

# Connecting SAP Datasphere with PowerBI Using OData Service

*A Guided Documentation by Adalbert Moczygeba, Sourabh Chourasiya, Srinivas Medidi, Camelot Management Consultants*

The business intelligence tool Power BI offers benefits and capabilities that enable organizations to leverage organizational data effectively, make informed decisions, and drive business growth.

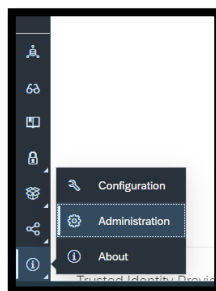
The combination of SAP Datasphere and Power BI enhances the capabilities of both platforms, enabling users to leverage SAP Datasphere's data sources within Power BI for comprehensive data analysis, visualization, and reporting. This integration promotes data-driven decision-making, improved collaboration, and greater agility in responding to business insights.

This documentation provides a general reference on how to integrate SAP Datasphere and Power BI using Data Service, to use SAP Datasphere entities as Data Sources for Power BI Queries. It will cover the following steps:

- Creation of OAuth Client
- Generating Code Grant Token
- Generating Refresh Token using Open API Postman
- Creating Power BI Blank Query Data Source
- Consuming SAP Datasphere Entities in Power BI

## Creation of OAuth Client

- Log into SAP Datasphere.
- Navigate to the 'Administration' page (found in the information section).



*Figure 1: Navigation to the administration page*

- Select the 'App Integration' tab.

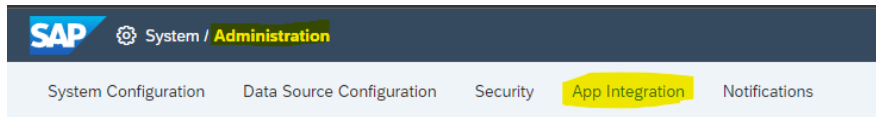


Figure 2: Finding the App Integration tab

- Scroll to the 'OAuth Clients' section. Click on 'Add a New OAuth Client'.

[+ Add a New OAuth Client](#)

Figure 3: Add connection

- Provide the name '**Power BI OAuth Client**' exactly as given.
  - In the "Purpose" field, select 'Interactive Usage'.
  - In "Redirect URL", provide the URL as follows:

<https://oauth.powerbi.com/views/oauthredirect.html>

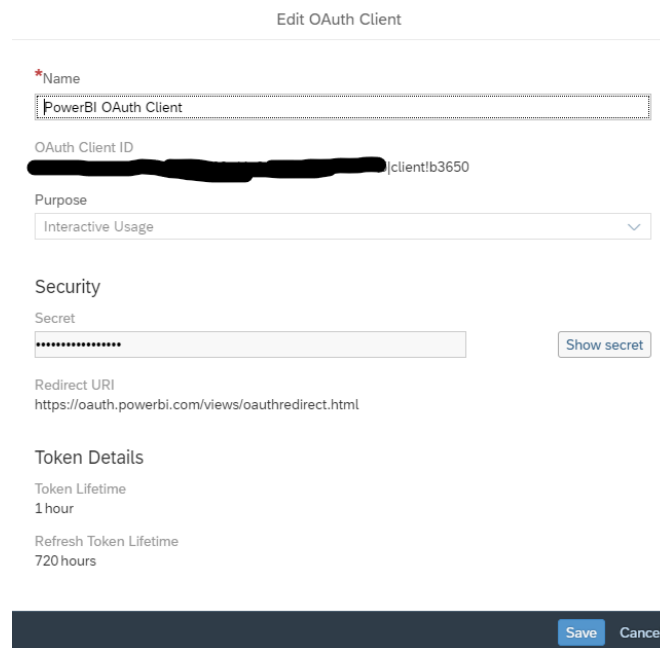


Figure 4: Power BI OAuth Client

- Click on the 'Save' Button.
  - OAuth Client is now created.
  - Click on "Edit Icon" to get your client ID and secret.
- While still in the OAuth Client section, make a note of the 'Authorization URL' and the 'OAuth Client ID' we generated before.

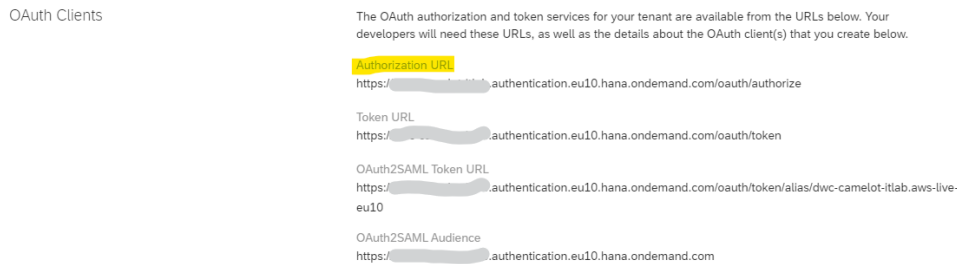


Figure 5: Power BI OAuth Client URL

## Generating Code Grant Token

- Generate the URL as shown below using the 'Authorization URL' and 'OAuth Client ID' we got from the step before.  
[https://\\*\\*\\*\\*\\*.authentication.eu10.hana.ondemand.com/oauth/authorize](https://*****.authentication.eu10.hana.ondemand.com/oauth/authorize)
- Open the generated URL in a new browser window. This initiates the authentication process to follow. After authentication, the page redirects to the Power BI authentication page.

