|  |  |
| --- | --- |
|  | http://1.bp.blogspot.com/-3yCmHZhDl3I/UVHm5wa90gI/AAAAAAAABPY/qtBlb_XrLaM/s1600/Screen+Shot+2013-03-26+at+12.19.27+PM.png |
|  |  |

BIM REST API 

Version 0.99 - Resources

* Information
* Project
* Revision
* User
* BCF
* Attachment

© BuildingSMART

info@buildingsmart.com

<https://github.com/BuildingSMART/BCF-API>

[1. Introduction 3](#_Toc390416152)

[1.1 BCF REST API 3](#_Toc390416153)

[1.2 Authentication 3](#_Toc390416154)

[1.3 Paging, Sorting, Filtering 4](#_Toc390416155)

[1.4 Caching 5](#_Toc390416156)

[1.5 Cross origin resource sharing (Cors) 6](#_Toc390416157)

[2. Resources 7](#_Toc390416158)

[2.1 Information Resources 8](#_Toc390416159)

[2.1.1 schemas (link.json) 8](#_Toc390416160)

[2.1.2 colours (colour.json) 8](#_Toc390416161)

[2.2 Project Resources 9](#_Toc390416162)

[2.2.1 teams (team.json) 9](#_Toc390416163)

[2.2.2 projects (project.json) 10](#_Toc390416164)

[2.2.3 domains (domain.json) 12](#_Toc390416165)

[2.3 Revision Resources 13](#_Toc390416166)

[2.3.1 revisions (revision.json) 13](#_Toc390416167)

[2.4 BCF Resources 14](#_Toc390416168)

[2.4.1 topics (topic.json) 14](#_Toc390416169)

[2.4.2 comments (comment.json) 16](#_Toc390416170)

[2.4.3 viewpoints (viewpoint.json) 18](#_Toc390416171)

[2.4.4 components (component.json) 20](#_Toc390416172)

[2.4.5 related\_topics (related\_topic.json) 21](#_Toc390416173)

[2.4.6 document\_references (document\_reference.json) 22](#_Toc390416174)

[2.5 User Resources 23](#_Toc390416175)

[2.5.1 Register 23](#_Toc390416176)

[2.5.2 Roles 23](#_Toc390416177)

[2.5.3 Rights 24](#_Toc390416178)

[2.6 Attachment Ressources 25](#_Toc390416179)

1. Introduction

BCFv2 is the most recent version of BCF.

Github:

<https://github.com/BuildingSMART/BCF>

## 1.1 BCF REST API

BCF REST API v0.99 is the most recent version of the BCF REST API.

Github:

<https://github.com/BuildingSMART/BCF-API>

Examples:

<http://docs.bcfapi.apiary.io>

* All API access is over HTTPS.
* Data is sent as query parameters and received as JSON.
* Every resource has a corresponding Json Schema.
* Json Hyper Schema for link definition

## 1.2 Authentication

OAuth2

## 1.3 Paging, Sorting, Filtering

* [Link header introduced by RFC 5988](http://tools.ietf.org/html/rfc5988#page-6).
* custom HTTP header ***X-Total-Count (HEAD)***

**page**

**page\_size**

**sort**

Only „descending“ if property exits. If no descending property is there -> ascending=true

if descending=true -> descending

if descending=false -> ascending

**"&" to combine different query parameters (operations)**

Example:

*v2/projects/{guid}/topics?page=1&page\_size=5&sort=priority&descending=true*

**Filter use escape characters**  (<http://www.december.com/html/spec/esccodes.html>)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Operation** | **Escape Characters** |
| > | Greater than | %3E |
| < | Less than | %3C |
| \* | Wild card for contains | \* |
| = | Equal to | %3D |
| >= | Greater than or equal to | %3E%3D |
| <= | Less than or equal to | %3C%3D |
| != | Is not equal to | %21%3D |
| & | And | %26 |
| | | Or | %7C |

Example:

*v2/projects/{guid}/topics?page=1&page\_size=5&sort=priority&descending=true&filter=(label%3D Architecture%7CTopicLabel%3D Structural)%26topic\_status%21%3DClosed*

## 1.4 Caching

ETags, or entity-tags, are an important part of HTTP, being a critical part of caching, and also used in "conditional" requests.

The ETag response-header field value, an entity tag, provides for an "opaque" cache validator.

The easiest way to think of an etag is as an MD5 or SHA1 hash of all the bytes in a representation. If just one byte in the representation changes, the etag will change.

ETags are returned in a response to a GET:

joe@joe-laptop:~$ **curl --include http://bitworking.org/news/**

HTTP/1.1 200 Ok

Date: Wed, 21 Mar 2007 15:06:15 GMT

Server: Apache

**etag: "078de59b16c27119c670e63fa53e5b51"**

Content-Length: 23081

…..

