Session 2.4

Enabling model viewer in Power BI using APS Viewer API

Lei He, KBR

Class Description

Power BI is the popular data engineering and visualisation tool for the enterprise users. It is also a great tool for managing / visualising the project / digital delivery information within the AEC industry. The presentation will demonstrate how to establish a Power BI custom visual that incorporates a 3D model viewer from Autodesk Platform Services (APS) Viewer API (formerly known as Forge Viewer).

The presentation will cover the following topics:
• Overview of the APS API and APS Viewer API
• Create your own Power BI custom visual with APS model viewer – a not-from-scratch approach
• Prepare the model properties using a Power BI custom data connector
• Connecting to the models stored on Autodesk Construction Cloud (ACC), especially for the newly establish ACC APAC region.
• Data modelling in Power BI to create your own storying telling experiences that interacts with model geometry
• Discussion of the opportunities to deploy the custom visual and the solution at the enterprise level

At the end of this presentation, the audience will have the basic understanding of how to enable the APS model viewer in a Power BI report from multiple available open-source repositories.

An open-source GitHub repository will be shared to help the audience to start and follow.

Lei He, KBR

About the Speaker:

Lei He
National Digital Engineering Lead, KBR

My passion lies in data engineering, data visualization, design automation, and system integration. I enjoy the challenges when developing and delivering the digital solutions targeting specific project information requirement, by integrating the systems such as BIM, GIS, and CDE.

Lei He, KBR

Introduction

Please visit the GitHub repo aps-powerbi-tools-apac prepared for the presentation. This is the collection of tools to allow us to establish APS model viewer in Power BI report. By connecting both model viewer and elements properties from ACC, you can customise your own story-telling experience inside Power BI.

Those four tools are:

- **aps-viewer-visual-apac** - A Power BI custom visual which integrates APS Model Viewer. This has been updated from its original repo to support models from ACC APAC region by acquiring the access token from **aps-shares-app**.
- **aps-shares-app** - A web app to register the model sharing from ACC and feeding the model access token back to APS Viewer custom visual.
- **aps-props-connector-apac** - This is the Power BI custom data connector to directly read model properties from all elements directly in Power BI using APS Model Derivative API. It has been updated to support ACC APAC region.
- **vscode-forge-tools-apac** - Visual Studio Code extension for accessing Autodesk Platform Services and data. This tool can help the users to get all inputs to Power BI APS Model Viewer, such as the unique identifier of the model (URN), model view id, and the properties of all elements. It has been updated to support ACC APAC region.

To prepare you own fast track, please clone or download the repo from GitHub - visualphyzx/aps-powerbi-tools-apac: Integrating APS Viewer and ACC data connector in Power BI.

Acknowledgement

All source codes used by this presentation were originally from two open-source projects developed and shared by Autodesk Platform Services and Petr Broz. Great thanks for their contributions.

- GitHub - autodesk-platform-services/aps-powerbi-tools: Collection of tools for accessing Autodesk Platform Services design data - 2D/3D views as well as element properties - from Power BI reports.

- GitHub - petrbroz/vscode-forge-tools: Visual Studio Code extension for accessing Autodesk Forge services and content.

Tools

Power BI Custom Visual

Power BI visuals SDK allows the users to create customised data visualisation experiences based on any JavaScript libraries such as D3 and jQuery. We will use APS Viewer SDK to enable a 3D model viewer in your Power BI report.

Microsoft introduces what the custom visuals are from the following links:

- Main sources for acquiring Power BI custom visuals - Power BI | Microsoft Learn
- Develop custom visuals in Power BI - Power BI | Microsoft Learn

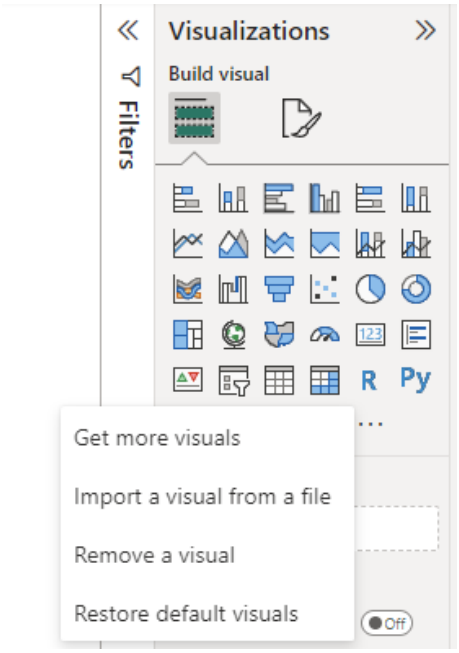


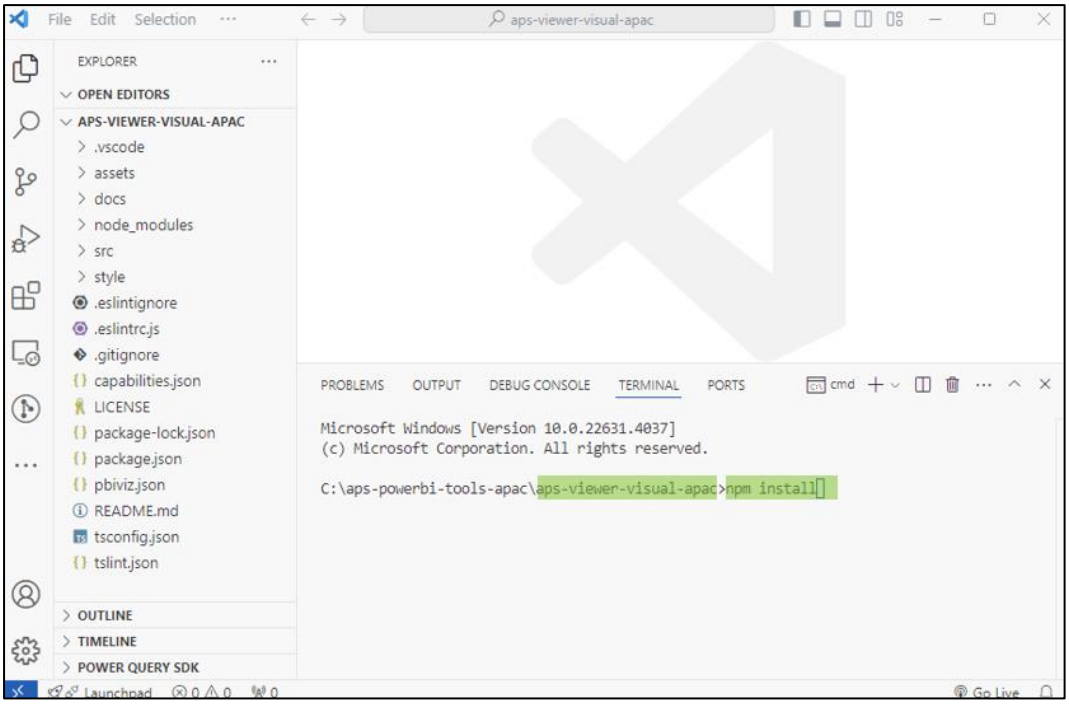
Figure 1: Options to manage custom visuals from Power BI

Please follow the tutorial of Develop a Power BI circle card visual as an example - Power BI | Microsoft Learn to review the basic requirement and steps to develop a Power BI Visual.

- A **Power BI Pro** or Premium Per User (PPU) account. If you do not have the one provided by your company, you can still develop and test your Power BI report if your **subscription of Microsoft 365** allows you to use Power BI Desktop and web portal. You just cannot share your Power BI report in a private way.
- Visual Studio Code (VSCode) - Download Visual Studio Code - Mac, Linux, Windows
- Install node.js - Node.js — Run JavaScript Everywhere (nodejs.org)
- Install pbiviz tool using npm in PowerShell or Windows Command Line
  `npm i -g powerbi-visuals-tools@latest`

Installing aps-viewer-visual-apac

1. Using VSCode to open the folder of aps-viewer-visual-apac under the cloned or downloaded presentation repo.
2. Open the command line tool from VSCode, ensure **aps-viewer-visual-apac** is your current working folder.
3. Run command:
   `npm install`



4. After the package installation finishes, run command:
   `pbiviz start`
   You will expect a lengthy list of messages like below:

```
C:\aps-powerbi-tools-apac\aps-viewer-visual-apac>pbiviz start
 info   powerbi-visuals-tools version - 5.4.3
 warn   Local valid certificate not found.
 info   Checking global instance of pbiviz certificate...
 info   Copy server certificate from global instance of pbiviz...

 info   Starting server...
 info   Start preparing plugin template
<i> [webpack-dev-server] Project is running at:
<i> [webpack-dev-server] Loopback: https://localhost:8080/
<i> [webpack-dev-server] On Your Network (IPv4): https://10.150.12.76:8080/
<i> [webpack-dev-server] Content not from webpack is served from 'C:\aps-powerbi-tools-apac\aps-view
tmp\drop' directory
 info   Server listening on port 8080
 info   Finish preparing plugin template
 info   Start packaging...
 info   Finish packaging
Webpack Bundle Analyzer saved report to C:\aps-powerbi-tools-apac\aps-viewer-visual-apac\webpack.sta
assets by path ../build/ 5.68 KiB
```

- The warning message of "Local valid certificate not found" can be ignored.
- Current custom visual is served from the working PC for development, at https://localhost :8080.
- For further debugging and development of custom visual using Power BI online services, please follow this tutorial [Develop a Power BI circle card visual as an example - Power BI | Microsoft Learn]

5. You can skip the last step, but any error messages generated from Step 4 can help you debug the code before packaging. Form the terminal window inside VSCode, use keyboard combination Ctrl+C to stop the service.

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                    cmd

    ./node_modules/powerbi-visuals-utils-formattingmodel/lib/utils/FormattingSe...(truncated) 1.
 generated]
  modules by path ./src/*.ts 25.5 KiB
    ./src/visual.ts 20.5 KiB [built] [code generated]
    ./src/settings.ts 1.34 KiB [built] [code generated]
    ./src/viewer.utils.ts 3.64 KiB [built] [code generated]
  ./.tmp/precompile/visualPlugin.ts 1.14 KiB [built] [code generated]
  css ../../Users/k147904/AppData/Roaming/npm/node_modules/powerbi-visuals-tools/node_modules/cs
!../../Users/k147904/AppData/Roaming/npm/node_modules/powerbi-visuals-tools/node_modules/less-lo
leSet[1].rules[2].use[2]!./style/visual.less 526 bytes [built] [code generated]
webpack 5.91.0 compiled successfully in 715 ms

 info   Stopping server...
<i> [webpack-dev-server] Gracefully shutting down. To force exit, press ^C again. Please wait...