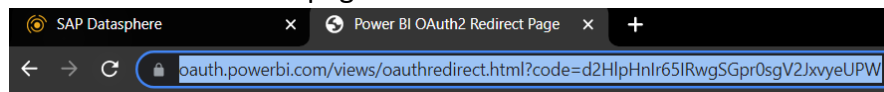


Figure 6: CODE\_URL

- If you are getting a "bad request" status like below, replace pipe symbol (|) in the client you ID with '%7C'.

**HTTP Status 400 – Bad Request**

Figure 7: Bad Request

- Copy the code (highlighted in the example below in yellow) from the Power BI authentication URL:

**Example:**

<https://oauth.powerbi.com/views/oauthredirect.html?code=d2HlpHnIr65IRwgSGpr0sgV2JxvyeUPW>

## Generating Refresh Token using Open API Postman

- To generate the "Refresh Token", we need an API.

In our example, we use the 'Postman' Application (free download available from <https://www.postman.com/downloads/>).

- Open the 'Postman' App.

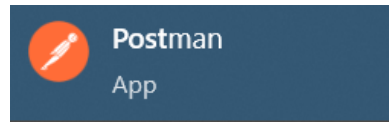


Figure 8: Postman App

- Navigate to 'Collections' and click on '+' symbol to create a new collection.

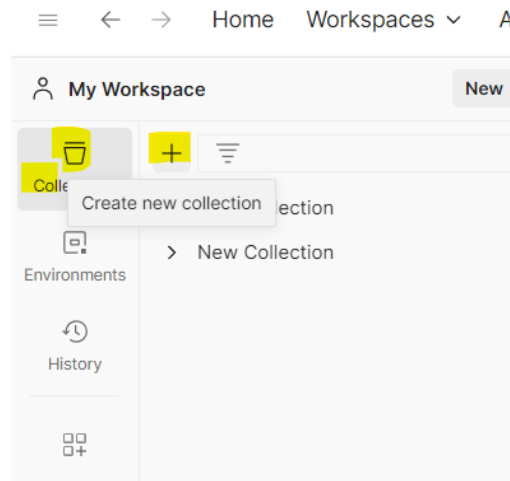


Figure 9: Create a new collection in Postman

- Expand the section 'New Collection', click on 'Add a request'.

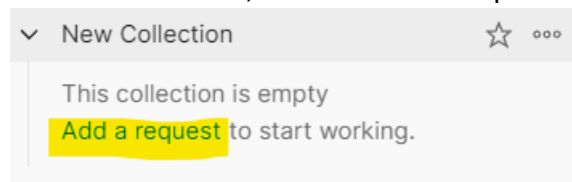


Figure 10: Add request

- You will see a screen like below. Update the request details to generate a refresh token.

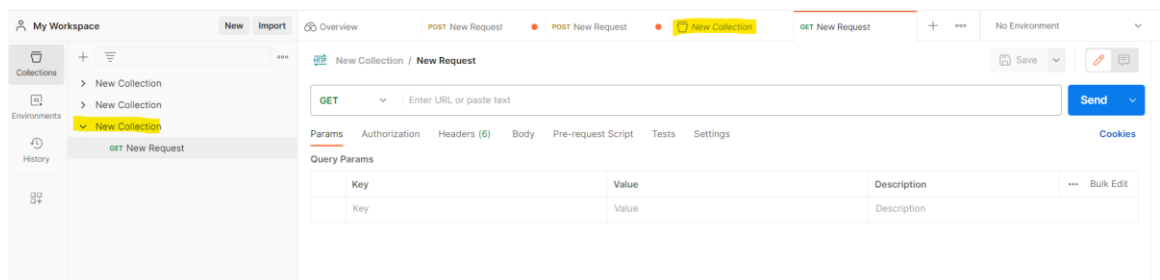


Figure 11: Updating request, step 1

- From the SAP Datasphere page (ideally still open in your first browser tab), copy the token URL.

The OAuth authorization and token services for your tenant are available from the URLs below. Your developers will need these URLs, as well as the details about the OAuth client(s) that you create below.

Authorization URL

https://[redacted]authentication.eu10.hana.ondemand.com/oauth/authorize

Token URL

https://[redacted]authentication.eu10.hana.ondemand.com/oauth/token

OAuth2SAML Token URL

https://[redacted]authentication.eu10.hana.ondemand.com/oauth/token/alias/dwc-camelot-itlab.aws-live-eu10

OAuth2SAML Audience

https://[redacted]authentication.eu10.hana.ondemand.com

Figure 12: Find the token URL in SAP Datasphere

- In Postman, change the request type to 'POST' and provide the token URL in the 'URL' section.

