# Autodesk 360 BIM Viewer in Web Central

This example embeds the Autodesk 360 cloud-based viewer into Web Central.

The goal is to provide a solution template that illustrates the technical features we need to integrate the cloud-based viewer into building lifecycle workflows for commissioning, maintenance, preventive maintenance, space planning, capital planning and lifecycle costing.

Specifically, the technical requirements are to:

* Show a model based on a URN.
* Thematically highlight rooms in the model.
* Find and highlight an item such as a room in the model.
* React to a click event on an item such as equipment in the model.

## Autodesk 360 API and Viewer

360 API Documentation - <http://developer.api.autodesk.com/documentation/v1/index.html>

Developer Portal Home - <https://developer.autodesk.com/>

360 Viewer and API – Forums - <http://forums.autodesk.com/t5/view-and-data-api/bd-p/95>.

Viewer API - <http://developer.api.autodesk.com/documentation/v1/vs/viewers/index.html>

## Steps to upload a model with Autodesk BIM 360 Cloud Server

1. Go to <https://developer.autodesk.com/> to register and get client\_id and client\_secret, which will be used to get the access key for all activities with the cloud;
2. Use the client\_id and client\_secret to get access key by curl command (see curl commands for detail), the access key will be expired in minutes;
3. Create a bucket to hold uploaded models in the cloud by curl command (see curl commands for detail);
4. Upload a model by curl command (see curl commands for detail);
5. Get the model’s URN based on bucket and model file by <https://www.base64encode.org/>. Un-encoded value looks like urn:adsk.objects:os.object:**mybucket**/**mymodelfile.dwfx**, and encoded URN looks like **dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6bXlidWNrZXQvbXltb2RlbGZpbGUuZHdmcw==**;
6. Register the model by curl command (see curl commands for detail);
7. Check the model status by curl command (see curl commands for detail), when its status is 100% complete, the model is ready to use.

## Curl commands

1. Create a bucket

curl -k --header "Content-Type: application/json" --header "Authorization: Bearer **LU9krbPH2LKhKTYOcZ8RYOVj7Nmd**" --data "{\"bucketKey\":\"**archibus\_test**\",\"policy\":\"transient\"}" <https://developer.api.autodesk.com/oss/v1/buckets>

1. Upload a model

curl -k -i -X PUT "https://developer.api.autodesk.com/oss/v1/buckets/**archibus\_test**/objects/**bosoff.rvt**" -H "Content-Type: application/octet-stream" -H "Authorization: Bearer **1YRskNunxALwtWI79gmI9kOH1IRu**" --data-binary @**bosoff.rvt**

1. Register a uploaded model

curl -k -H "Content-Type: application/json" -H "Authorization:Bearer **pFqF59FQPzbP0m5JUY6Rv5SR4snn**" -i -d "{\"urn\":\"**dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6YXJjaGlidXNfdGVzdC9ib3NvZmYyLnJ2dA==\**"}" <https://developer.api.autodesk.com/viewingservice/v1/register>

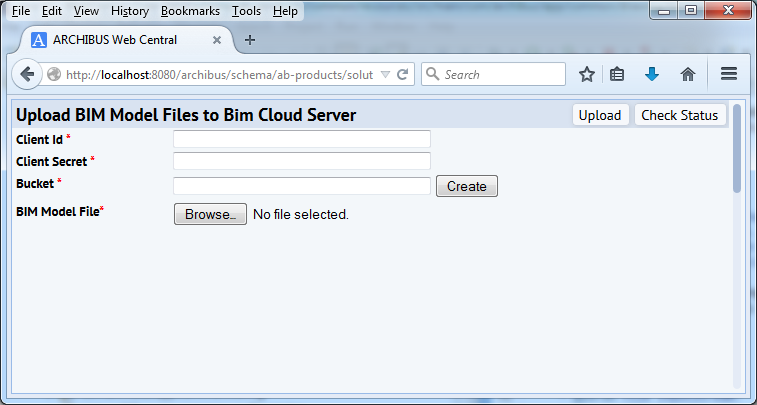
1. Check the registered model’s status

curl -k -i -H "Authorization: Bearer **dyWP1KeXyV5PB78tOzouXziOWjl7**" -X GET [https://developer.api.autodesk.com/viewingservice/v1/**dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6YXJjaGlidXNfdGVzdC9ib3NvZmYyLnJ2dA==**/status](https://developer.api.autodesk.com/viewingservice/v1/dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6YXJjaGlidXNfdGVzdC9ib3NvZmYyLnJ2dA==/status)

1. Retreat access key

curl --data "client\_id=**56vRODyX63SwkFRCbuQWfIi5QULGlVg5**&client\_secret=**9HUtjlMAyL7IeqR4&grant\_type=client\_credentials**" https://developer.api.autodesk.com/authentication/v1/authenticate --header "Content-Type: application/x-www-form-urlencoded" -k

## The view to facilitate uploading and registering with BIM 360 Cloud Server



You need to provide all required info.

If having not created a bucket, please provide the bucket and click its “Create” button.

If there is no error message after clicking “Upload” button, you could click “Check Status” to check uploaded file’s status after uploading progress bar is disappeared.

If the status is “success”, it’s ready to load the file from cloud server by its urn value and review 3D model.

## Core BIM service DrawingBimService

/\*\*

\* Gets Autodesk View 360 Data Access key.

\*

\* @param clientId like "56vRODyX63SwkFRCbuQWfIi5QULGlVg5".

\* @param clientSecret like "9HUtjlMAyL7IeqR4".

\* @return String access token.

\* @throws ExceptionBase if Could throws an exception.

\*/

String getAccessToken(final String clientId, final String clientSecret)

/\*\*

\*

\* Converts revit guid to model valid guid.

\*

\* @param revitGuid revit guid.

\* @return model valid guid.

\*/

String convertRvitGuid(final String revitGuid)

/\*\*

\*

\* Gets the map of BIM Guid and corresponding Primary Key value.

\*

\* @param tableName String.

\* @param restriction String.

\* @param isRevit boolean.

\* @return Map<String, String>.

\*/

Map<String, String> getGuidPKValueMap(final String tableName, final String restriction,

final boolean isRevit)

/\*\*

\*

\* Gets the map of BIM Guid and corresponding Primary Key value.

\*

\* @param objectIdDbFullPath String.

\* @param accessToken String.

\* @return Map<String, Integer>.

\*/

Map<String, Integer> getRevitGuid2NodeMap(final String objectIdDbFullPath,

final String accessToken)

/\*\*

\*

\* Gets paired guid and database defined color map .

\*

\* @param asset String.

\* @param viewName String.

\* @param dataSourceId String.

\* @param restriction String.

\* @param isRevit boolean.

\* @return Map<String, String>.

\*/

Map<String, String> getGuidRGBColorMap(final String asset, final String viewName,

final String dataSourceId, final String restriction, final boolean isRevit)

/\*\*

\*

\* Creates Bucket.

\*

\* @param bucketName bucket Name.

\* @param clientId client Id.

\* @param clientSecret client Secret key.

\* @throws ExceptionBase if Could throws an exception.

\*/

void createBucket(final String bucketName, final String clientId, final String clientSecret)

/\*\*

\*

\* Uploads a file.

\*

\* @param file InputStream.

\* @param fileName file name.

\* @param bucketName bucket Name.

\* @param clientId client Id.

\* @param clientSecret client Secret key.

\* @param registering boolean to register it.

\* @throws ExceptionBase if Could throws an exception.

\*/

void uploadFile(final InputStream file, final String fileName, final String bucketName,

final String clientId, final String clientSecret, final boolean registering)

/\*\*

\*

\* Registers uploaded File.

\*

\* @param fileName file name.

\* @param bucketName bucket Name.

\* @param clientId client Id.

\* @param clientSecret client Secret key.

\* @throws ExceptionBase if Could throws an exception.

\*/

void registerFile(final String fileName, final String bucketName, final String clientId,

final String clientSecret)

/\*\*

\*

\* Check uploaded file's processing Status.

\*

\* @param fileName file name.

\* @param bucketName bucket Name.

\* @param clientId client Id.

\* @param clientSecret client Secret key.

\* @throws ExceptionBase if Could throws an exception.

\* @return Status.

\*/

Status checKStatus(final String fileName, final String bucketName, final String clientId,

final String clientSecret)

## JavaScript BIM Control Class

A new JS class Ab.bim.Autodesk is designed to wrap up APIs to interact with models.

Followings are the class contructor and public APIs:

/\*\*

\* @param divId: the id of div element to hold BIM 3D.

\* @param client\_id: get from https://developer.autodesk.com/

\* @param client\_secret: get from https://developer.autodesk.com/

\*/

constructor: function(divId, client\_id, client\_secret)

/\*\*

\* Loads model and displays it.

\* @param urn: Autodesk uploaded model URN.

\* @param isRevit: boolean if model is revit file.

\*/

load: function(urn, isRevit)

/\*\*

\* Selects the item of the mdoel.

\* @param revitGuid: revit guid.

\* @param selectedColor: color code like 0xFFFFFF or new THREE.Color("rgb(138,43,226)").

\* @param callBack: call back function with selected item's Guid as parameter.

\*/

select: function(revitGuid, selectedColor, callBack)

/\*\*

\* Selects and Zooms into the item of the model.

\* @param revitGuid: revit guid.

\* @param selectedColor: color code like 0xFFFFFF or new THREE.Color("rgb(138,43,226)").

\* @param callBack: call back function with selected item's Guid as parameter.

\*/

selectZoomIn: function(revitGuid, selectedColor, callBack)

/\*\*

\* Sets model item's click event.

\* @param asset: such as rm or eq.

\* @param callBack: callBack call back function with primary keys value as parameter like 'BOSOFF;01;105' for rm asset.

\* @param restriction: restriction like "rm.bl\_id='BOSOFF'".

\*/

onClick: function(asset, callBack, restriction)

/\*\*

\* Sets the thematically highlighting of the model items like rooms.

\* @param asset: asset such as rm or eq.

\* @param viewName: view name.

\* @param dataSourceId: datasource id.

\* @param restriction: restriction like "rm.bl\_id='BOSOFF'".

\* @param option: {isolation: false} if isolating the highlighted items in the model.

\*/

setAssetHighlighting: function(asset, viewName, dataSourceId, restriction, option)

/\*\*

\* Gets Autodesk cloud server access key.