C:\aps-powerbi-tools-apac\aps-viewer-visual-apac>
```

6. Run command line to create a custom visual deployable package.
   `pbiviz package`
   A. pbiviz file should be found under "**dist**" folder after the process finishes.

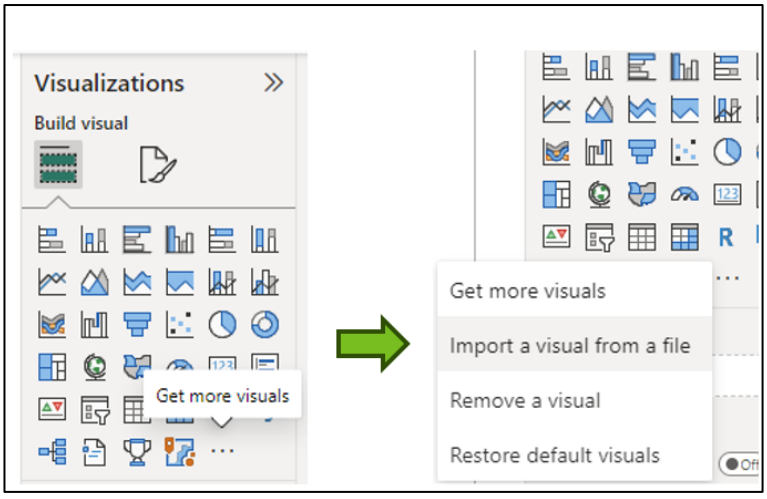
   aps_viewer_visual_apac_a4f2990a03324cf79eb44f982719df44.0.0.8.1.pbiviz

You can ignore the errors and warnings if the build completes successfully. We can resolve those errors and warnings when deep diving into Power BI Custom Visual development.

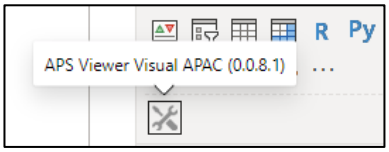
```
error  Linter found 28 errors and 0 warnings. Run with --verbose flag to see details.
info   Lint check completed.
warn   Local valid certificate not found.
info   Checking global instance of pbiviz certificate...
info   Copy server certificate from global instance of pbiviz...
info   Start preparing plugin template
info   Finish preparing plugin template
info   Start packaging...
info   Package compression enabled
info   Package created!
info   Finish packaging
Webpack Bundle Analyzer saved report to C:\aps-powerbi-tools-apac\aps-viewer-visual-apac
l

done   Build completed successfully
warn   Visual doesn't support some features recommended for all custom visuals:
```

7. Lauch Power BI Desktop and create an empty report. Import the pbiviz file generated from last step.

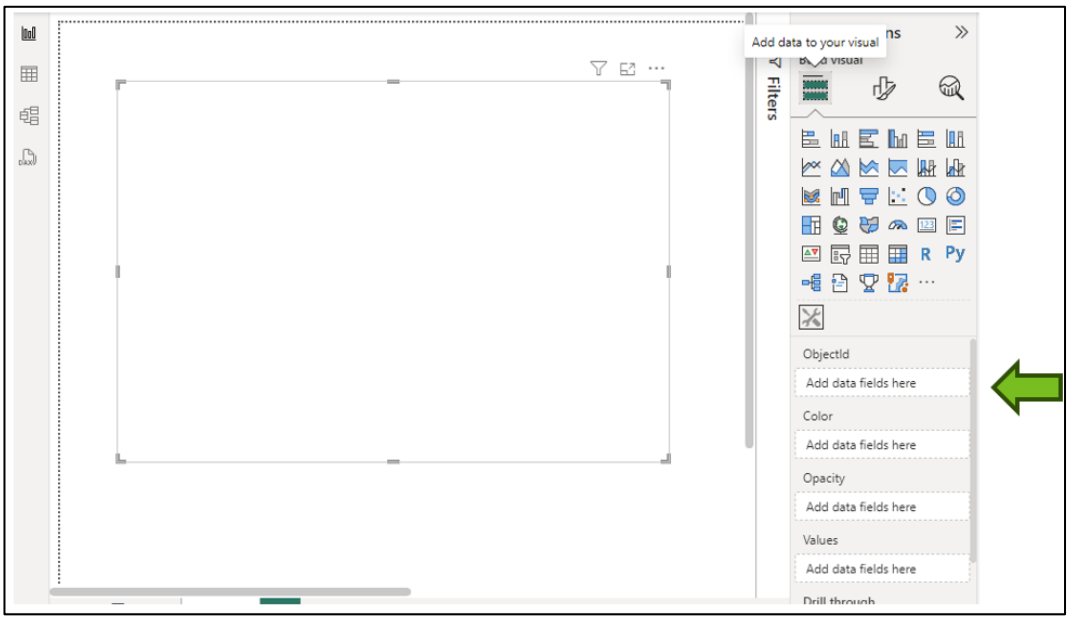


If successful, we are expected to see the icon of the custom visual listed on the Visualisations Pane.

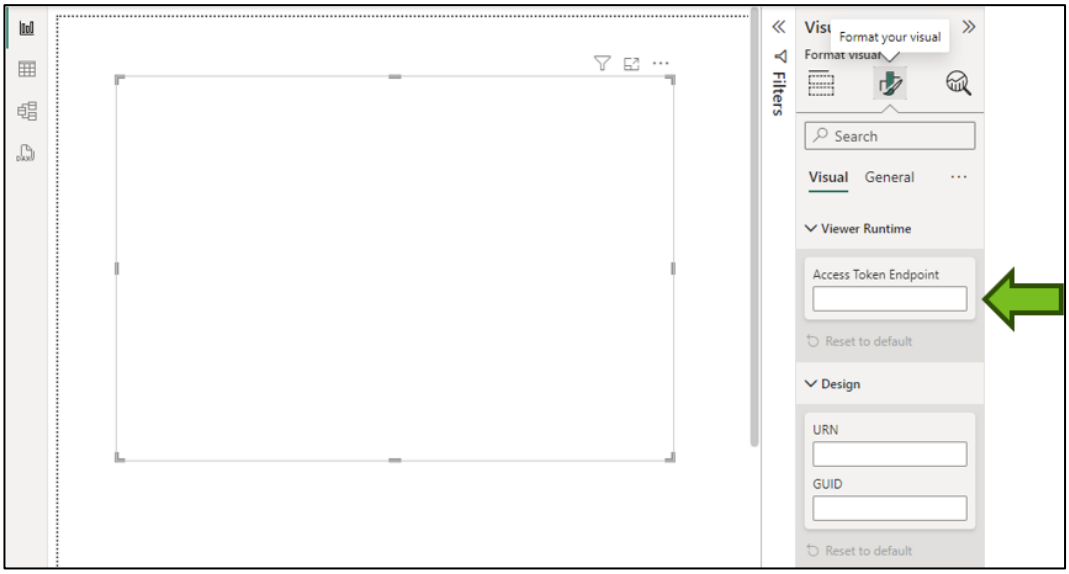


Click the icon will add the visual into the Report View of Power BI Desktop. With the visual selected, check the inputs requirement from "Data" view and "Formats" view.

Lei He, KBR



Providing data inputs from data model – we will use the custom data connector from **aps-props-connector-apac.**



Providing format inputs – we will facilitate endpoint and model identifier (URN) from **aps-shares-app.**

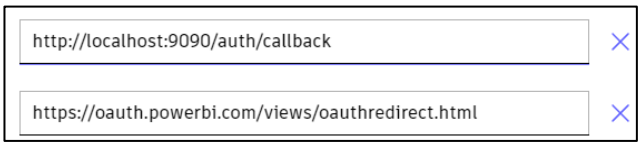
Keep the Power BI Desktop open, we can fill the inputs generated from next section.

Installing aps-shares-app

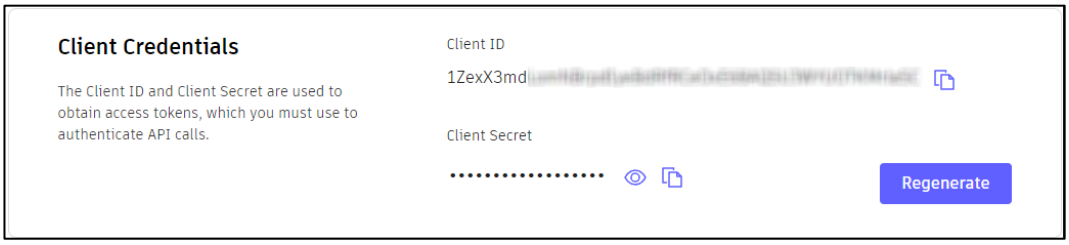
Lei He, KBR

Web app aps-share-app is created to provide the required inputs to **aps-viewer-visual-apac**. The installation of this app can be summarised as follows:

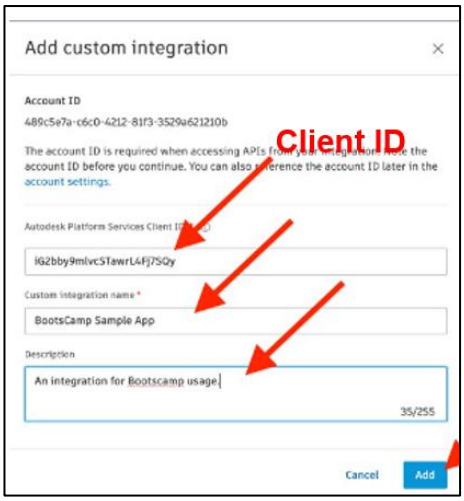
1. Create the APS app on Autodesk Platform Services (formerly Forge) – please follow the tutorial from APS Getting Started | Autodesk Platform Services Tutorials. Register http://localhost:9090/auth/callback as one of the **callback URL**. Please note **port number** (9090 in our case) needs to be the same as what **aps-shares-app** occupies when deployed locally. Developers can register multiple callback URL from the same app. *We will re-visit this section at the later part.



   Record the **Client ID** and **Client Secret** from Client Credentials section.

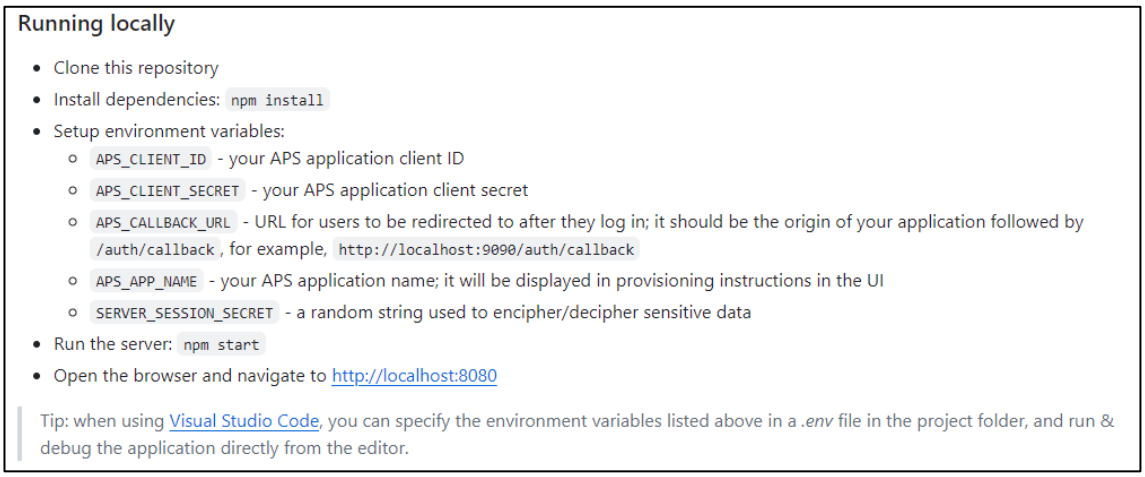


   Register Custom Integration on ACC Admin portal to grant the connection of the registered app. **Client ID** recorded earlier needs to be provided during the registration process.

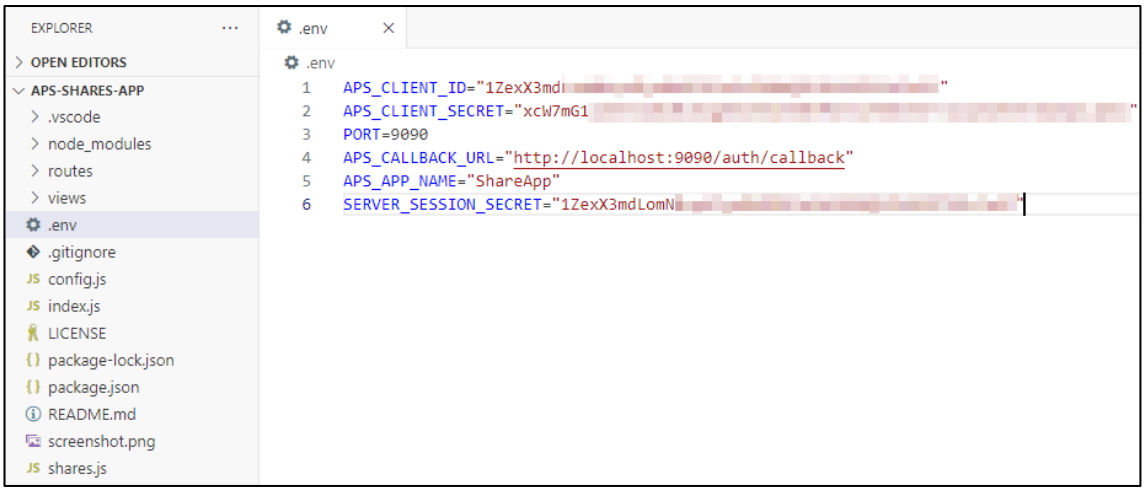


2. Navigate to the folder aps-shares-app (cloned or downloaded previously), open the folder in Visual Studio. Follow README.md file – **Running locally** Section.

Lei He, KBR

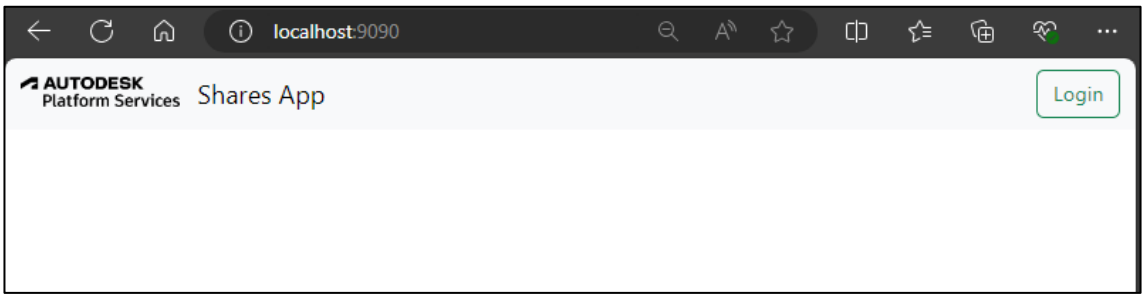


Create a .env file under the aps-shares-app folder. Open it in VSCode or any text editor. Add the following environment variables as the example below:

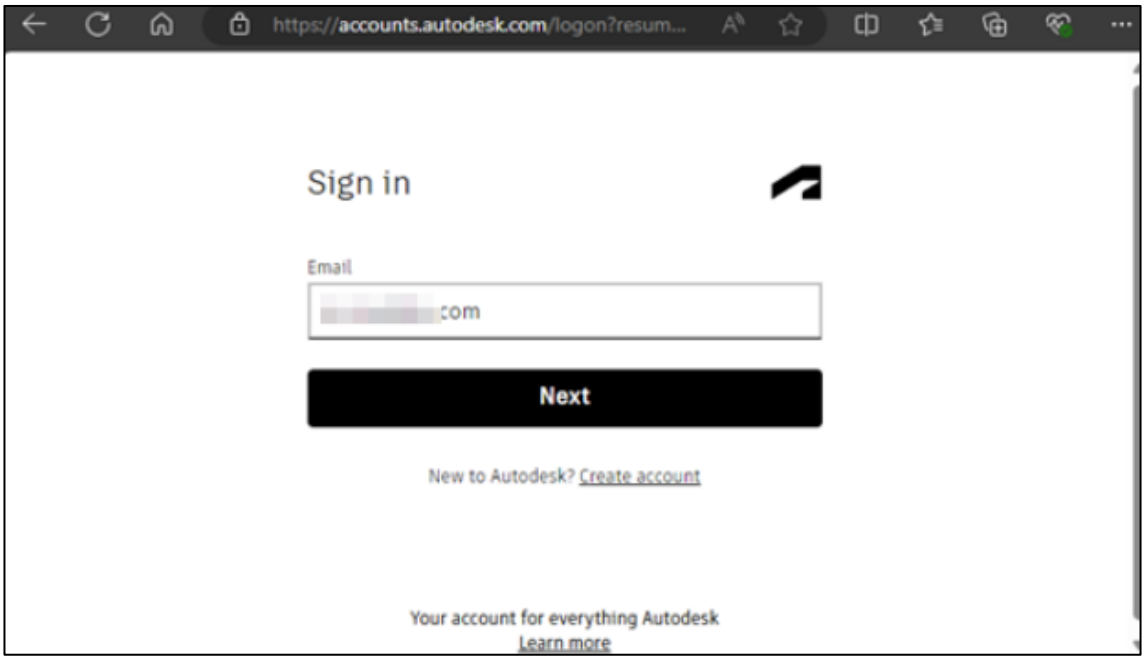


The values of APS_CLIENT_ID, APS_CLIENT_SECRET, APS_CALLBACK_URL shall be the same as **Client ID**, **Client Secret**, **Callback URL** from the APS App registration (previous step).

3. Running the app by using command
   `npm start`
   If successful, we shall see the message as below:
   `Server listening on port 9090...`
   Launch your Internet browser (Edge or Chrome), the type in URL address **localhost:9090**
   You will expect to see the following page:

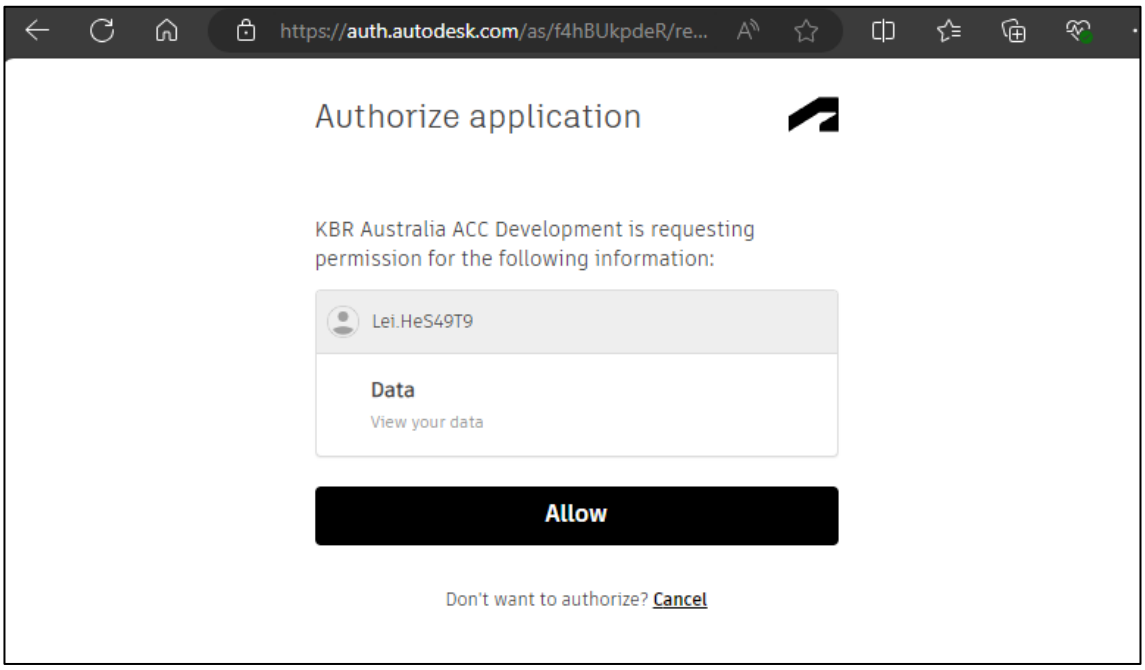


Click **Login** button will lead you to the sign-in window to ACC with Autodesk account. If SSO is enabled, you can use your company account to sign in.

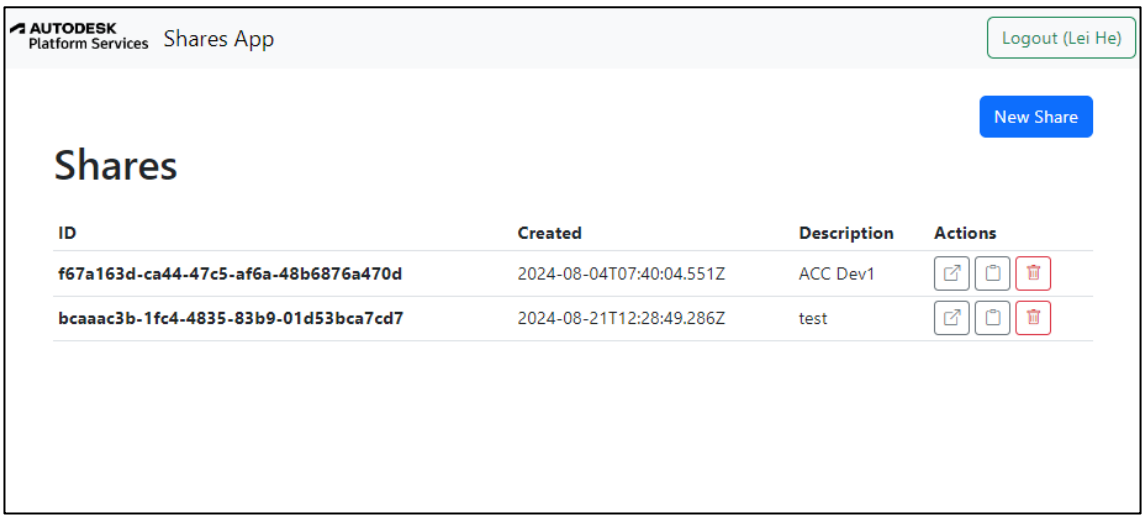


After signing in, the app will ask your to grant it the "**Data:Read**" permission to read what you are available to view from ACC. The scope is set at the time of authentication. It is to make sure the access token the app is requesting only have the necessary permission as it requires. The details of **scope** concept can be found from [Scopes | Authentication (OAuth) | Autodesk Platform Services].

Lei He, KBR

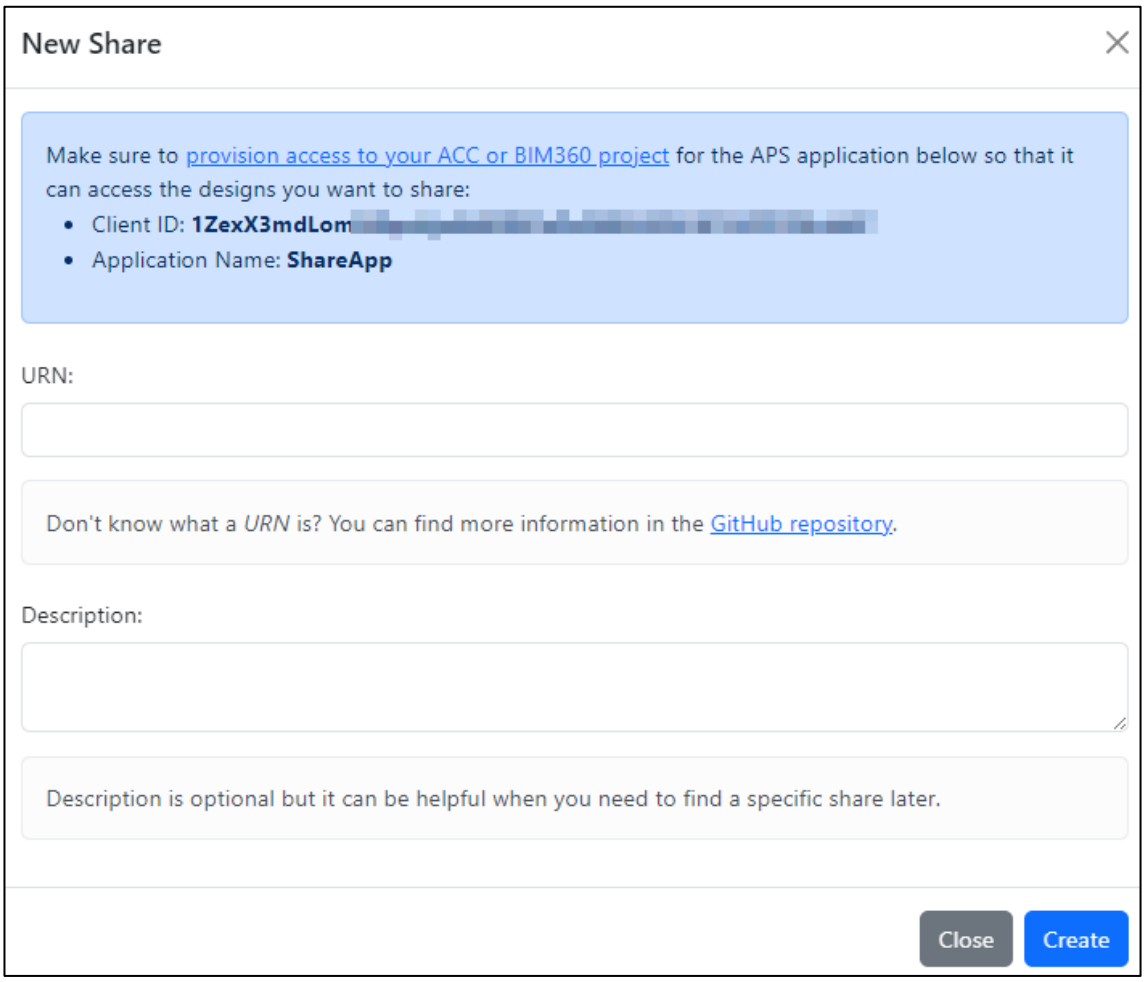


After allowing the permission, the user will be redirected back to the app. Users may be able to see all "**Shares**" created previously from this app, or other applications but may share the same **Client ID.**



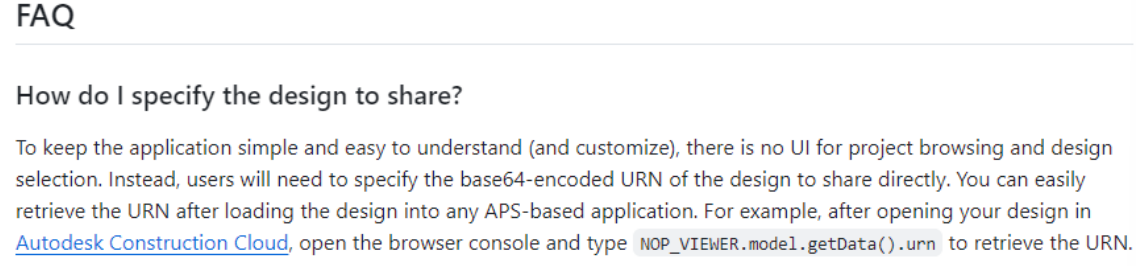
4. Clicking "**New Share**" button to start the process. A **Share** is a model at the specific version which you want to be accessible from Power BI.

Lei He, KBR



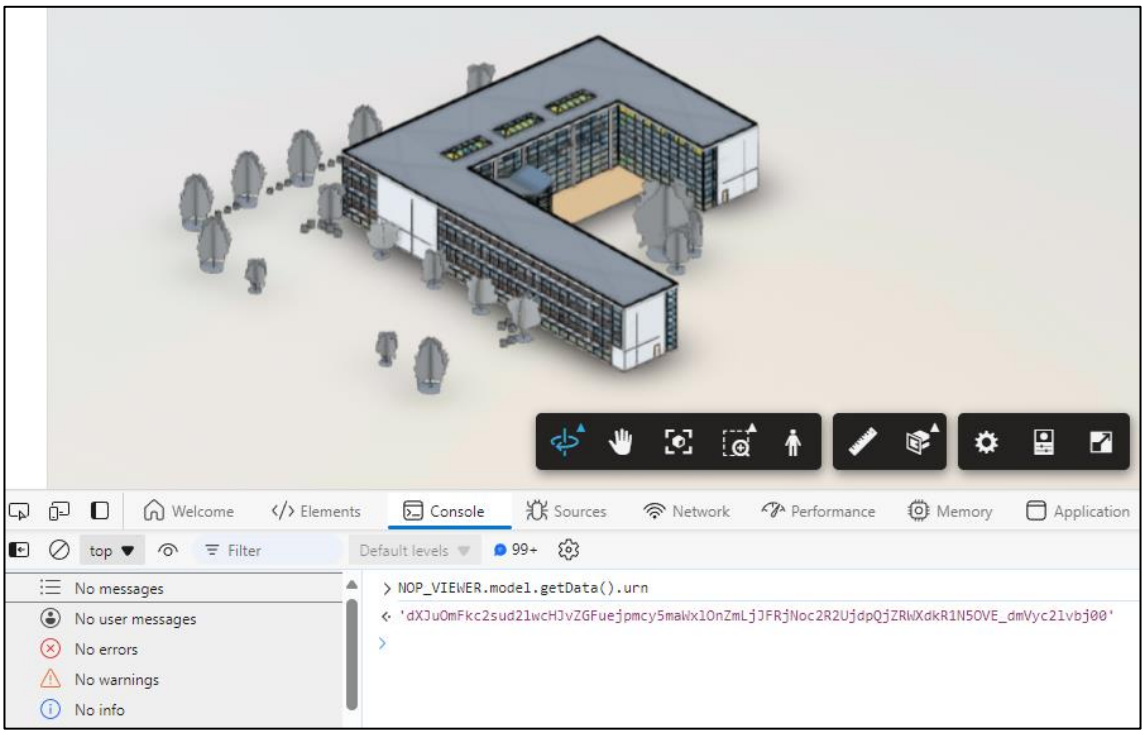
The dialog informs the user – to be able to connect to the model stored on ACC, the permission to access must be granted to this app – which already completed on Step 1.

URN is the identifier to a specific version of a specific model (a .rvt or .dwg file for example). There are many ways to get the URN for our models on ACC. We can follow the instruction of README.md to get the value of URN.

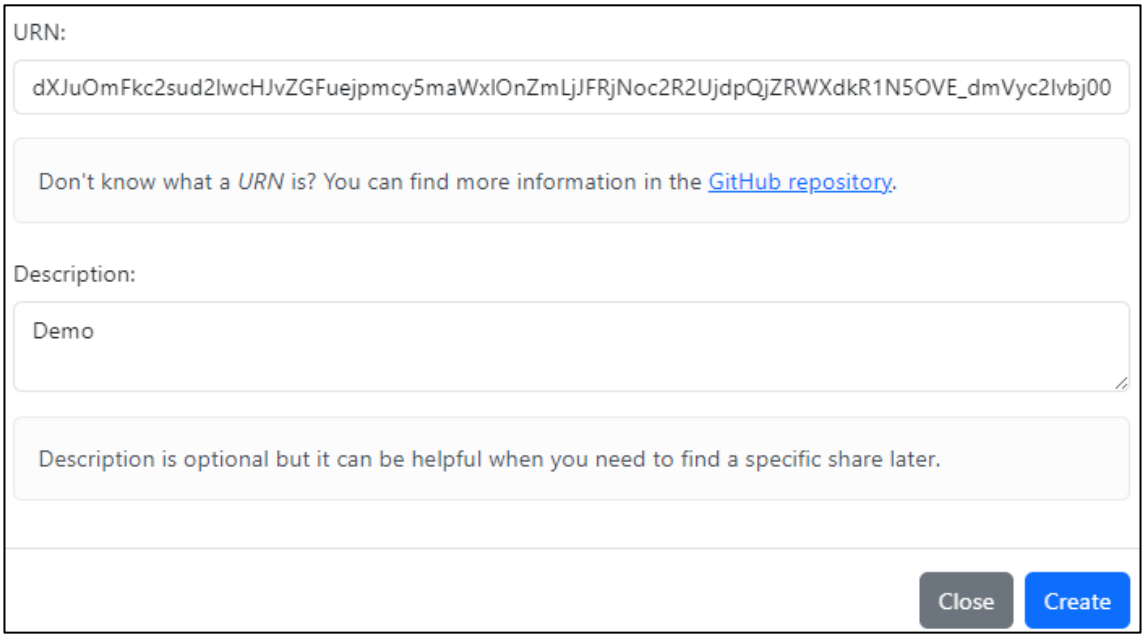