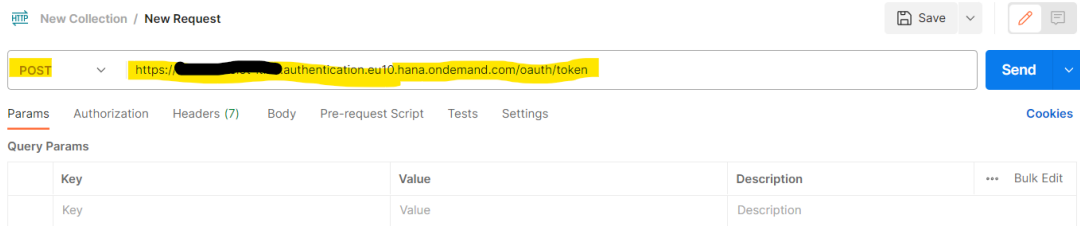


Figure 3: Updating request, step 2

- In the 'Authorization' tab, select 'Basic Auth' in the drop-down menu.

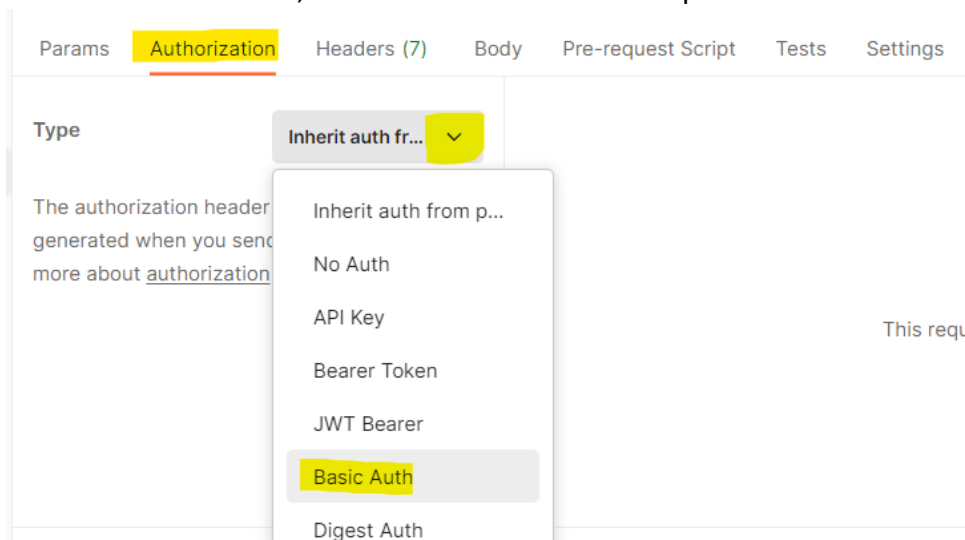


Figure 14: Updating request, step 3

- In the username and password boxes, please provide the information copied from the OAuth Client.

- Username – Provide ‘OAuth Client ID’.
- Password – Provide the ‘Secret key’.

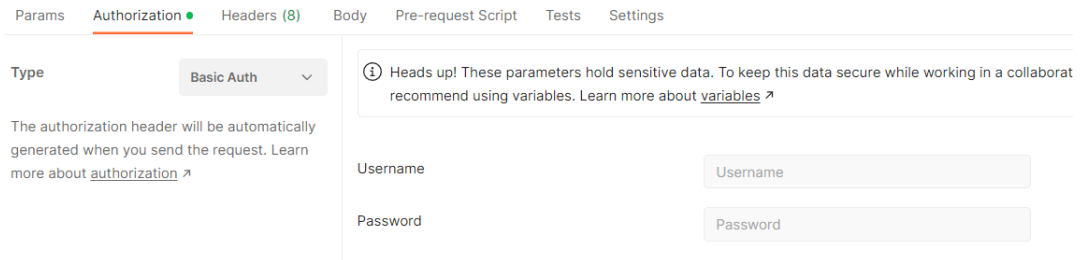


Figure 15: Updating request, step 4

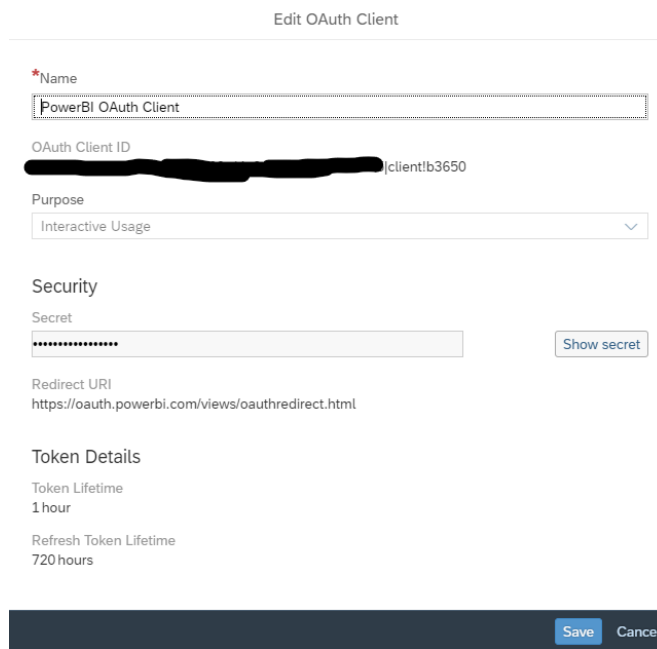


Figure 16: OAuth details

- In the ‘Headers’ section, add the two entries like below.

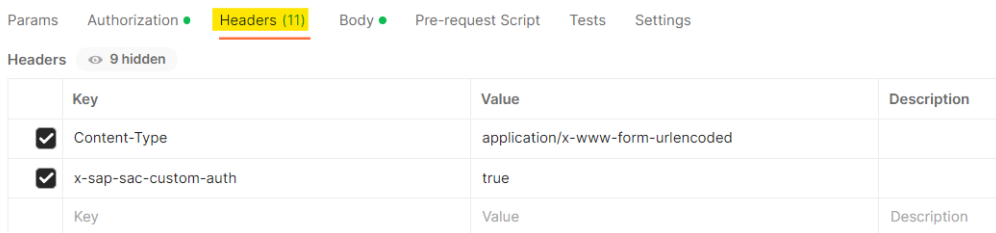


Figure 17: Updating request, step 5

- In the ‘Body’ section, add the three entries like below.  
The ‘code’ shown in yellow was generated previously during the step ‘Generating Code Grant Token’ section.

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

none
  form-data
  x-www-form-urlencoded
  raw
  binary
  GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> code	0x5UQijSqyIS6U7KPHxcBpLv9R7r1y4F	
<input checked="" type="checkbox"/> grant_type	authorization_code	
<input checked="" type="checkbox"/> response_type	token	
Key	Value	Description

Figure 18: Updating request, step 6

- Finally click on the 'Send' button. This will generate the 'Refresh Token' for us. Please save a copy of the 'refresh token', as we need it at a later stage.

```

1  {
2    "access_token": "I6ImhdHBz018vZhdJLWnhbWsb3QtaxRsYwIuYXV9aGVudG1jYXRpb24uZXUxMC5oYW5hLm9uZGVtYW5kLmNvbnB9b2t1b19",
3    "token_type": "bearer",
4    "id_token": "I6ImhdHBz018vZhdJLWnhbWsb3QtaxRsYwIuYXV9aGVudG1jYXRpb24uZXUxMC5oYW5hLm9uZGVtYW5kLmNvbnB9b2t1b19rZlI",
5    "refresh_token": "d8d7728b18c941688c1c5d83c0c625-x",
6    "expires_in": 3599,
7    "scope": "openid approuter-sac-saceu10!t3650.sap.fpa.user uaa.user",
8    "jti": "9e93c95a1d"
9  }
    
```

Figure 19: Request Output

### Creating Power BI Blank Query Data Source

- Open Power BI and create a new blank query.

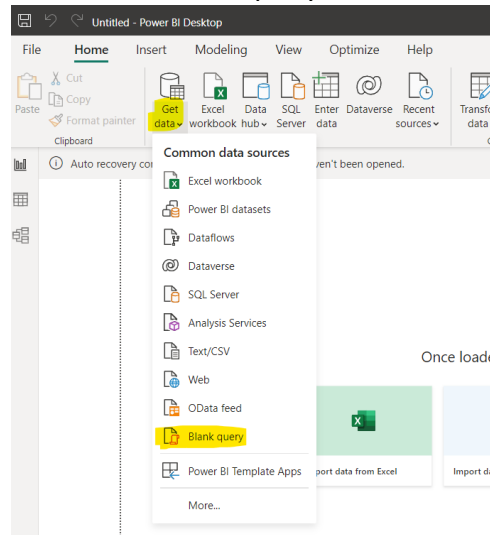


Figure 20: Power BI, step 1

- For setting up the data connection, the power script language will be used. To initiate this, open the "Advanced Editor".

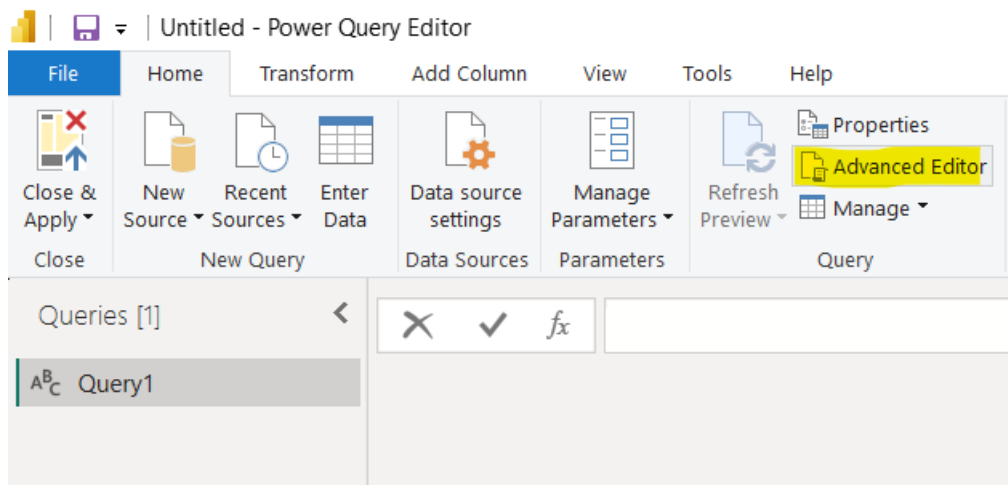


Figure 21: Power BI, step 2

In 'Advanced Editor', update the entries '**Authorization Token URL**', '**Refresh Token**', '**Client-ID**', and '**Secret**' with your details.

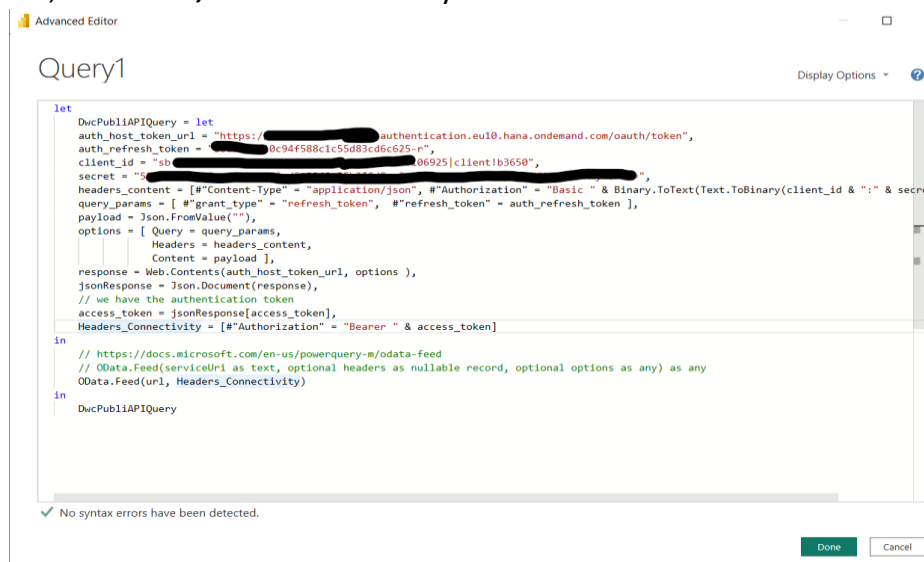


Figure 22: Power BI, step 3

### Power Script example

```
let
    DwcPubliAPIQuery = let
        auth_host_token_url =
            "https://****.authentication.eu10.hana.ondemand.com/oauth/token",
        auth_refresh_token = "*****c625-r",
        client_id = "sb-*****925|client!b3650",
        secret = "*****yNsF8o=",
        headers_content = [#"Content-Type" = "application/json", #"Authorization" = "Basic " &
            Binary.ToText(Text.ToBinary(client_id & ":" & secret), 0)],
        query_params = [#"grant_type" = "refresh_token", #"refresh_token" = auth_refresh_token
    ],
```



```

payload = Json.FromValue(""),
options = [ Query = query_params,
            Headers = headers_content,
            Content = payload ],
response = Web.Contents(auth_host_token_url, options ),
jsonResponse = Json.Document(response),
// we have the authentication token
access_token = jsonResponse[access_token],
Headers_Connectivity = [#"Authorization" = "Bearer " & access_token]
in
// https://docs.microsoft.com/en-us/powerquery-m/odata-feed
// OData.Feed(serviceUri as text, optional headers as nullable record, optional options as
any) as any
OData.Feed(url, Headers_Connectivity)
in
DwcPubliAPIQuery

```

- Click “Done” to close the “Advanced Editor”.
- Create ‘New Parameter’ from ‘Manage Parameters’.

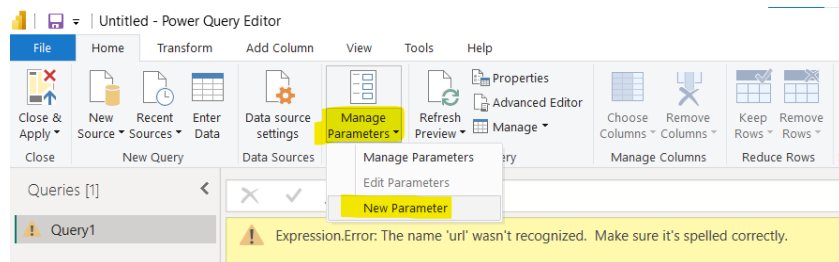


Figure 23: Power BI, step 4

Provide these details:

Name: url

Required Check Box – Ticked

Type – Text

Suggested Values – Any Value

Current Value - [https://\\*\\*\\*\\*.cloud.sap/api/v1/dwc/catalog/](https://****.cloud.sap/api/v1/dwc/catalog/)

Replace \*\*\*\* with <tenant\_url>

Or

[https://<tenant\\_url>.cloud.sap/dwaas-core/odata/v4/catalog/](https://<tenant_url>.cloud.sap/dwaas-core/odata/v4/catalog/)

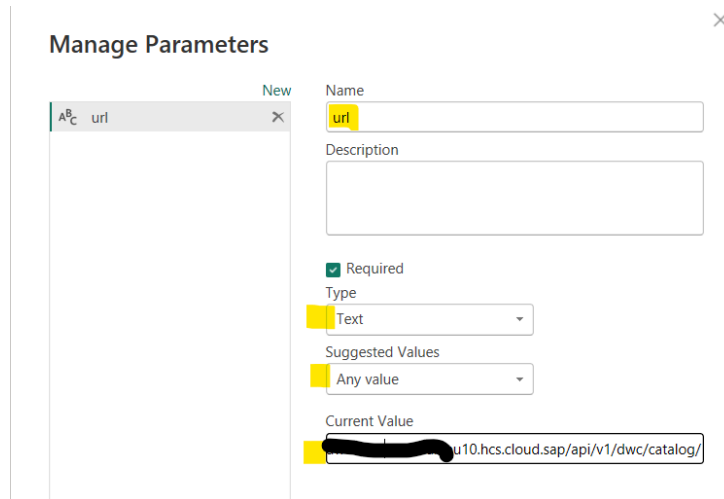


Figure 24: Power BI, step 5

- Click on 'OK'.
- If the query does not run successfully at this point, you may see a screen like below.

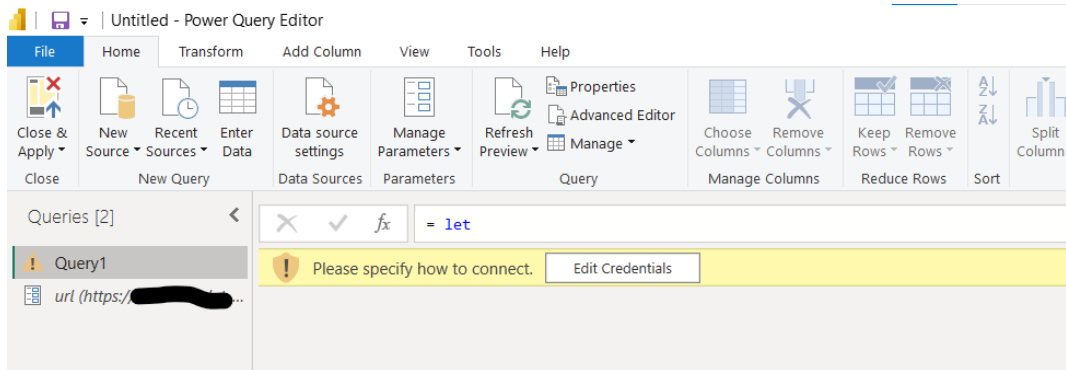


Figure 25: Power BI, step 6

- To fix this, click on 'Edit Credentials'. This takes you through some checks, kindly accept those.

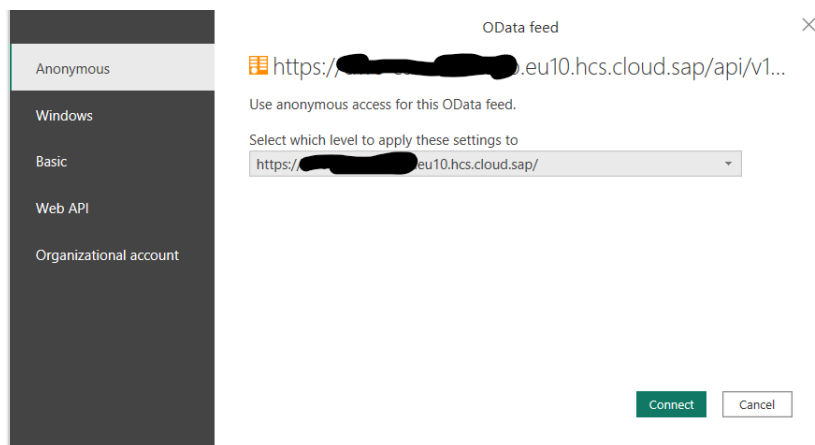


Figure 26: Power BI, step 7

The last step informs on privacy levels. Please read carefully and accept.

### Privacy levels

The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found [here](#).

- Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person.

- After successful execution, you should see a screen as below:

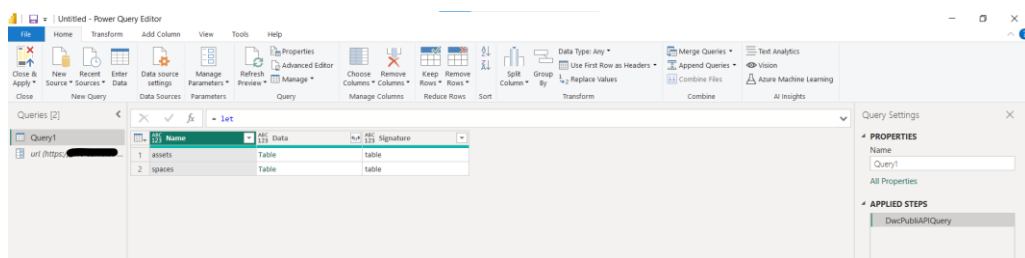


Figure 27: Power BI, step 8

## Consuming SAP Datasphere Entities in Power BI

- By finishing the steps described so far, you established a working connection between SAP Datasphere with Power BI. You should see spaces and the consumable entities by clicking on the 'Table' link in the data column.

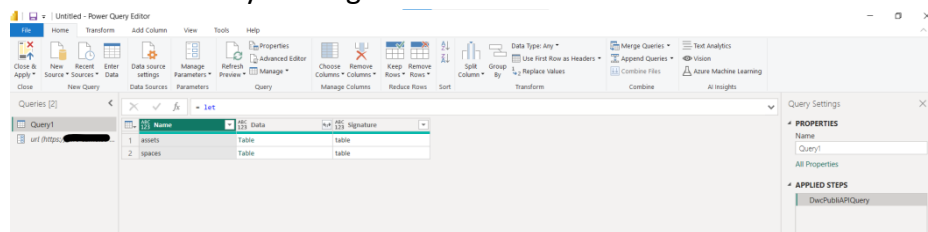


Figure 28: Power BI, step 9

- At this point in the process, it is difficult to see the data available in the entities. We need to create a function first and then add a new column to the table. With this, we are able to see the data in the Datasphere entities.
- To create a function, open the context menu with a right click on the query and select 'Create Function...'

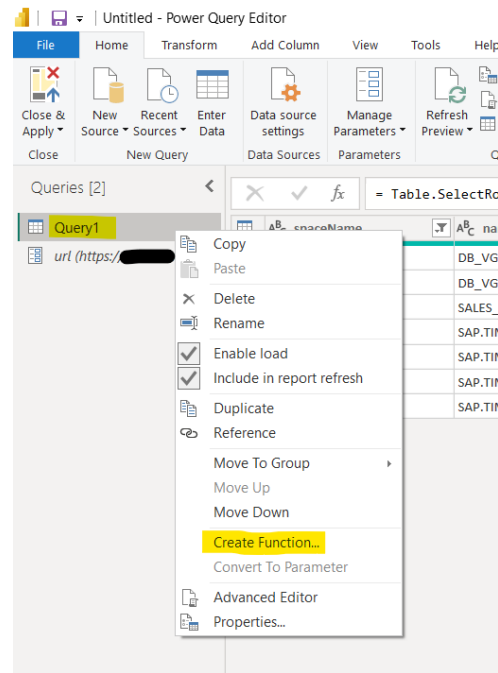


Figure 29: PowerBI, step 10

Provide function name as 'DWC OData API'.



Figure 30: PowerBI, step 11

- Provide the below URL after replacing the placeholder with your tenant url  
`https://<tenant_url>.cloud.sap/api/v1/dwc/catalog/`

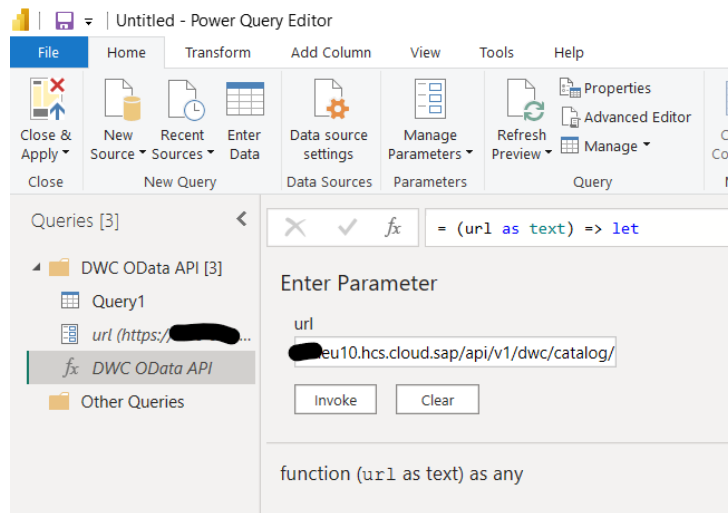


Figure 31: Power BI, step 12

- Click on 'Invoke'. Then you can see data retrieved already using the custom function.

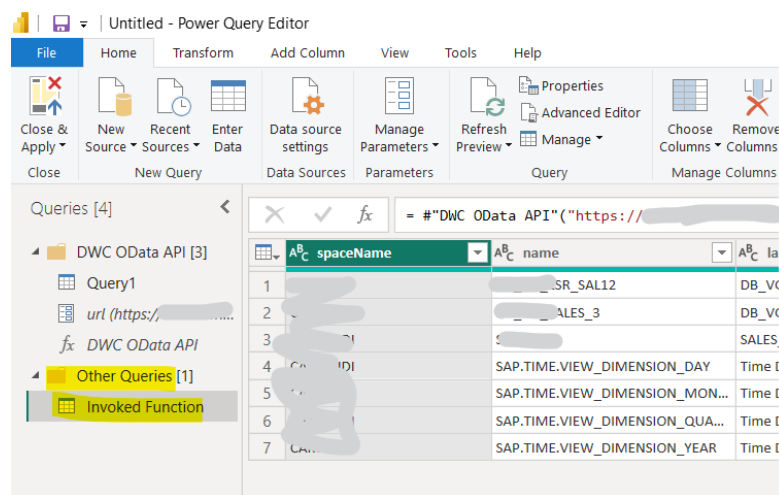


Figure 32: Power BI, step 13

- Next, we are going to add a new column to this asset table which provides a link to check the data in Datasphere model/entity.
- Go to the 'Add Column' tab and click on 'Invoke Custom Function'.

