A token will be generated based on the new client_id and cookies which will allow single-signon logins at the same time and increase security because credentials are not entered and transferred many times and the token is only valid for this device and client (Browser, Mobile App).

Obsolete example using internal authentication service, deprecated in 2026:





HTTP Method

POST

URL / Resource / JSON Structure

Resource: authorize

URL: https://api-stage.bimplus.net/v2/authorize

JSON Structure:

Name	Mandatory / Optional	Type	Description
user_id	mandatory	string	The email address of the user
password	mandatory	string	The user's password
client_id	optional	string	The identifier of the used client
application_id	optional	string	The id of the application

Description

Authenticate by requesting and receiving a token.

Request

Headers

```
Content-Type: application/json
```

JSON

```
{
 "user_id" : "test@bimplus.net",
 "password" : "test"
}
```

Response

Status

```
Status: 200 OK
```

JSON

```
{
access_token: "e3271e89d8ea474c82745039bc2bed0d"
expires_in: 2591999
client_id: "9fd0bb9d-570b-4719-bfae-93e2f879c19a"
token_type: "BimPlus"
}
```

Use the client_id "9fd0bb9d-570b-4719-bfae-93e2f879c19a" to generate a token for multiple logins.

HTTP Method

POST

URL / Resource / JSON Structure

Resource: authorize

URL: https://api-stage.bimplus.net/v2/authorize

JSON Structure:

Name	Mandatory / Optional	Type	Description
user_id	mandatory	string	The email address of the user
password	mandatory	string	The user's password
client_id	optional	string	The identifier of the used client
application_id	optional	string	The id of the application

Description

Authenticate by requesting and receiving a token.

Request

Headers

```
Content-Type: application/json
```

JSON

```
{
 "user_id" : "test@bimplus.net",
 "password" : "test",
 "client_id" : "9fd0bb9d-570b-4719-bfae-93e2f879c19a"
}
```

Response

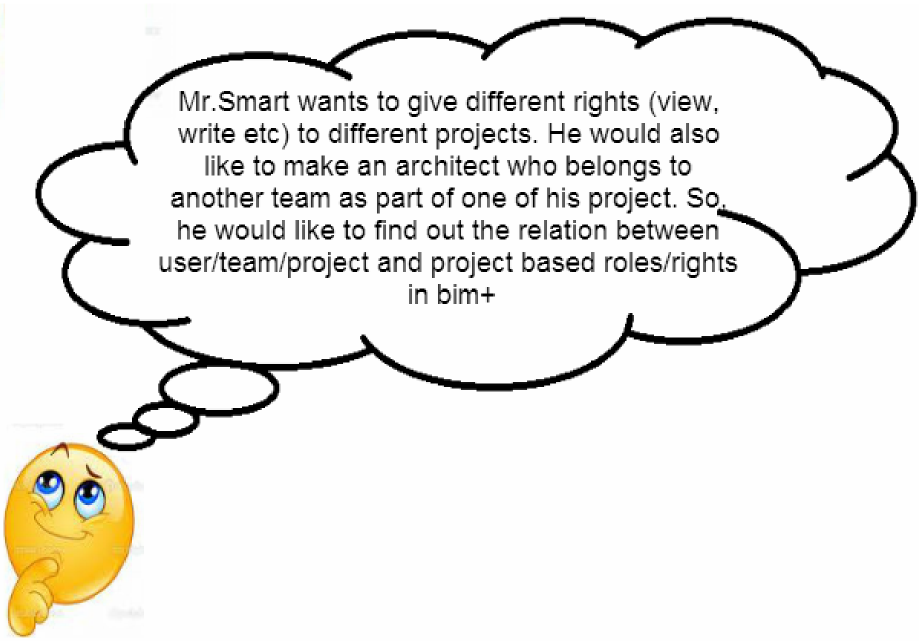
Status

```
Status: 200 OK
```

JSON

```
{
access_token: "c33fa174169e46999fe82fb07fc53e3d"
expires_in: 2591999
client_id: "9fd0bb9d-570b-4719-bfae-93e2f879c19a"
token_type: "BimPlus"
}
```