```
NOP_VIEWER.model.getData().urn
```

Lei He, KBR

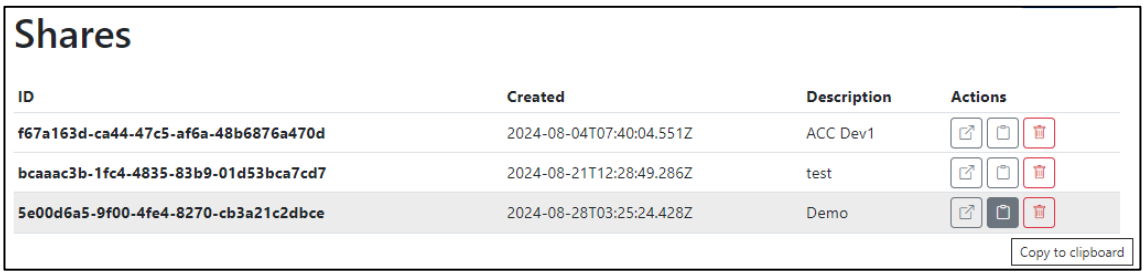


You will find the tool **vscode-forge-tools-apac** - A VSCode extension will provide you better user experience (and more features) to get the URN value.
Copy and paste the URN value into **New Share** dialog and add the description if needed.



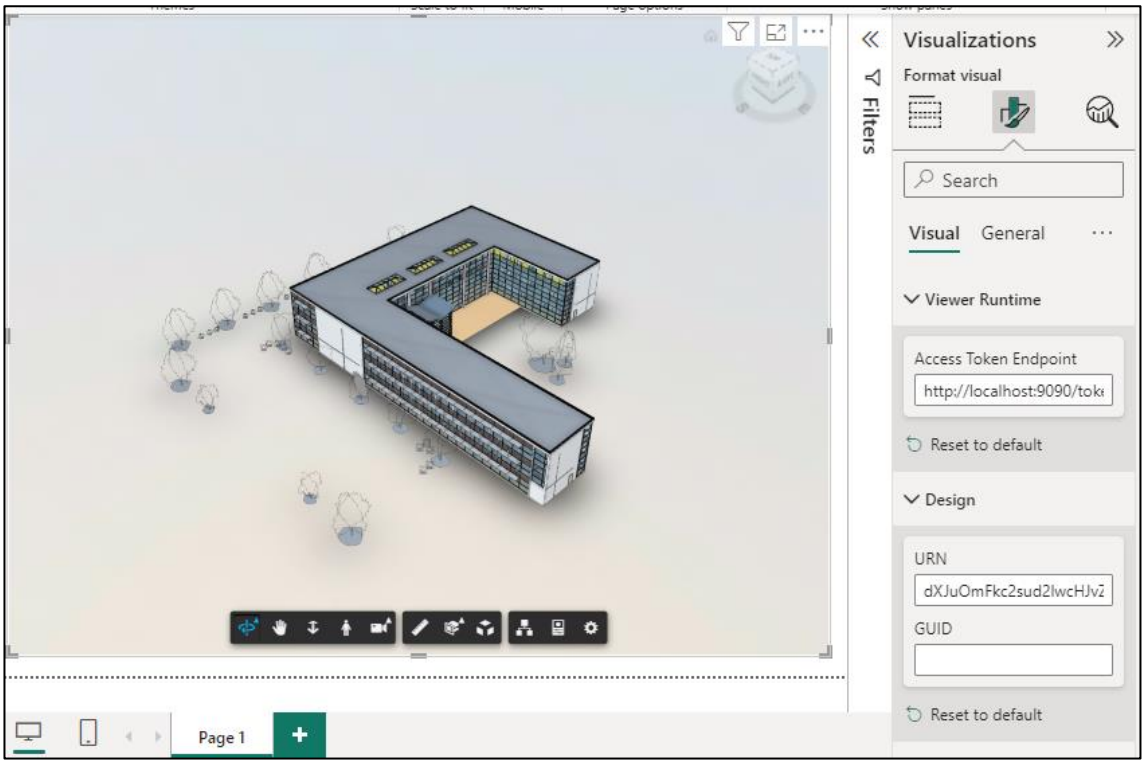
Click **Create** then a new **Share** to the demo model is now created. Click **Copy** button to get the **Endpoint** for Power BI APS model viewer to use.

Lei He, KBR



Please record both **URN** and **Endpoint** from previous steps.

5. Keep the Aps-Shared-App live from previous steps. From the Power BI Desktop where we have imported the APS Model Viewer custom visual, fill the **Endpoint** and **URN** value. The viewer in Power BI shall be able to load the model from ACC.

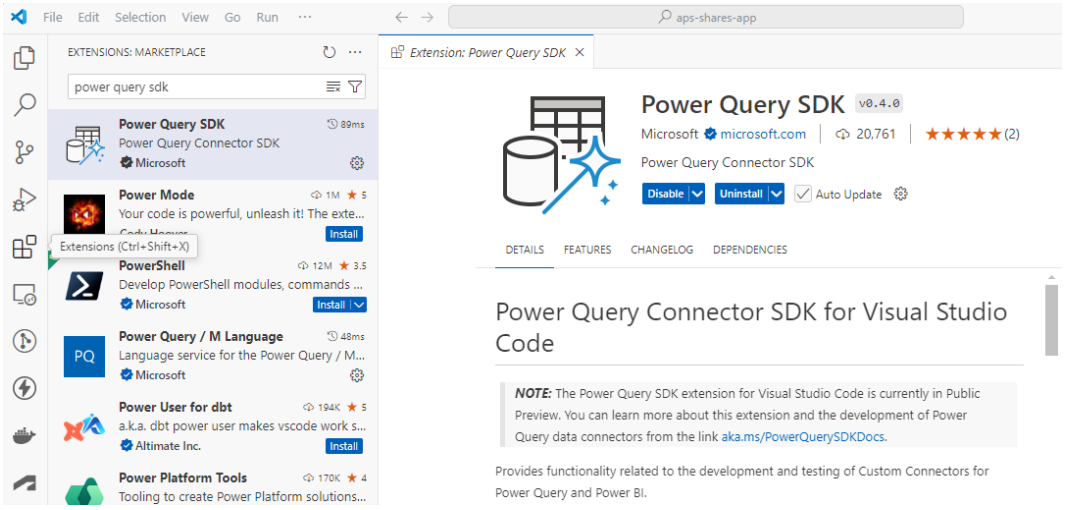


Connect ACC Model Data

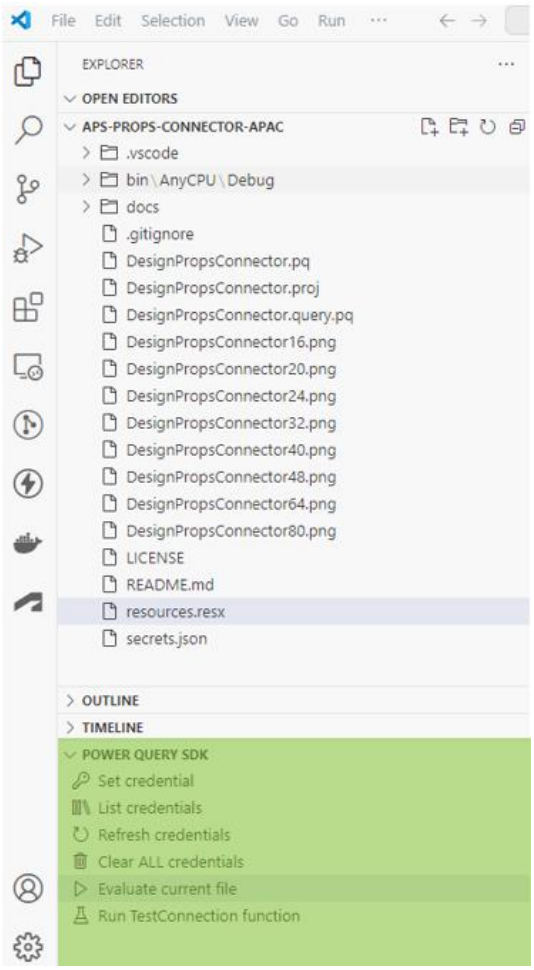
There are many ways to read the properties of model elements when we have the model saved on ACC. For this presentation, we are introducing **aps-props-connector-apac** – a custom data connector in Power BI which can directly retrieve the model data from ACC utilising APS Model Derivative API.

1. Install **Power Query SDK for Visual Studio Code.** Go to **Extensions** and search for "**power query sdk**." Select it and install.

Lei He, KBR



2. Use a separate VSCode Window to open the folder of **aps-props-connector-apac.** If Power Query Connector installed correctly, VSCode shall understand the current of **aps-props-connector-apac** is a Power Query Connector project and then launch the User Interface.

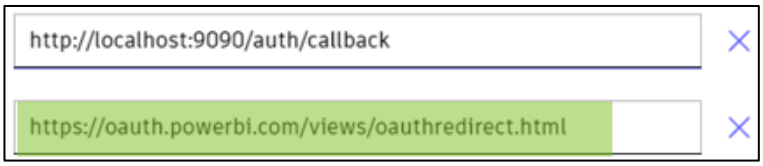


Lei He, KBR

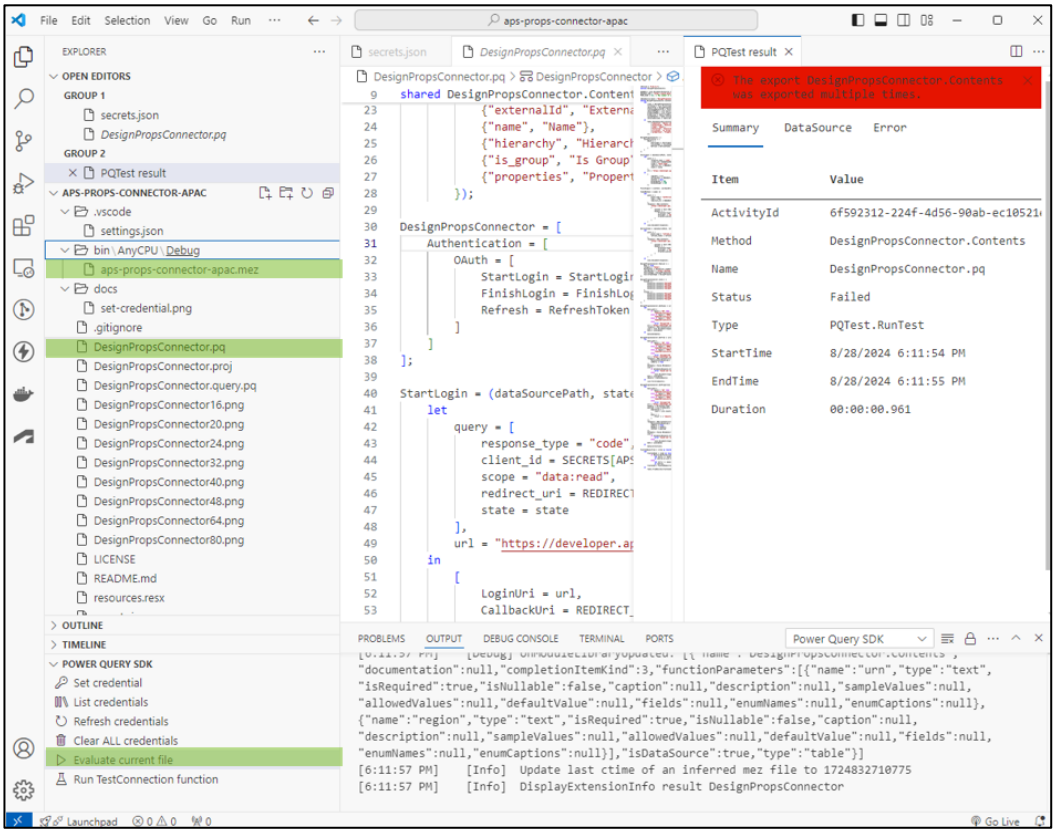
3. Create a secrets.json file in the root folder of **aps-props-connector-apac**, populate it with your APS app **Client ID** and **Client Secret**.