## 1.5 Cross origin resource sharing (Cors)

The server will put the "Access-Control-Allow-Headers" in the response header and specify who can access the server(json) resources. The client can look for this value and proceed with accessing the resources.

The server has a web config file .. "\*" means the server allow the resources for all domains.

<httpProtocol>

<customHeaders>

<add name="Access-Control-Allow-Headers" value="Content-Type, Accept, X-Requested-With, Authorization" />

<add name="Access-Control-Allow-Methods" value="GET, POST, PUT, DELETE, OPTIONS" />

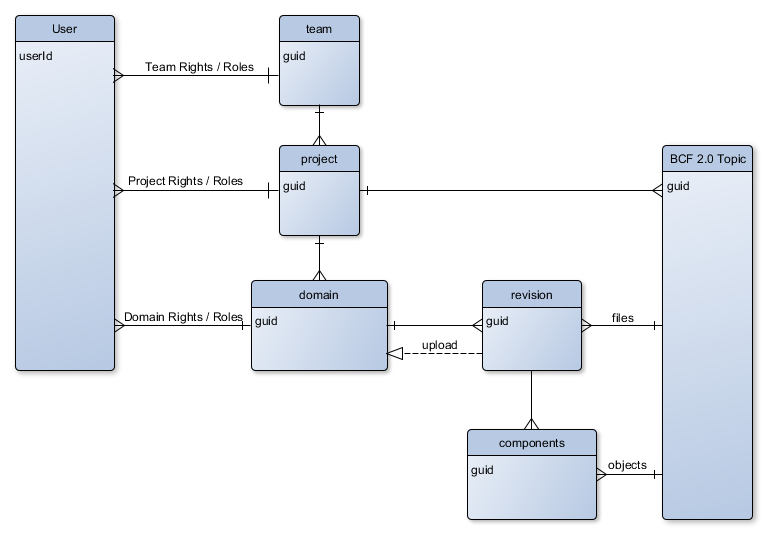
<add name="Access-Control-Allow-Origin" value="\*" />

</customHeaders>

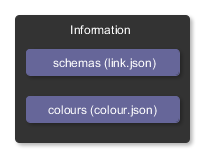
</httpProtocol>

2. Resources

* Information
* *Project*
* *BCF*
* *Revision*
* *User*
* Attachment



## 2.1 Information Resources



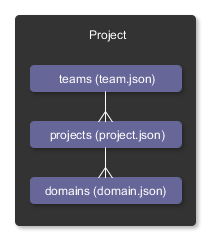
##### 2.1.1 schemas (link.json)

|  |
| --- |
| **GET** /V0.99/schemas   * GET - Retrieve link schema that contains all available schemas in BIM-API. |
| **GET** /V0.99/schemas/{rel}   * GET - Retrieve a specific schema |

##### 2.1.2 colours (colour.json)

|  |
| --- |
| **GET** /V0.99/colours   * GET - Retrieve ARGB values for colours. |

## 2.2 Project Resources



##### 2.2.1 teams (team.json)

|  |
| --- |
| **GET, POST** /V0.99/teams   * GET - Retrieve a list of teams where the currently logged on user is assigned to with his specific roles. * POST - Add a new team |
| **GET, PUT, DELETE** /V0.99/teams/{guid}   * GET - Retrieve a specific team * PUT - Modify a specific team * DELETE - Delete a specific team |

##### 2.2.2 projects (project.json)

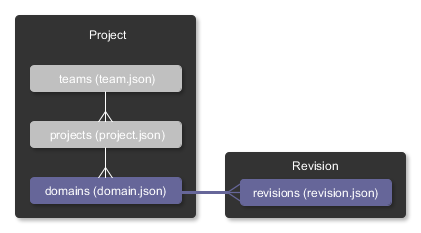
|  |
| --- |
| **GET, POST** /V0.99/ teams/{guid}/projects   * GET - Retrieve a list of projects where the currently logged on user is assigned to with his specific roles. * POST - Add a new project |
| **GET, PUT, DELETE** /V0.99/projects/{guid}   * GET - Retrieve a specific project * PUT - Modify a specific project * DELETE - Delete a specific project   Long URL: /V0.99/ teams/{guid}/projects/{guid} |
| **GET, POST, PUT,DELETE** /V0.99/projects/{guid}/thumbnail   * GET - Retrieve the project thumbnail * POST - Add the project thumbnail * PUT - Change the project thumbnail * DELETE – Delete the project thumbnail   Long URL: /V0.99/ teams/{guid}/projects/{guid}/thumbnail |
| **GET, POST, PUT, DELETE** /V0.99/projects/{guid}/extension   * GET - Retrieve the project extension schema (extensions.json) * POST - Add the project extension schema (extension.json) * PUT - Change the project extension schema (extensions.json) * DELETE – Delete the project extension schema (extensions.json)   Long URL: /V0.99/ teams/{guid}/projects/{guid}/extensions |

|  |
| --- |
| **GET, POST** /V0.99/projects/{guid}/attachments   * GET - Retrieve the list of attachments on a project * POST - Add a new attachment to a project   Long URL: /V0.99/ teams/{guid}/projects/{guid}/attachments |
| **GET, PUT, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * PUT - Change a specific attachment * DELETE - Delete a specific attachment   Long URL: /V0.99/ teams/{guid}/projects/{guid}/attachments/{guid} |