Rights & Roles. Relation between User, Team and Project

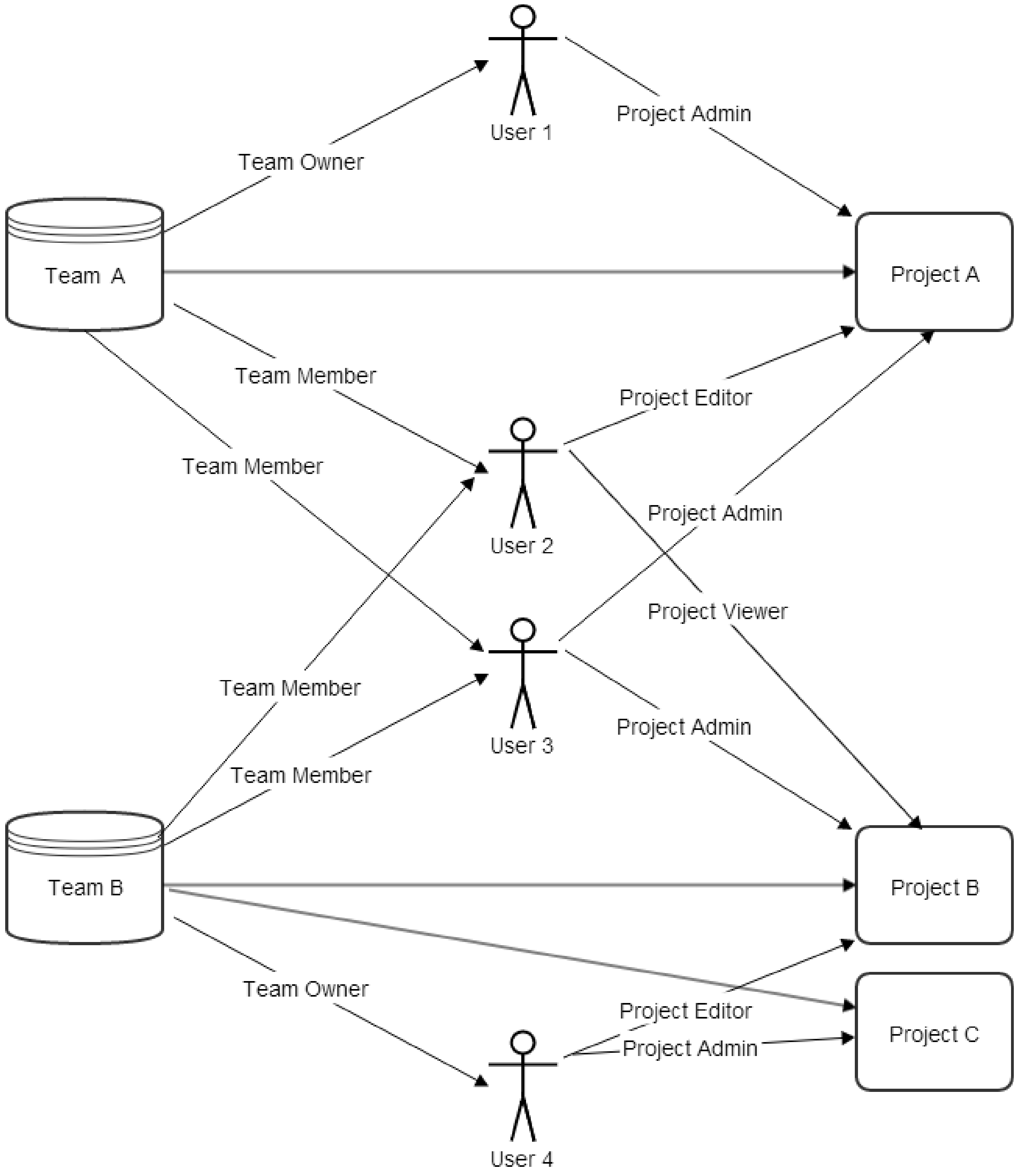

Mr.Smart wants to give different rights (view, write etc) to different projects. He would also like to make an architect who belongs to another team as part of one of his project. So, he would like to find out the relation between user/team/project and project based roles/rights in bim+

Rights vs Roles

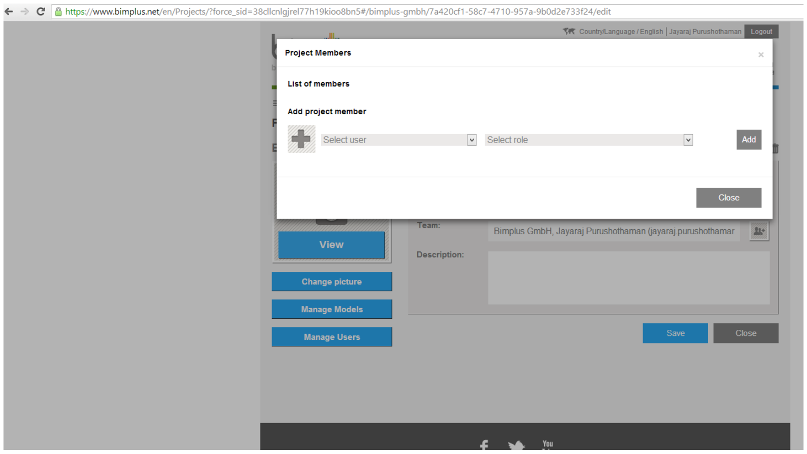
	Account / Team Owner	**Project Admin**	**Project Editor**	**Project Viewer**
Create Project	**x**			
Admin Project (eg. invite)	**x**	**x**		
Edit Project	**x**	**x**	**x**	
Delete Project	**x**	**x**		
View Project	**x**	**x**	**x**	**x**
View all Models	**x**	**x**	**x**	**x**
Create Model	**x**	**x**		
Admin Model (eg. invite)	**x**	**x**		
Assign User to Model	**x**	**x**		
Edit Model	**x**	**x**		
Import Data	**x**	**x**		
Delete Model	**x**	**x**		
View Model	**x**	**x**	**x**	**x**

- Please note that the the column names are roles and the horizontal ones are rights.
- There can be many Project Owners although there can be only one Team/Account Owner.

Relation between User, Team & Projects



So, in Bimplus the users have project based roles. After assigning the registered user to a team, the Team Owner or the Project Admin of that particular project has to send an invitation (via portal. See the picture below) to the user who belongs to one of his team.



https://www.bimplus.net/en/Projects/?force_sid=38cllcnlgjrel77h19kioo8bn5#/bimplus-gmbh/7a420cf1-58c7-4710-957a-9b0d2e733f24/edit
Country/Language / English
Jayaraj Purushothaman
Logout

Project Members

List of members

Add project member

Select user
Select role
Add
Close
View
Change picture
Manage Models
Manage Users

Team: Bimplus GmbH, Jayaraj Purushothaman (jayaraj.purushothamar

Description: