

[CS468447]

[Efficient Ways to Visualize and Analyze Data in a BIM 360 Project with Power BI]

[Xiaodong Liang]
[Autodesk]

Learning Objectives

- Explore the BIM 360 data available for inspecting a project.
- Discover the different ways to utilize and integrate BIM 360 project data with Power BI.
- Learn how to use the Forge API to extract data automatically.
Learn how to build an interactive visualization tool using Forge and Power BI.

Description

BIM (Building Information Modeling) practitioners are seeking more-efficient ways to analyze the accumulated data in BIM 360 projects on a daily basis. Microsoft Power BI is a popular, powerful tool that provides intelligent dashboards to inspect the data. In this class, we will show you three different ways to analyze BIM 360 projects data with Power BI for BIM coordination and communication purposes. The first way will be with no coding; the second will be with a little bit of coding; and the third way will be with real programming (and handy templates). You will also learn how to build a simple app that integrates Power BI and Forge. Join us and discover which method is suited for your skill set and your workflow, or combine them in different scenarios.

Speaker(s)

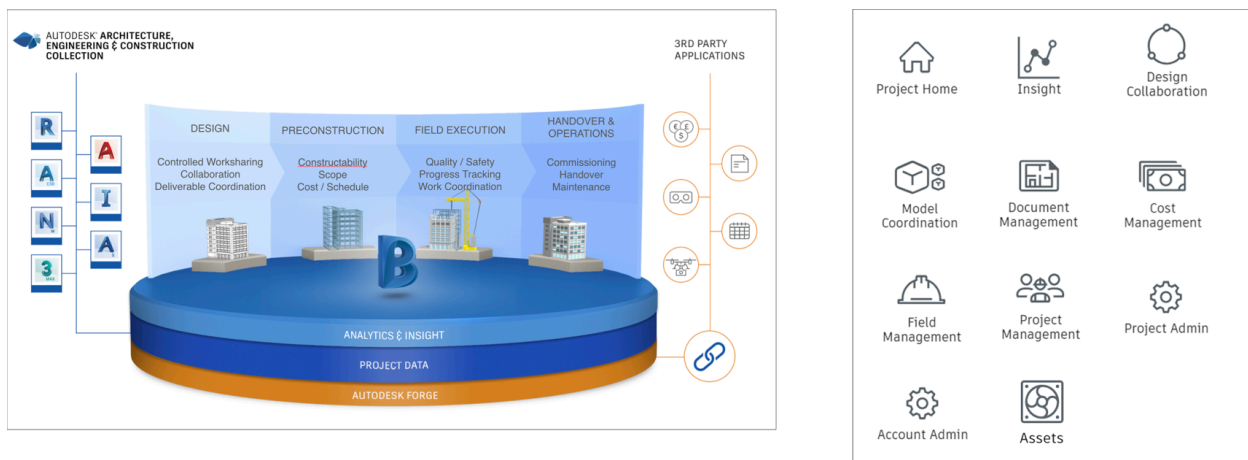
Xiaodong Liang joined Autodesk in 2007 as a developer advocate. Starting with desktop products' APIs, Xiaodong is now engaging in Autodesk Forge and AEC solutions, including BIM360 and Revit etc. He is one of the contributors of Forge blog and ADN DevBlog, researching integrations by Forge and web & cloud technology . Xiaodong is based in China, yet mingling with the global community of Autodesk and programming. He is also the ambassador of Intel IoT technology

Analyze BIM project by BIM 360 data and Power BI

About BIM 360 Data

BIM360 is a modules collection across whole lifecycle of Building Information Modeling. It is a cloud-based platform from design, plan, pre-construction, on-site management, finally to commissions, maintenances and operations. It is collaboration solution that connects the entire project team and helps streamline BIM project review and coordination workflows. No matter owners, general contractors, subcontractors, suppliers can work on with the same platform and same data, following with the up-to-date information.

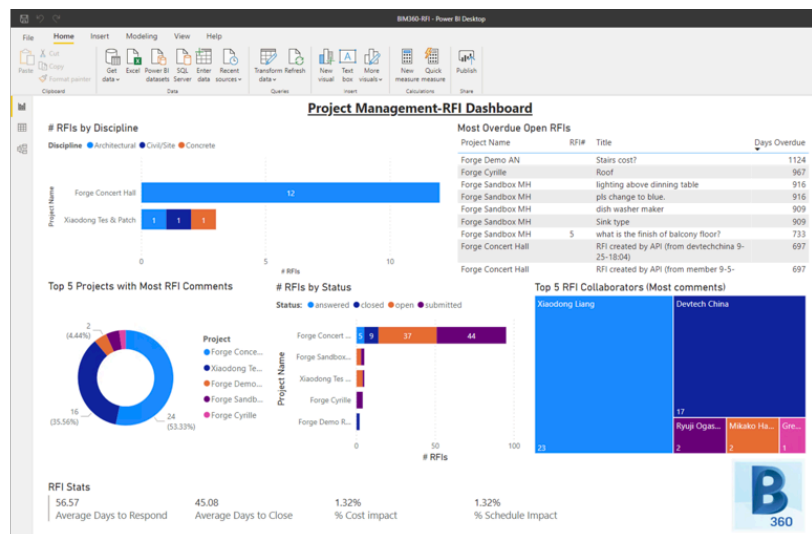
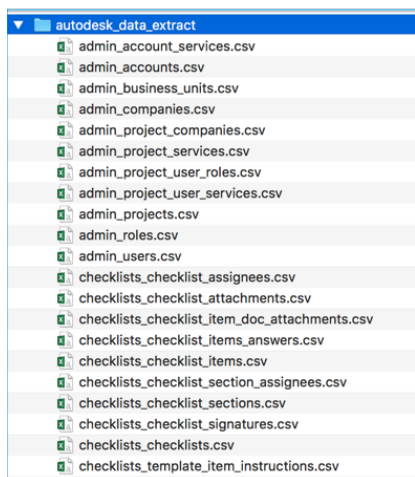
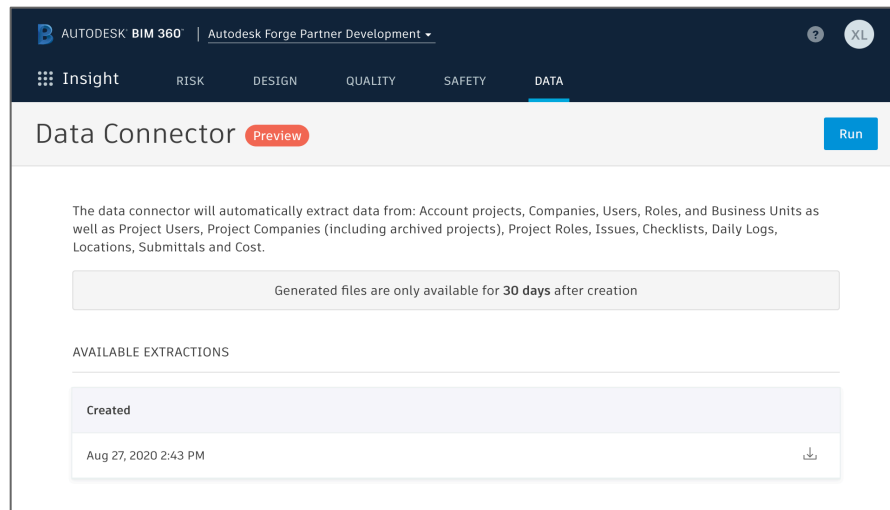
In the process of project, a huge number of data will be generated, not only models, documents, users, but also issues, RFIs, reports, activities etc. With these data, the administrator and each part can see the dynamic view of the history of the project, progress of a project, the health of the project and the risks and actions that can make the project with better quality and manage the project cost to the most extent..



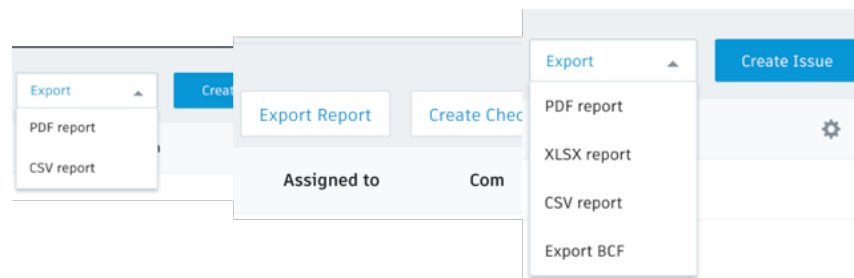
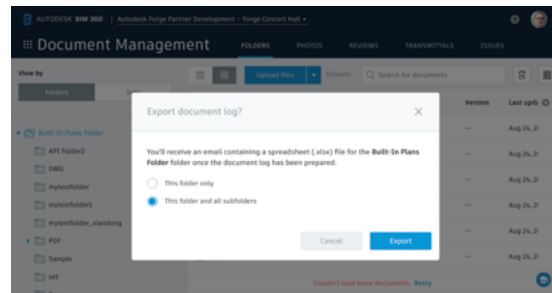
About Power BI

BIM360 allows you to centralize your project's information. Insights module can analyze project data and generate some views about how a project is going on, while in many cases, it requires some customizable visualization and analysis of the data. It will be more widely used by sending data to special data analysis app. PowerBI is one of typical tools to inspect data. In addition, with the aggressive evolving of Artificial Intelligence, more and more projects are connecting web services of AI.

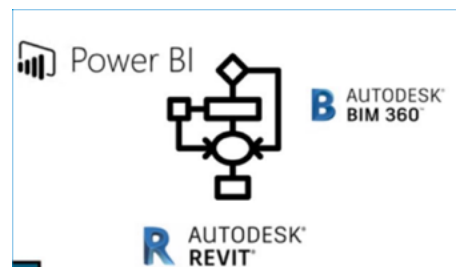
PowerBI provides rich types of visuals: general dashboards types, and huge number of custom visuals. The visuals are built from same data source. Based on that, Power BI core engine plays a role to bridge the visuals interactively and intelligently. e.g. when a cluster of types in Pie chart is selected, the corresponding rows of table will be also summarized. It can support various data source such as file, database, streaming data, on-demand custom data etc. It also allows extending the basic visuals abilities. The 3rd party can produce their own visuals to present data



If we do not want to do coding, the possible ways to build a Power BI report with BIM360 data would be: first: Data Connector, it exports account-level data. either import the excel files, or re-use the templates Data Connector exported. Secondly we can take advantage of the export functionalities of specific modules. Almost in each modules of BIM360, export function is available. e.g. document logs tells document lists, basic information, issues/rfi/checklist report can export the corresponding information. These exported files can be excel or csv, or pdf format. the excel or csv data can be imported to PowerBI report.



One more data source is [Revit Database Link](#). It is a feature of Revit. It can dump the model meta data in the format of mdb. This is a database which can be imported to Power BI data source.



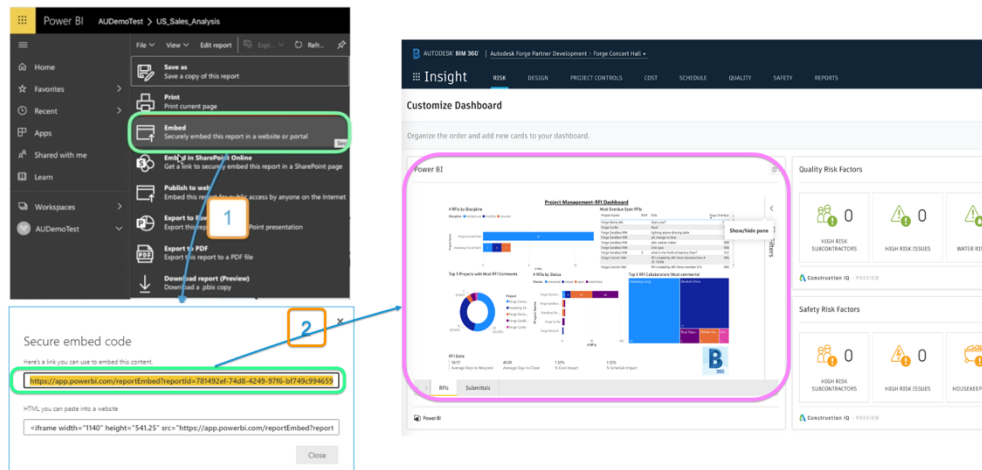
Power BI Partner Card

Insights module provides some dashboards to present the project information. [Partner card](#) allows you to embed 3rd party apps inside the layout of Insights. The project admin or users can check the data within one environment.

The card is an iFrame. iFrame is a terminology in HTML programming. By specifying the url of 3rd party solution, the container of iFrame will display the contents of other web sources.

You have probably seen countless YouTube videos embedded on sites. That is also an example of iFrame.

To place a PowerBI report, it is very straightforward, you will firstly have produced a report/dashboard of PowerBI, get embed link of the report, and input to the link to the card of Insights. It will ask for logging with Power BI account. Once it is validated, the corresponding PowerBI report will be presented in BIM360.



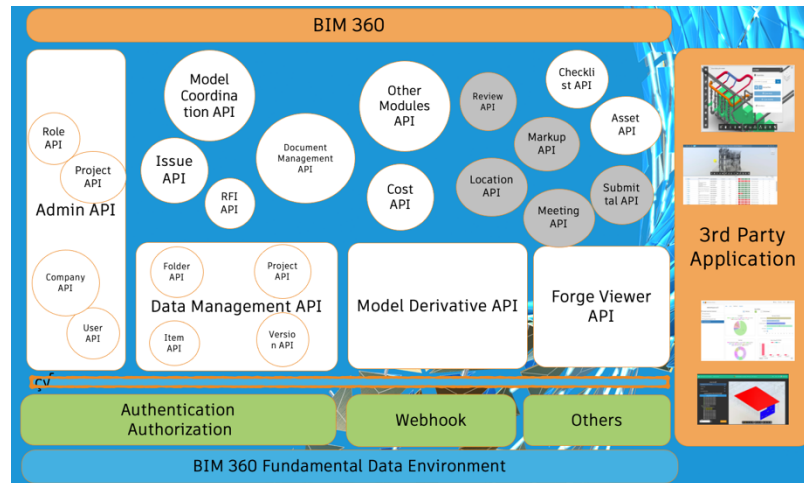
This assumes the user has been granted with the access to this report.

BIM 360 Forge API & Data Extraction

It is convenient to get data or template without coding, while the downside is it is hard to extract and build customized data, and the process is manual, exporting manually, importing manually. not efficient in some ways. In the next section, we will take a look at how to extract data from BIM360 by code.

BIM 360 is built on Forge. Forge is a web services and API which provides micro-services. API means application programming interface. e.g. the document management folder/models/documents/versions are accessible by Data Management API. We can get folder/models/documents/versions list and their data by the API. BIM 360 makes a user friendly interface to present the data and builds the workflow for BIM requirement. Another example is: we can see metadata and properties of BIM model in BIM360. We can view a model in the browser. The metadata and properties are actually extracted by Forge Model Derivative service, and viewing is adopting a Forge Viewer. Forge Viewer is a browser-based components that can be imbedded in the web page. Again BIM 360 makes a user friendly interface to present the data and builds the workflow for BIM requirement.

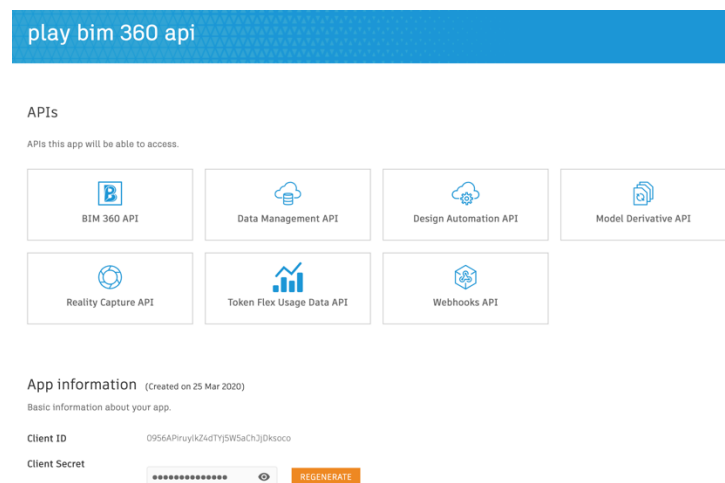
In a word, BIM 360 modules are based on the pieces of Forge services and complies the specific functionalities of BIM. In the picture, BIM360 shares the general services such as Data Management, Model Derivative API, while it has also the specific APIs of its modules features, e.g. Admin API for managing role, user, Document Management API is related with Issue, RFI, Checklists. And on top of that, more advanced APIs are available, such as Model Coordination API, Cost API. The 3rd party application can integrate with any level of these APIs, by these APIs, connect the fundamental data of BIM 360. When we say 3rd party application, one of scenarios is about using Forge service to access BIM 360 data.



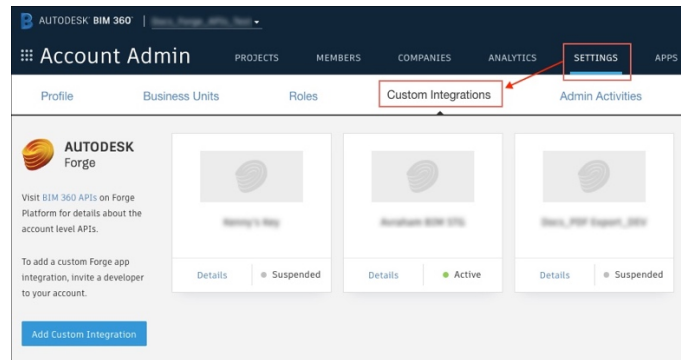
P.s. those grey out bubbles refer to the APIs that have not been exposed.

Create Forge App

To work with BIM 360 API, you will firstly create a Forge account, and create the Forge app. In this app configuration, Forge generates a unique client id and secret for you. Follow [this tutorial to create app](#).

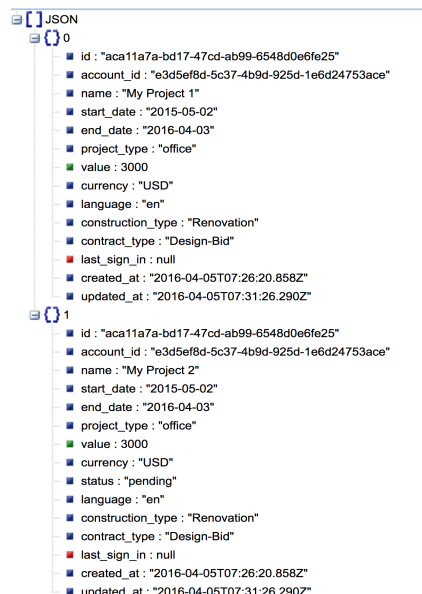


Step 2, the BIM account admin will need to permit your app to access data of the BIM account. Follow [this tutorial](#) to enable the access:



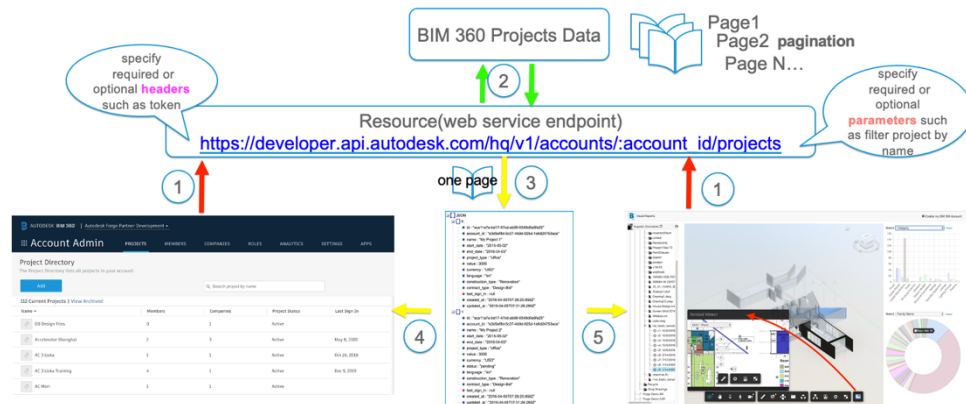
Step 3, the Forge app or scripts will call Forge services to access data. In this step, the app sets. For some data of BIM 360, we can use Forge id and secret to get access token, access token will be used in each call of web service. Forge service will verify if it is from a valid app by the token, and finally send BIM data accordingly. While for some data, the app or scripts will have to ask the BIM 360 to user to [get the authorization](#), and get access token. To be short, token is to ensure data security.

Step4, the data format Forge is json. Json is Key-value pair or array. E.g. the picture shows two projects data. id, name, start date, end date, value, address. Then we can import them data to PowerBI report.



Forge service follows the schema of restful. Restful is abbreviated from Representational state transfer (REST). it is in the format of endpoint. Each endpoint will tell: Resource URL, access

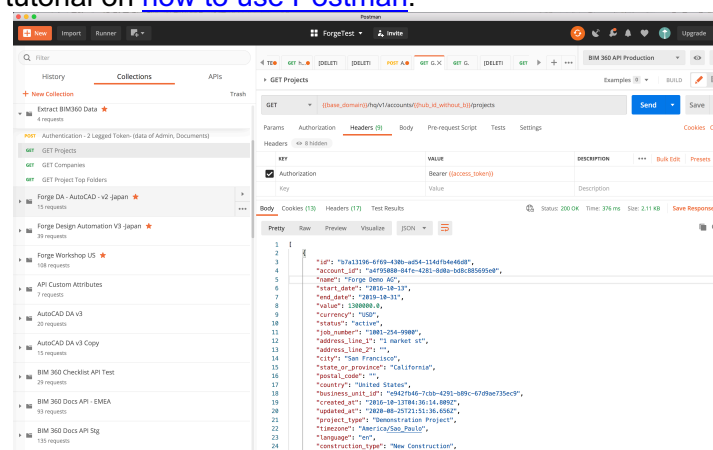
data or modify data, and the required header and parameters. This is a [recommended tutorial on what is Restful](#).



One resource will include a huge number of items. e.g. projects list, issue list, rfi lists. By default, any web service does not return all items in one call, instead, we need to specify how many items that can be exported in one page. in the specific example, the maximum number of one page for projects list is 200. We will also tell Forge which page to call. A successful call of Forge will respond with the projects data of one page. By the data, either BIM 360 or 3rd party app can integrate with the workflow.

Postman Scripts

Before working with app or scripts, we can use some tools of Restful to test the endpoints. One of my favorites is Postman. It is an app to test endpoint of web service. The UI is very friendly. Easy use, easy troubleshooting. It also supports batch running which means you can run a few endpoints in sequence. The output of the call can also be a way to get bim 360 data. One recommended tutorial on [how to use Postman](#).



On Forge documents (<https://forge.autodesk.com/> or <https://github.com/autodesk-forge>) , there are a lot of Postman demo scripts. you may google with postman + forge. Some demo scripts are available at Forge team individual Github repositories.

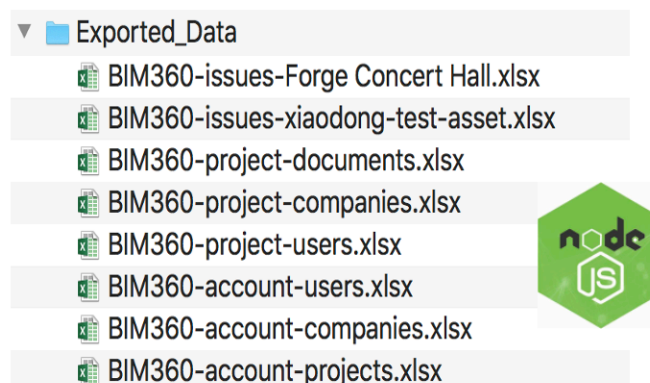
As mentioned above, some data of BIM360 requires the user authorizes the web app to access the data. It is normally by [get the authorization](#). One blog is about [Using 3 legged OAuth for Forge API's with Postman](#).

Node.js Tool to Extract BIM360 Data

With a little bit more programming skills, you can build a script to export data automatically with more flexibility and customizability. One famous script demo is written by my colleague [Sam Nseir](#). This code is importing by R script. R programming is an incredibly easy to learn programming language. It is designed for statisticians and data scientists and can be executed in Power BI. This is the [repository](#), in which it describes this scripts in detail. The only I would remind is when working with R programming, try to debug the scripts by other tool firstly, it can help you check if there is problem. Because if executing scripts directly, PowerBI cannot tell the exact reasons for some failures/problems.

The other sample is written by [Xiaodong](#) recently. It is a program of Node.js, also a hot programming and powerful. Especially, it is script, but a lot of tool can debug it. easy troubleshooting. And it is easier to implement authorization workflow, and pagination.

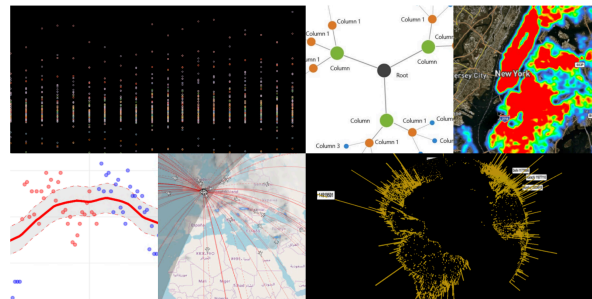
The [readme of this tool](#) tells the steps how to use. It currently supports exporting administrator data (projects, companies, users, roles) to excel files. It also supports exporting documents data of one project. And issues list of one project. You could extend the tool with more exporting. e.g. if you want to add exporting rfis list, check [rfi API reference](#), follow the similar code of [exporting issue](#) to implement the the workflow to get rfis. Define [columns of excel](#) with which data you want to export.



For some nested data, you would call more endpoints to get them. e.g. comments list of rfi, we will call [GET:Comments](#) once the rfi id is known.

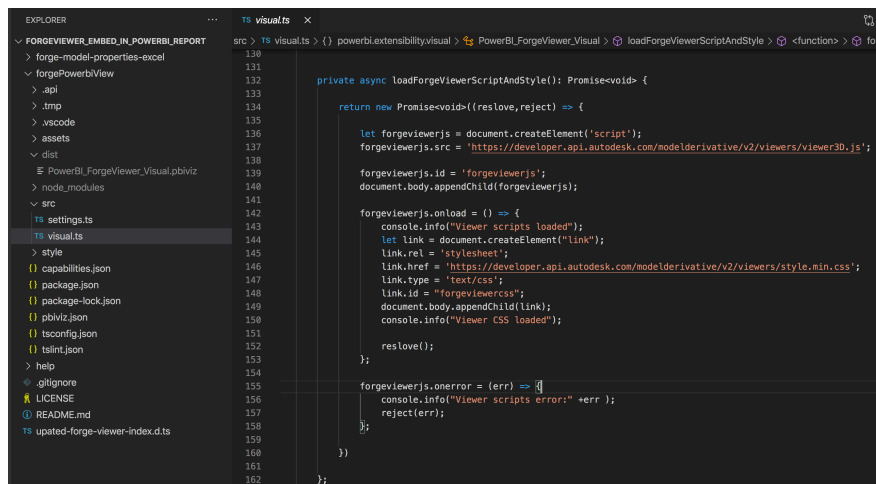
Power BI Visual of Forge Viewer

We have heard a lot of customers asking the abilities. This scenario comes from the requirement that the customers will need to work within Power BI only and have not chance to program. But they want to connect the visual model of BIM with Power BI dashboards. One 3rd party partner of Forge (VCad) has provided such solution. If you would like to purchase their existing visual, this could be one choice.



As mentioned above, Forge Viewer is a browser-based component. Technically speaking, it is also a HTML element. It can render the 3d model and 2d drawings without asking client to install any software or plugin. In the environment of PowerBI web version, it supports to imbed such HTML element natively. It is called [custom Visual](#). In custom visual, you can design your own dashboards or the custom visualization of data. heatmap, GIS map, and any fancy data simulations. The skeleton of Custom Visual is [TypeScripted](#).

By the PowerBI tutorial, we can setup the code of custom visual easily. Most external libraries will be imported locally. While due to the usage of Forge Viewer Javascript libraries, we cannot import the libraries locally, yet we need to define the logic to import Forge Viewer3D.min.js and Style.min.css through the host url of Forge, like we usually do with Forge Viewer app. This [blog](#) tells in detail, including the test code project. (appreciate for my colleague **Frederic Duszyk** for helping in this researching)



```

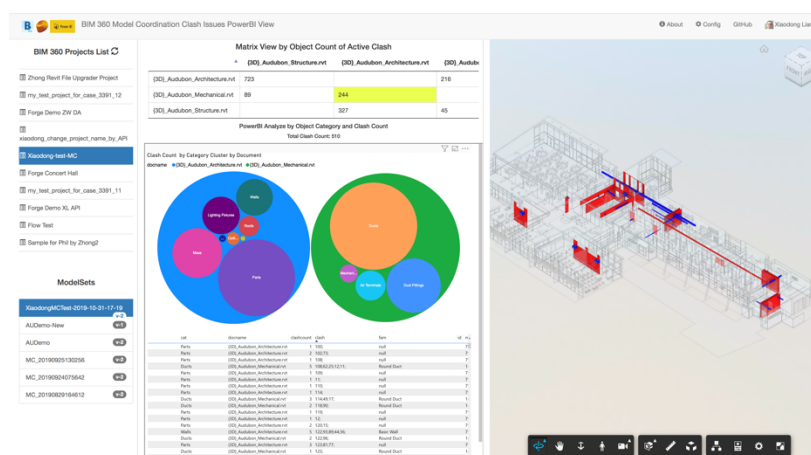
130  src > TS visual.ts > {} powerbi.extensibility.visual > PowerBI_ForgeViewer_Visual > loadForgeViewerScriptAndStyle > <function> > forge
131
132  private async loadForgeViewerScriptAndStyle(): Promise<void> {
133
134      return new Promise<void>((reslove, reject) => {
135
136          let forgeviewerjs = document.createElement('script');
137          forgeviewerjs.src = 'https://developer.api.autodesk.com/modelderivative/v2/viewers/viewer3d.js';
138
139          forgeviewerjs.id = 'forgeviewerjs';
140          document.body.appendChild(forgeviewerjs);
141
142          forgeviewerjs.onload = () => {
143              console.info("Viewer scripts loaded");
144              let link = document.createElement("link");
145              link.rel = 'stylesheet';
146              link.href = 'https://developer.api.autodesk.com/modelderivative/v2/viewers/style.min.css';
147              link.type = 'text/css';
148              link.id = "forgeviewercss";
149              document.body.appendChild(link);
150              console.info("Viewer CSS loaded");
151
152              reslove();
153          };
154
155          forgeviewerjs.onerror = (err) => {
156              console.info("Viewer scripts error:" + err );
157              reject(err);
158          };
159      });
160  };
161
162

```

Power BI Embedded in Web App with Forge Viewer

The last scenario is to imbed Power BI report inside an app. Power BI provides a web component that can be inserted to a web page. As you remember, BIM 360 insights can embed Power BI report as a partner card. Actually, it is also using the same technology: Power BI component.

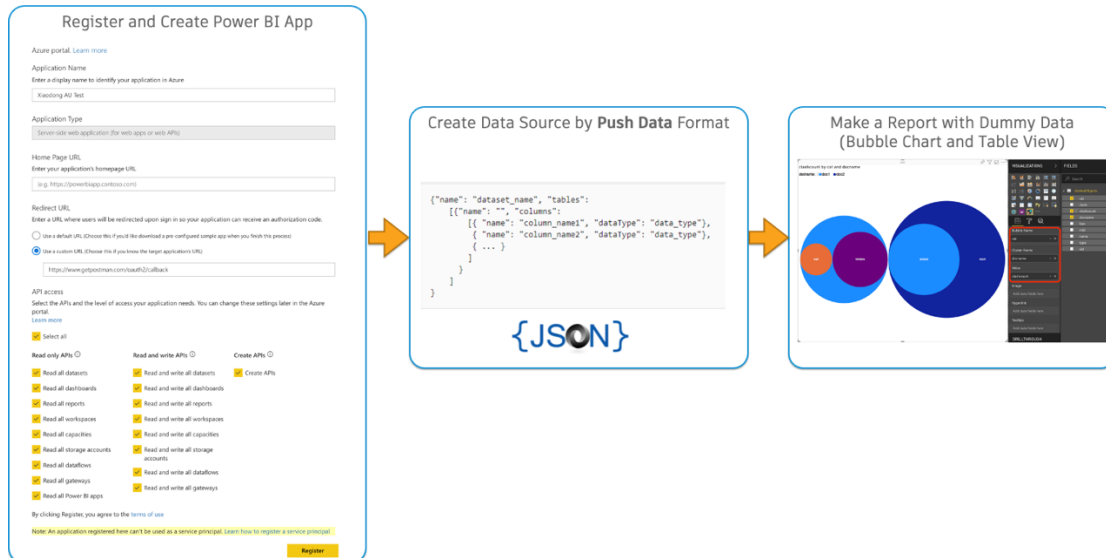
Here is a comprehensive sample. It calls BIM 360 API to get model coordination collection of one project. As mentioned before, model coordination will check the aggregated models to detect the clash of the designs. This sample extracts the clash data of one pair of documents in the coordination space, dump the properties of the clash elements and send to Power BI report template. The report presents a bubble chart and a table view. The bubble chart indicates how many elements are clashed in one documents, clustered by the element types. e.g. windows, doors, ducts pipes, walls, framing, air terminals etc. The bubble chart and table view are associated, once the one bubble is selected, the summarized table rows will be highlighted. The corresponding elements will also be highlighted in Forge Viewer.



Prerequisites of Power BI

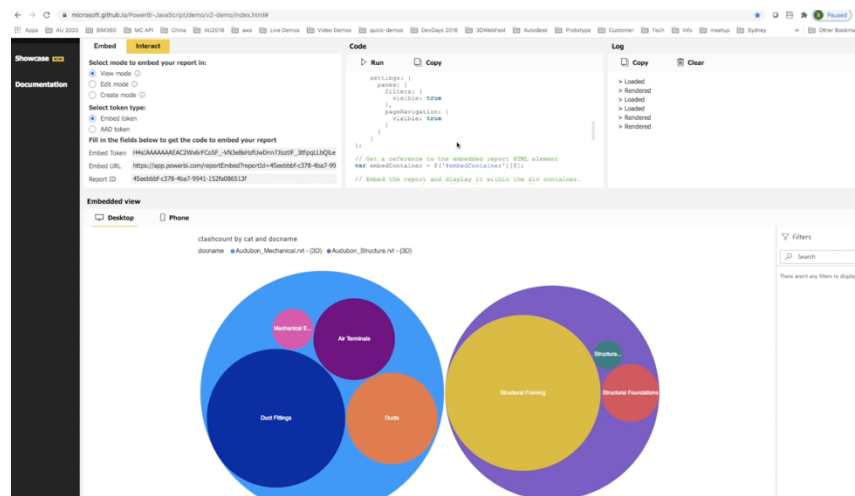
To work with the imbedded component of Power BI, you will also need to [create Power BI app](#). In this app, configure which permissions for this app. Such as manipulating report, dataset, etc.

Next, create a data source for the report. Obviously, since the sample is web integration, all workflow of data is running on the cloud, it will not be suitable to use file as data-source. In addition, if using database type, there is a [limitation of refresh rate](#). That is to say, in one day, the report can only be refreshed with limit times if using database as source, but in this sample, it will refresh the BIM 360 clash data on-demand, so the refresh rate is not fix. So I selected [Push-Data type](#). We need to create a Push-Data resource, and design a report based on the dummy Push data. In the report, one bubble chart and one table view will be deployed.

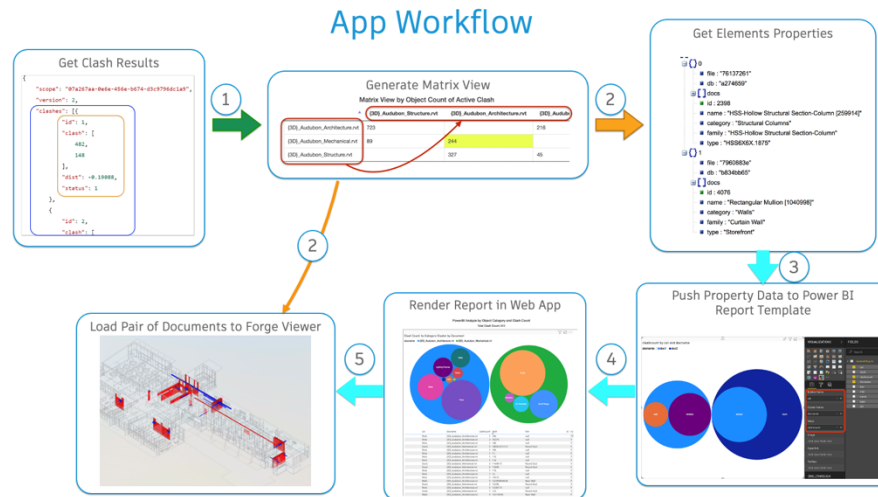


Power BI Embedded Component

The [PowerBI tutorial](#) has a very nice harness, this tool shows how to initialize the imbedded report, how to delegate events such as visual selected and some configurations.



when the app is running , it will firstly get all clash data of one coordination space by Model Coordination API, by these data, it generates matrix view of clash. i.e. the clashes elements of every pair of documents. if clicking one pair, the corresponding documents will be loaded into Forge Viewer. In the same time, the code will build the data array of elements properties and update the Power BI data resource, pushing property data to Power BI report template, finally it will render the updated report.



More words about tricks: we use Push-Data as data source because it has not refresh limit. We use Model Coordination API to extract clash data of model coordination space. And Index Property API to extract elements properties.

In this sample, we do not ask Power BI user to login. To implement this, we need to follow these steps to get Open Access Token, accessing the report like administrator of the PowerBI space.

One more trick is about associating the selecting events of Power BI and Forge Viewer. If it is table view, it is straightforward, we can get the table data directly, and get clash id, by the clash id, get clashed elements, finally isolate the elements in Forge Viewer. While if it is bubble chart, when a bubble is selected, we will get summarized data, find corresponding table rows and finally get clashed elements. Appreciate for my colleague [Jan Liska](#) for helping on this point.

[This blog](#) tells more.

So summary, when work with Power BI + BIM360 data, the possible scenarios would be:

- No programming
 - Data Connector
 - Partner Card
- Extract/Export functionalities
 - A little programming
 - Call BIM 360 API by Restful tool such as Postman, exporting data in Json format
 - Use R scripts/templates to execute the extracting
- More programming
 - By more advanced programming such as Node.js. customized based on BIM 360 Export Tool
- Comprehensive programming
 - Embed Forge Viewer inside Power BI report
 - Embed Power BI report inside a Forge app