##### 2.2.3 domains (domain.json)

|  |
| --- |
| **GET, POST** /V0.99/projects/{guid}/domains   * GET - Retrieve a list of domains within a project where the currently logged on user is assigned to with his specific roles. * POST - Add a new domain to a project   Long URL: /V0.99/ teams/{ guid }/projects/{guid}/domains |
| **GET, PUT, DELETE** /V0.99/domains/{guid}   * GET - Retrieve a specific domain * PUT - Modify a specific domain * DELETE - Delete a specific domain   Long URL: /V0.99/ teams/{ guid }/projects/{guid}/domains/{guid} |

## 2.3 Revision Resources

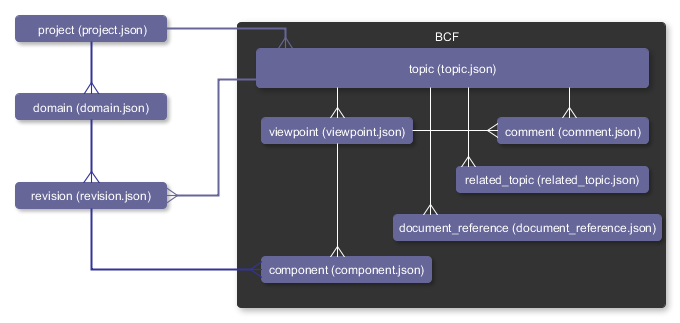


Revisions (IFC-Files) are belonging to a *Domain* where they are uploaded to.

##### 2.3.1 revisions (revision.json)

|  |
| --- |
| **GET, POST** /V0.99/domains/{guid}/revisions   * GET - Retrieve revisions of a domain. * POST - Add (Upload) a new revision to a domain   Long URL: /V0.99/ teams/{guid }/projects/{guid}/domains/{guid}/revisions |
| **GET, DELETE** /V0.99/revisions/{guid}   * GET - Retrieve a specific revision * DELETE - Delete a specific revision   Long URL: /V0.99/ teams/{guid }/projects/{guid}/domains/{guid}/revisions/{guid} |

## 2.4 BCF Resources



##### 2.4.1 topics (topic.json)

|  |
| --- |
| **GET, POST** /V0.99/projects/{guid}/topics   * GET - Retrieve topics of a project (default sort = CreationDate) * POST - Add a new topic to a project   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics |
| **GET, PUT, DELETE** /V0.99/topics/{guid}   * GET - Retrieve a specific topic * PUT - Modify a specific topic * DELETE - Delete a specific topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid} |
| **GET** /V0.99/topics/{guid}/revisions   * GET - Retrieve all revisions related to a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/revisions |

|  |
| --- |
| **GET, POST** /V0.99/topics/{guid}/attachments   * GET - Retrieve the list of attachments on a topic * POST - Add a new attachment to a topic   Long URL: /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/attachments |
| **GET, PUT, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * PUT - Change a specific attachment * DELETE - Delete a specific attachment   Long URL: /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/attachments/{guid} |

##### 2.4.2 comments (comment.json)

|  |
| --- |
| **GET, POST** /V0.99/topics/{guid}/comments   * GET - Retrieve comments of a topic * POST - Add a new comment to a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/comments |
| **GET, PUT, DELETE** /V0.99/comments/{guid}   * GET - Retrieve a specific comment * PUT - Modify a specific comment * DELETE - Delete a specific comment   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/comments/{guid} |
| **GET, POST, DELETE** /V0.99/comments/{guid}/viewpoint   * GET - Retrieve the viewpoint related to a comment * POST - Add a viewpoint relation to a comment * DELETE - Delete the viewpoint relation on a comment   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/comments/{guid}/viewpoints |
| **GET, POST, DELETE** /V0.99/comments/{guid}/reply\_to   * GET - Retrieve the replyTo comment related to a comment * POST - Add a replyTo comment relation to a comment * DELETE - Delete the replyTo comment relation on a comment   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/comments/{guid}/reply\_to |

|  |
| --- |
| **GET, POST** /V0.99/comments/{guid}/attachments   * GET - Retrieve the list of attachments on a comment * POST - Add a new attachment to a comment   Long URL: /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/attachments |
| **GET, PUT, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * PUT - Change a specific attachment * DELETE - Delete a specific attachment   Long URL:  /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/comments/{guid}/attachments/{guid} |