```
{
    "APS_CLIENT_ID": "<your client id>",
    "APS_CLIENT_SECRET": "<your client secret>"
}
```

4. We will need to register an additional Callback URL to APS app on https://aps.autodesk.com/myapps. This is for authenticating users who will request model properties when using the data connector.



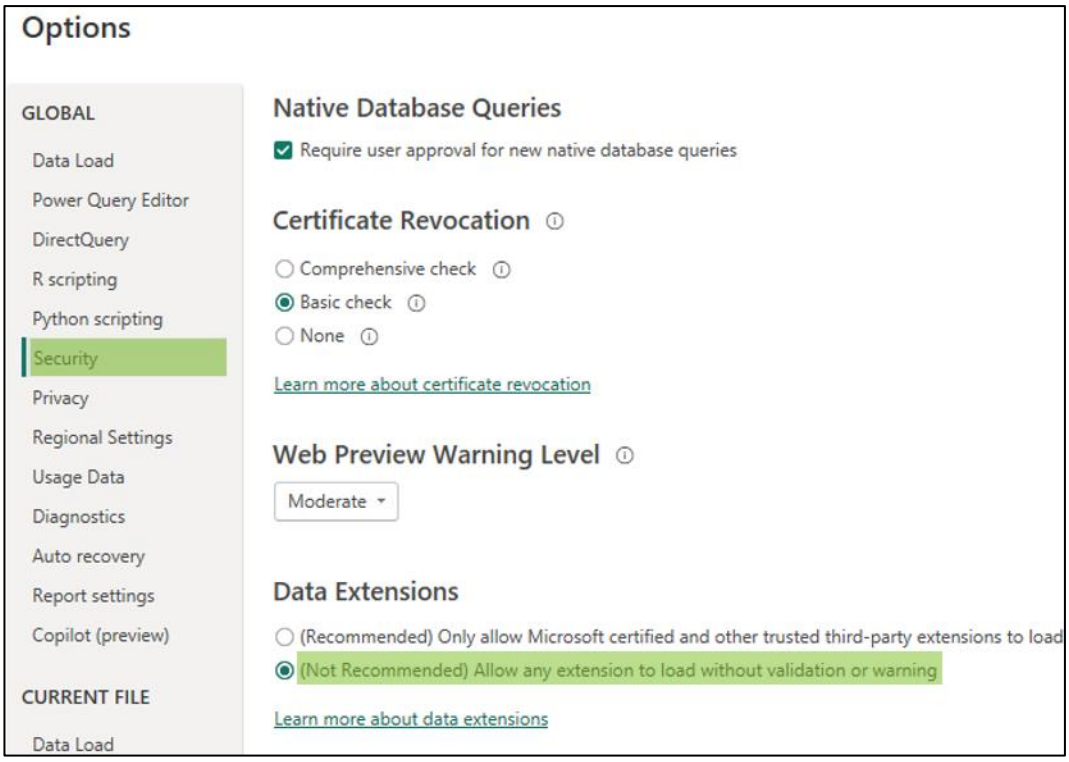
5. Select **DesignPropsConnector.pq** file and click **Evaluate current file**. It will fail as we have not provided authentication information, but the data connector (**aps-props-connector-apac.mez**) will still be built and saved into **bin** folder.



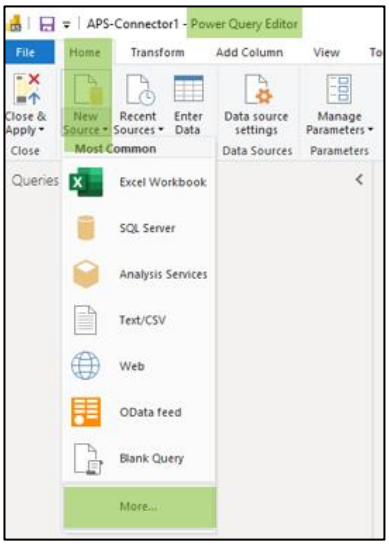
Copy **aps-props-connector-apac.mez** to ***[Documents]\Microsoft Power BI Desktop\Custom Connectors.***

Lei He, KBR

6. In the previous Power BI Report, which we already have the model viewer available, check the **Options** from **Options and Settings**. Make sure we are **allowing the extensions to load without validation or warning**.

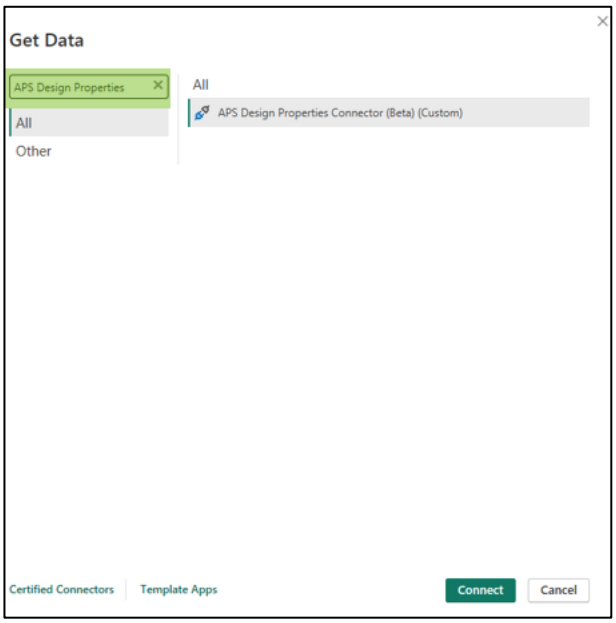


7. To load the model properties from the same model connected in APS model viewer, click **More** from **New Source** on **Home** ribbon of **Power Query Editor.**

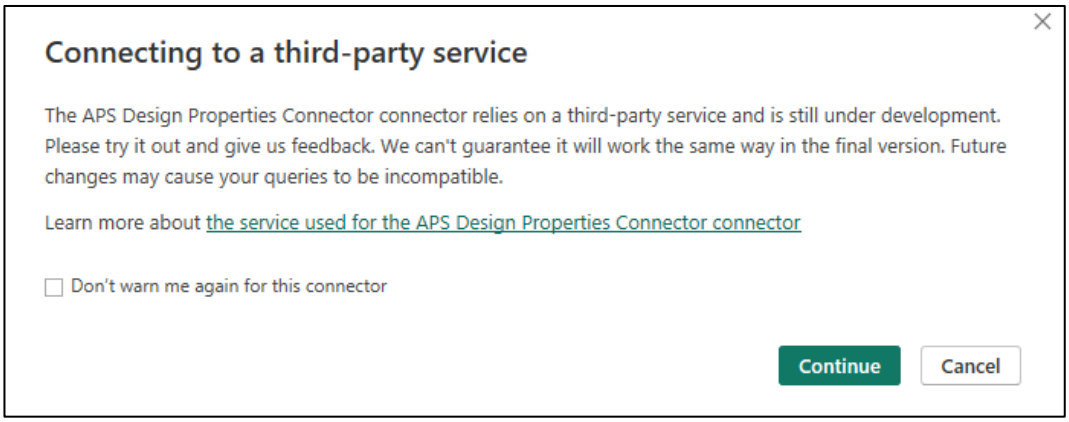


Search for **APS Design Properties** in the dialog box, select the custom connector we created previously and click **Connect** button.

Lei He, KBR



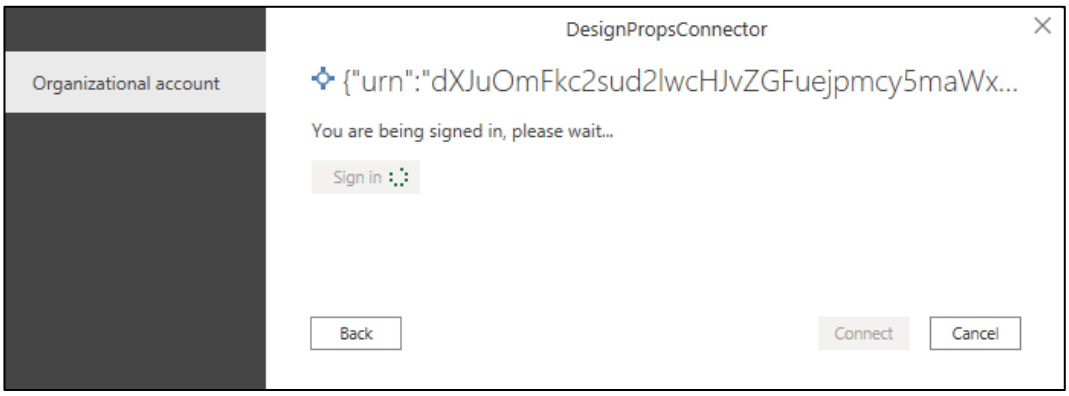
Ignore and Continue from a dialog box with warning message.



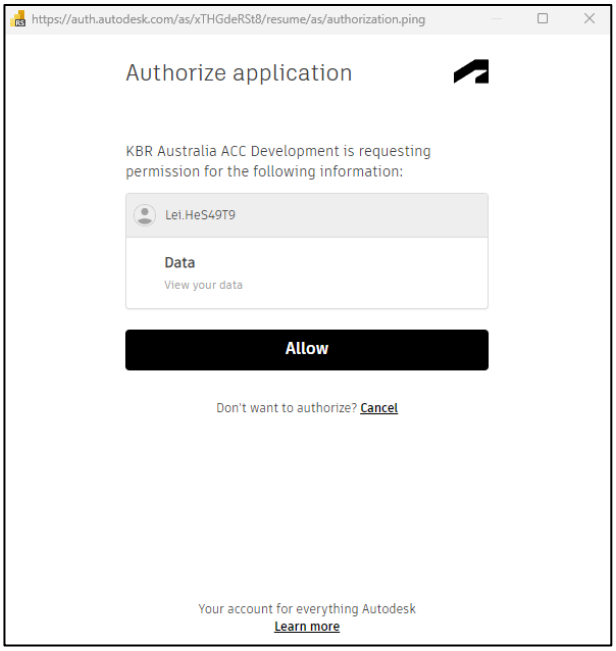
Fill the **URN** of the same model which we already connected in Power BI. Use **APAC** for the value of region.

A new dialog box popping up to ask you for **Sign in** to ACC. Follow the steps to authenticate yourself as the user with at least "Read" permission of the model

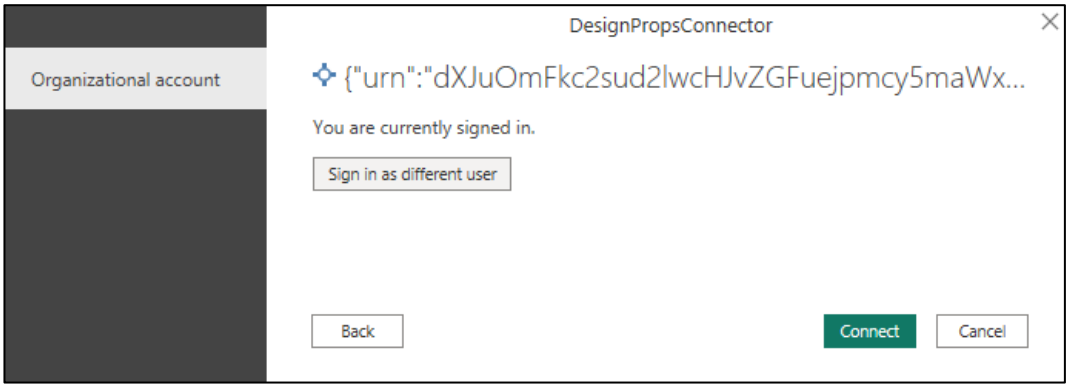
Lei He, KBR



A similar dialog will be shown to request user's permission to read the data for the user as we have seen from aps-share-app.

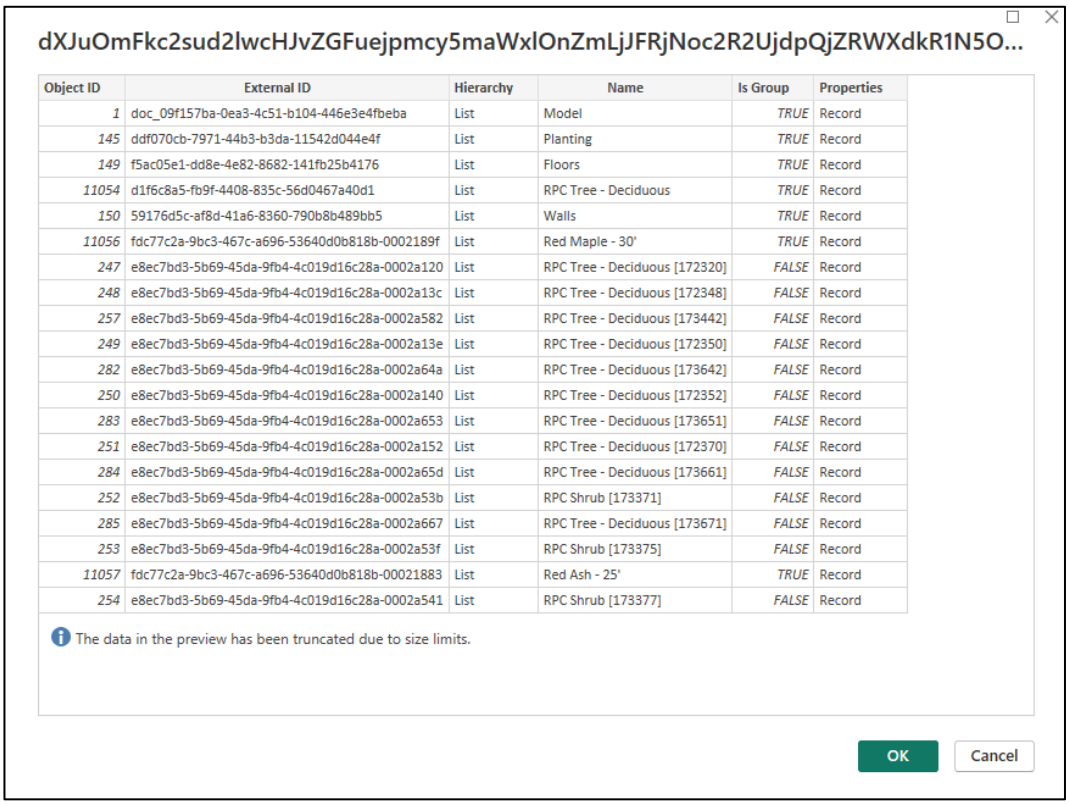


Click **Allow** and then **Connect**



When finishing, all model properties will be available as a table for further process.

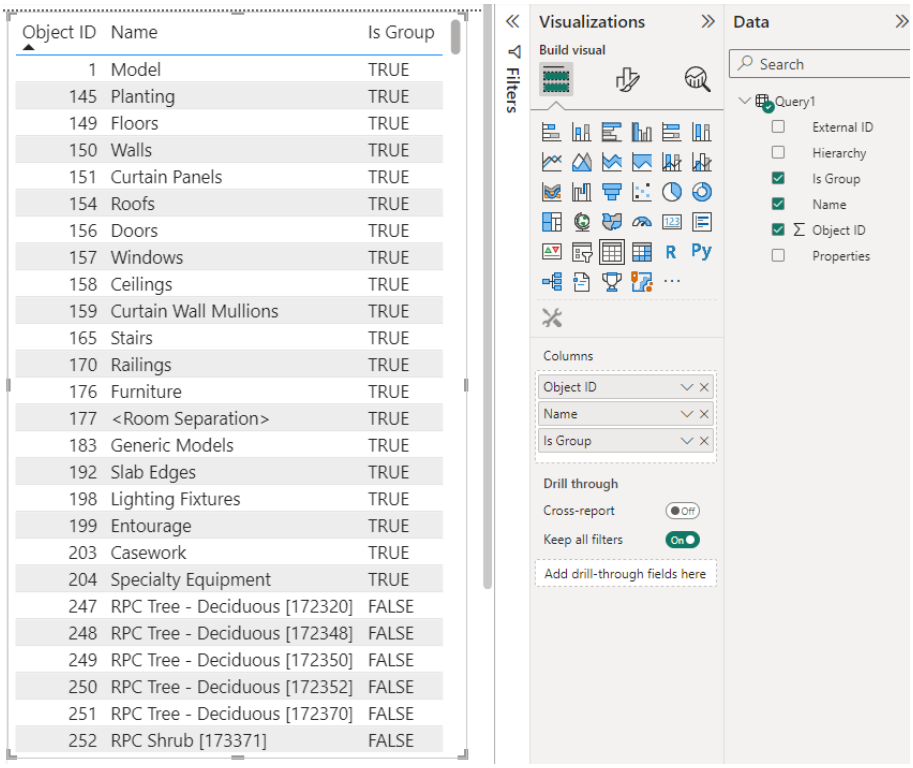
Lei He, KBR



For more information of how to develop, test and build the custom Power Query data connector, please refer to Develop a connector using the Power Query SDK - Power Query | Microsoft Learn.

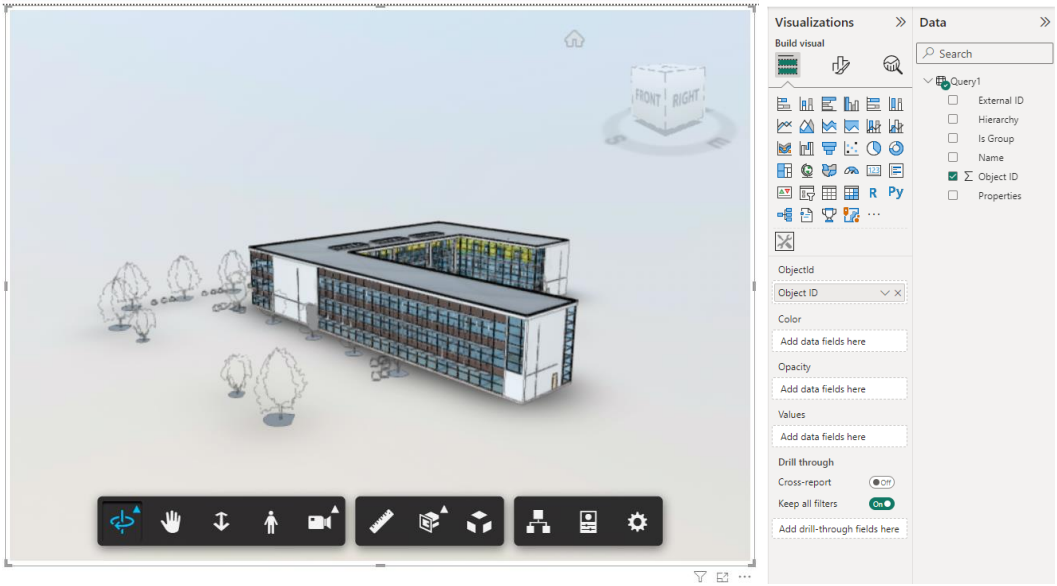
8. After the model properties data fully loaded into Power BI report, create a table view in the report and drag three columns of [Object ID], [Name], [Is Group] from **Data** panel to **Visualisations** Panel. We will be able to see a table with all the rows from the Query1 (from previous steps).

Lei He, KBR



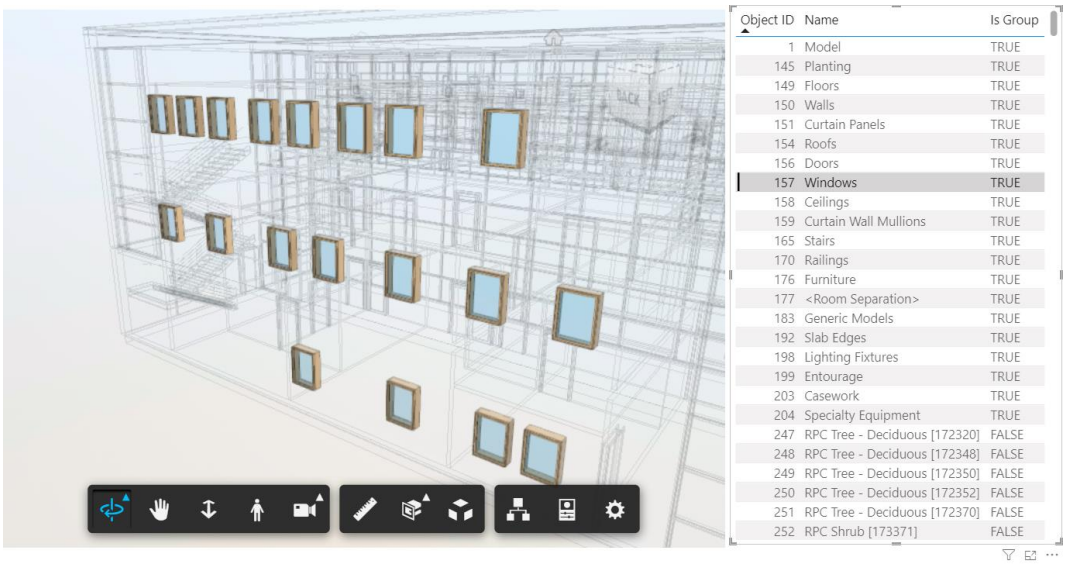
Object ID	Name	Is Group
1	Model	TRUE
145	Planting	TRUE
149	Floors	TRUE
150	Walls	TRUE
151	Curtain Panels	TRUE
154	Roofs	TRUE
156	Doors	TRUE
157	Windows	TRUE
158	Ceilings	TRUE
159	Curtain Wall Mullions	TRUE
165	Stairs	TRUE
170	Railings	TRUE
176	Furniture	TRUE
177	<Room Separation>	TRUE
183	Generic Models	TRUE
192	Slab Edges	TRUE
198	Lighting Fixtures	TRUE
199	Entourage	TRUE
203	Casework	TRUE
204	Specialty Equipment	TRUE
247	RPC Tree - Deciduous [172320]	FALSE
248	RPC Tree - Deciduous [172348]	FALSE
249	RPC Tree - Deciduous [172350]	FALSE
250	RPC Tree - Deciduous [172352]	FALSE
251	RPC Tree - Deciduous [172370]	FALSE
252	RPC Shrub [173371]	FALSE

Select the Power BI APS Model viewer and drag the [Object ID] to **ObjectId.** [Object ID] is the essential data column to establish the interaction between the model viewer and other Visuals available from the same report.



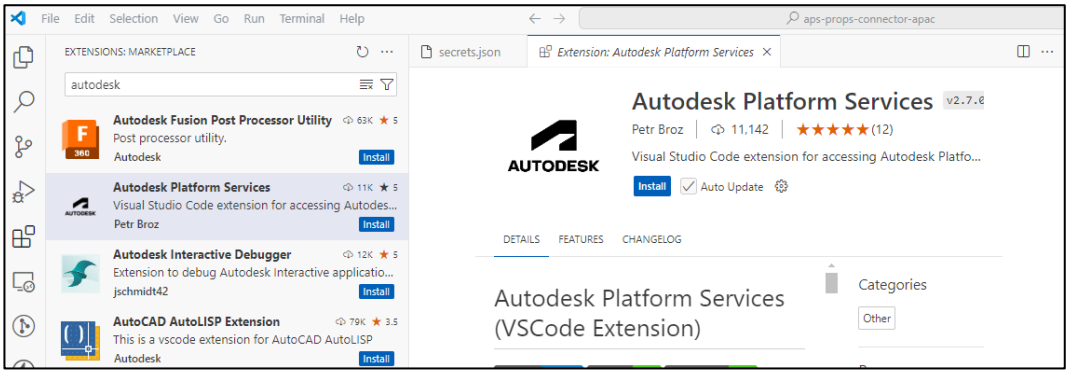
Select any row from the table will now zoom and highlight to its correspondent model element (multiple elements if the Object ID is from a group).

Lei He, KBR



VSCode APS extension

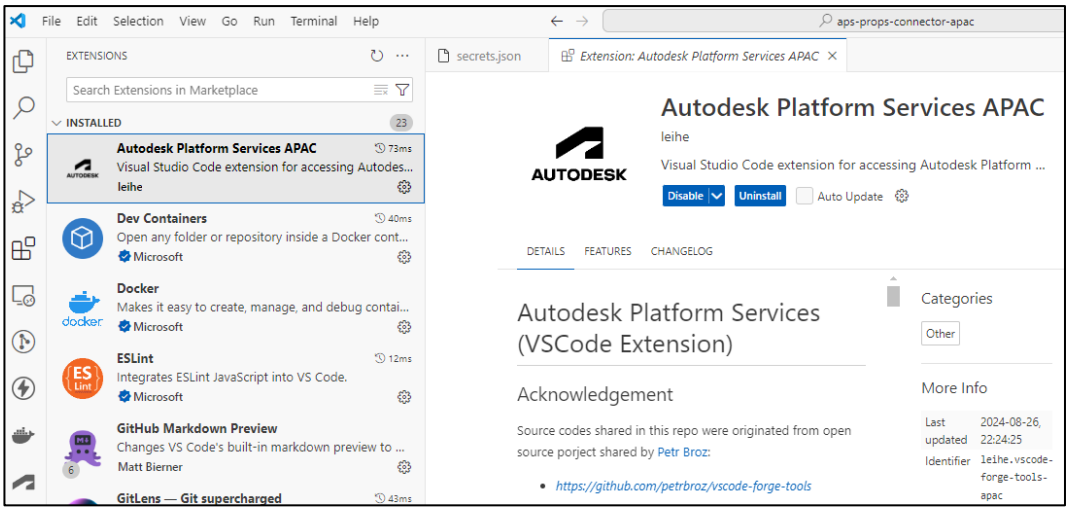
Petr Broz developed a VSCode extension for APS services. The tool provides better user experience and can be used to download model properties data or extract the model URN from ACC (and much more features). This extension is publicly available from VSCode marketplace.



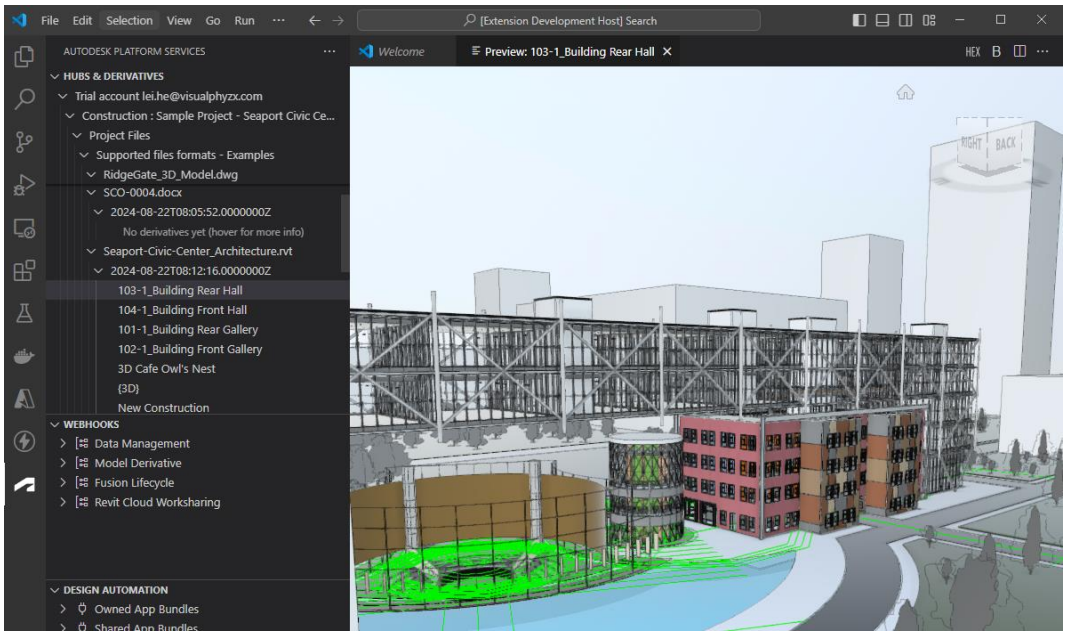
The current version of this tool cannot access ACC APAC region unfortunately. To be able to use it for ACC APAC region, I have updated it as **vscode-forge-tools-apac.** Users are welcome to update and compile it for their own use by following the provided documentation or use the .vsix file saved on the root folder - aps-powerbi-tools-apac/vscode-forge-tools-apac/vscode-forge-tools-apac-2.7.1.vsix at main · visualphyzx/aps-powerbi-tools-apac · GitHub.

Lei He, KBR

Please follow this link to manually install the extension - [How to install VS code extension manually? - Stack Overflow](#).
After manually installing:

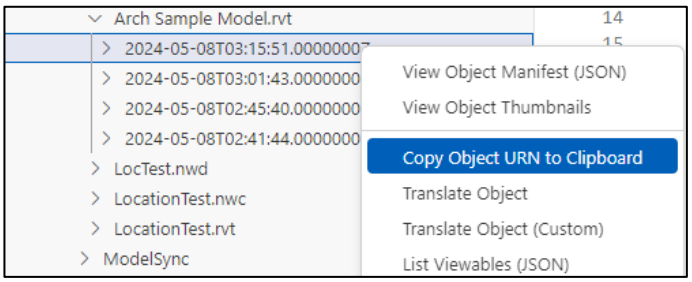


Users can iterate all projects on ACC and can preview the model directly inside VSCode.



Users can copy **URN** of the model at the specified version.

Lei He, KBR



Users can download model properties in json format directly inside VSCode (as we did in Power BI data connector).