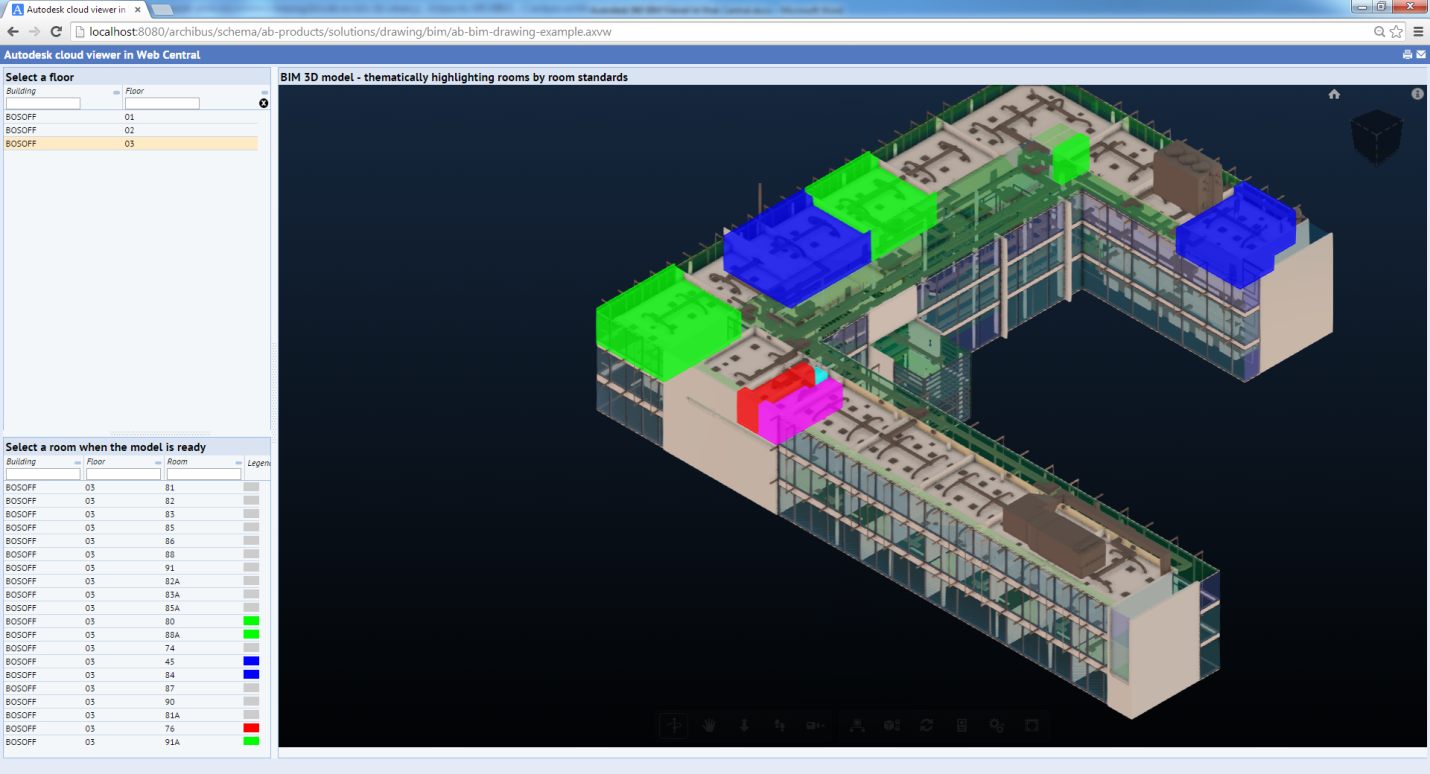
\*/

getAccessToken: function()

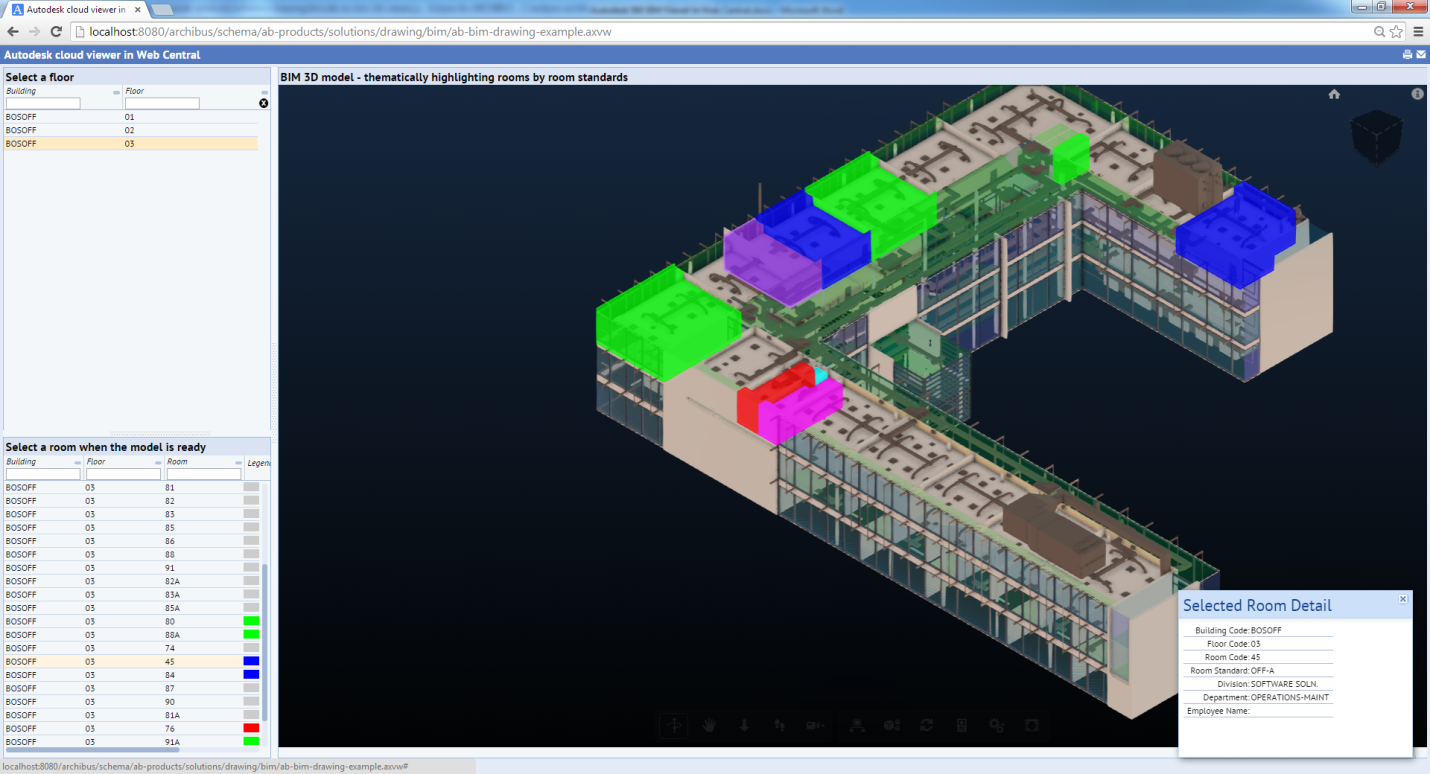
## To view following examples, please import sample data bosoff\_fl.xlsx , bosoff\_rm.xlsx, bosoff\_eq.xlsx and bosmed\_ rm(room standards).xlsx into database

## Example 1: <http://localhost:8080/archibus/schema/ab-products/solutions/drawing/bim/ab-bim-drawing-rooms-example.axvw>

### Select a floor to open its corresponding 3D model, and selected floor’s rooms are highlighted by their standards

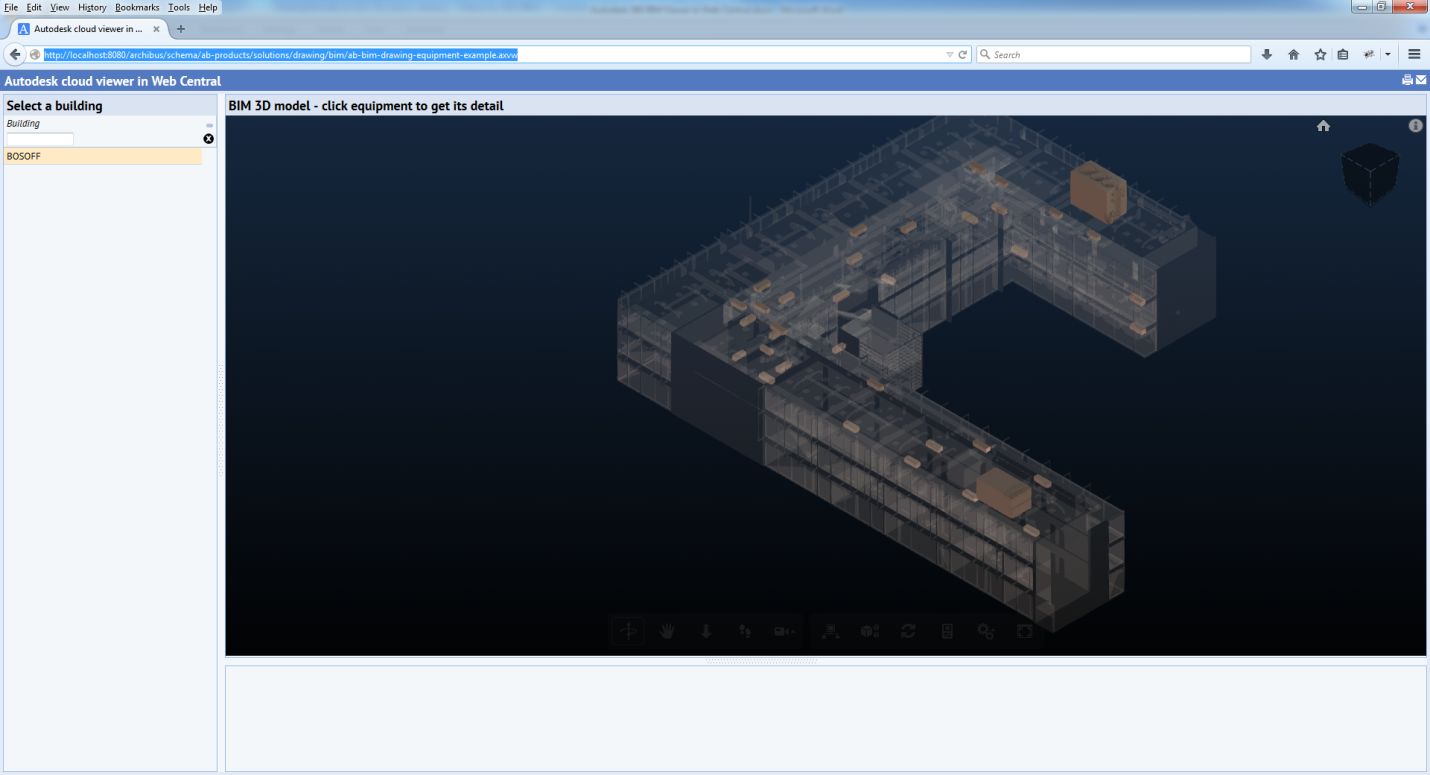


### Select a room to find and highlight it in the model, and show its detail report

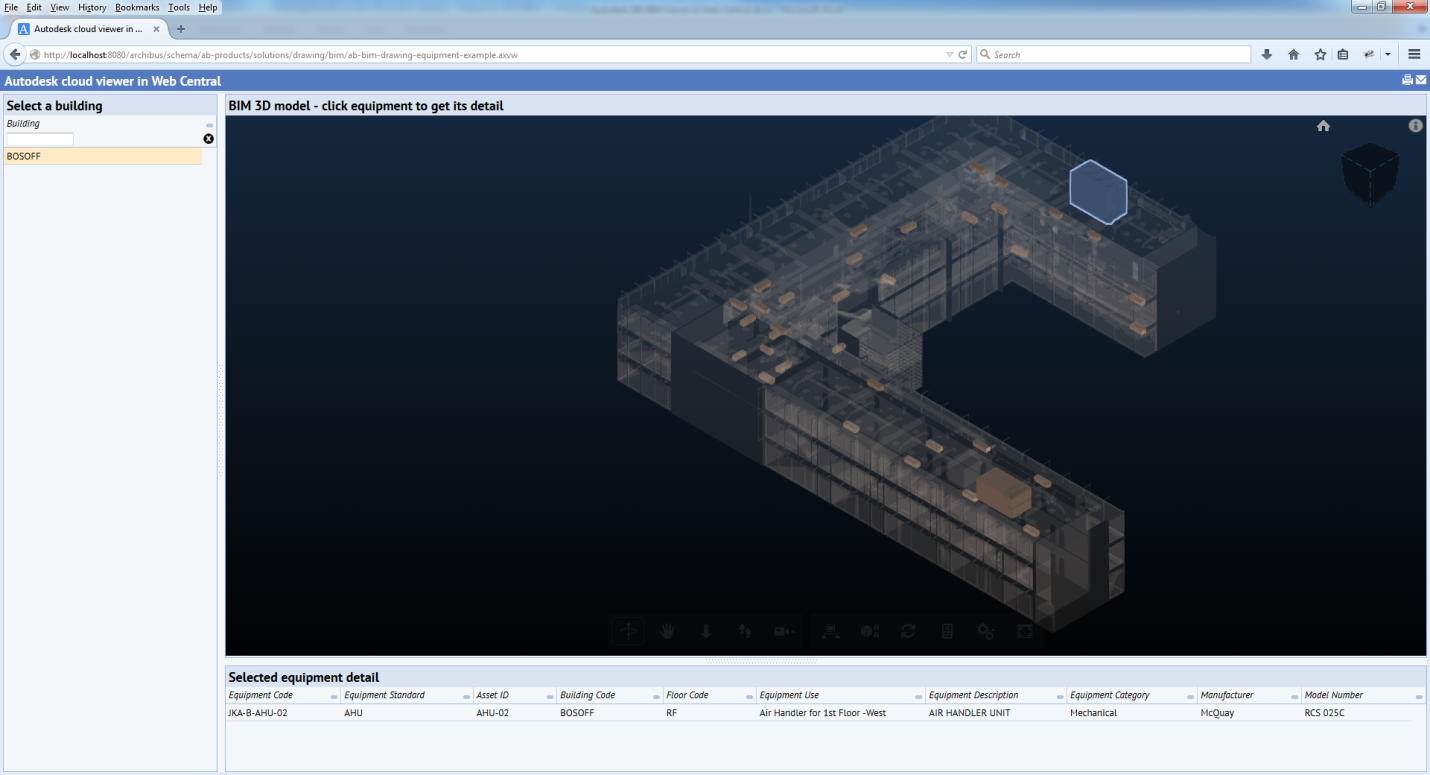


## Example 2: <http://localhost:8080/archibus/schema/ab-products/solutions/drawing/bim/ab-bim-drawing-equipment-example.axvw>

### Select a building to open its 3D model, and its equipment items are clickable



### Click equipment item to display its detail report



### Notice: Autodesk 360 Viewer requires browsers to have a good support of WebGL. Based on testing, the example view runs fine on Chrome (V39), Firefox (V34) and Opera (V26), but it’s not working on IE(V11). WebGL-supported browsers - <http://en.wikipedia.org/wiki/WebGL>