##### 2.4.3 viewpoints (viewpoint.json)

|  |
| --- |
| **GET, POST** /V0.99/topics/{guid}/viewpoints   * GET - Retrieve viewpoints of a topic * POST - Add a new viewpoint to a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints |
| **GET, PUT, DELETE** /V0.99/viewpoints/{guid}   * GET - Retrieve a specific viewpoint * PUT - Modify a specific viewpoint * DELETE - Delete a specific viewpoint   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid} |
| **GET, POST** /V0.99/viewpoints/{guid}/comments   * GET - Retrieve all comments related to a viewpoint   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/comments |
| **GET, POST, DELETE** /V0.99/viewpoints/{guid}/bitmap   * GET - Retrieve the bitmap related to a viewpoint * POST - Add a bitmap to the viewpoint * DELETE - Delete the bitmap of the viewpoint   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/bitmap |

|  |
| --- |
| **GET, POST** /V0.99/viewpoints/{guid}/attachments   * GET - Retrieve the list of attachments on a viewpoint * POST - Add a new attachment to a viewpoint   Long URL:  /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/attachments |
| **GET, PUT, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * PUT - Change a specific attachment * DELETE - Delete a specific attachment   Long URL:  /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/attachments/{guid} |

##### 2.4.4 components (component.json)

|  |
| --- |
| **GET, POST** /V0.99/viewpoints/{guid}/components   * GET - Retrieve components of a viewpoint * POST - Add a new component to a viewpoint   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/components |
| **GET, PUT, DELETE** /V0.99/components/{guid}   * GET - Retrieve a specific component * PUT - Modify a specific component * DELETE - Delete a specific component   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/components/{guid} |
| **GET, POST** /V0.99/components/{guid}/attachments   * GET - Retrieve the list of attachments on a component * POST - Add a new attachment to a component   Long URL:  /V0.99/ teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/attachments |
| **GET, PUT, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * PUT - Change a specific attachment * DELETE - Delete a specific attachment   Long URL:  /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/viewpoints/{guid}/components/{guid}/attachments/{guid} |

##### 2.4.5 related\_topics (related\_topic.json)

|  |
| --- |
| **GET, POST** /V0.99/topics/{guid}/related\_topics   * GET - Retrieve related topics of a topic * POST - Add a new related\_topic to a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/related\_topics |
| **DELETE** /V0.99/topics/{guid}/related\_topics/{guid}   * DELETE - Delete a related topic from a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/related\_topics/{guid} |

##### 2.4.6 document\_references (document\_reference.json)

|  |
| --- |
| **GET, POST** /V0.99/topics/{guid}/document\_references   * GET - Retrieve documents referenced of a topic * POST - Add a new document reference to a topic   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/document\_references |
| **GET, DELETE** /V0.99/topics/{guid}/document\_references/{guid}   * GET - Retrieve a document referenced to a topic * DELETE - Delete a document reference   Long URL: /V0.99/teams/{guid}/projects/{guid}/topics/{guid}/document\_references/{guid} |

## 2.5 User Resources

There are some user rights and roles predefined in BIM – API.

##### 2.5.1 Register

To register a user, use the webplatform of the BIM – API server.

##### 2.5.2 Roles

BIM-Manager:

Has all rights within a “Project”

Domain-Manager:

Has all rights within a “Domain”

Domain-User:

Has specific rights within a “Domain”

##### 2.5.3 Rights

|  |  |  |  |
| --- | --- | --- | --- |
| *Rights* | **BIM - Manager** | **Domain - Manager** | **Domain - User** |
| Create, Edit, Delete Project | x |  |  |
| Invite User to Project | x |  |  |
| Create, Edit, Delete Domain | x |  |  |
| View all Domains | x |  |  |
| Assign User to Domain | x |  |  |
| View Domain | x | x | x |
| Upload Revision to Domain | x | x |  |
| Add, Edit, Delete BCF-Topics and Comments | x | x | x |

Open questions to discuss:

What happens if a BCF-Topic is related to more Domains / Revisions and the user has no Rights on one of them ?

How is the handling of 2 or more revisions from different Domains (p.e. Architecture and Heating) in generell related to the viewer. Do bim+ / Catenda / Graphisoft / Solibri allow to open several Revisions in parallel based on the Right “View Domain” ?

## 2.6 Attachment Ressources

An attachment can be added to the following objects:

* Project
* Topic
* Viewpoint
* Comment
* Component

|  |
| --- |
| **GET, POST** /V0.99/objects/{guid}/attachments   * GET - Retrieve the list of attachments of an object * POST - Add a new attachment to a project |
| **GET, DELETE** /V0.99/attachments/{guid}   * GET - Retrieve a specific attachment * DELETE - Delete a specific attachment |