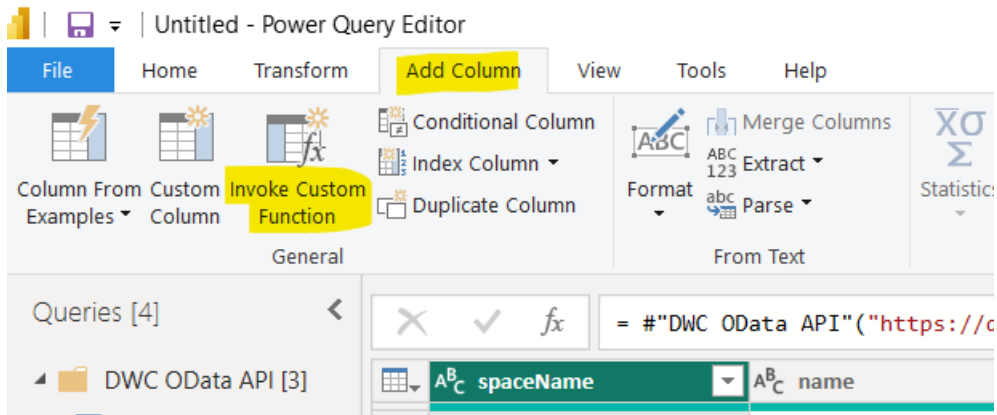


Figure 33: Power BI, step 14

- Provide 'Relational Data Navigation' as name. On 'Function query', select the custom function we defined earlier: 'DWC OData API'.

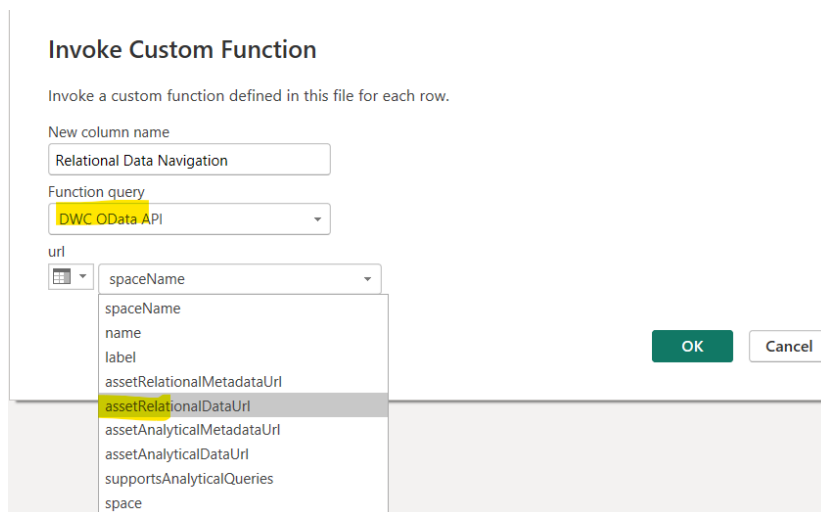


Figure 34: Power BI, step 15

- Conform with 'OK'.
- It will take some time until the services of all assets are checked. If everything goes well, you will see a new column at the end of the table like the one highlighted below.

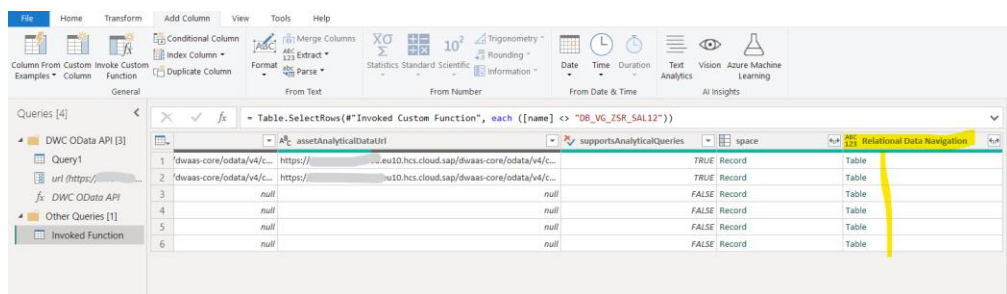


Figure 35: Power BI, step 16

- By clicking on the table link, we can see the data in the asset.

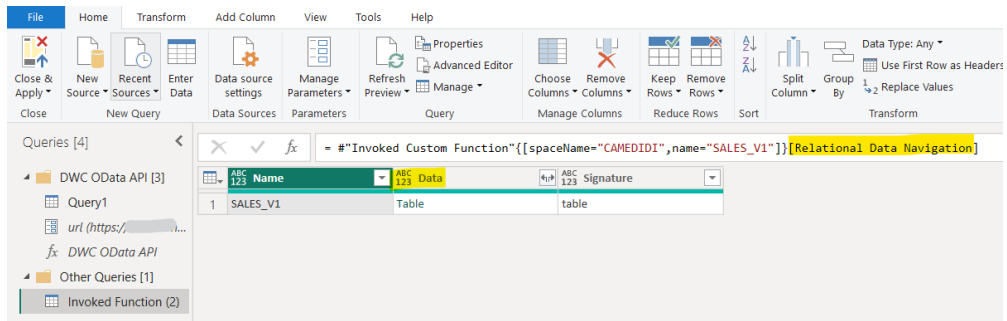


Figure 36: Power BI, step 17

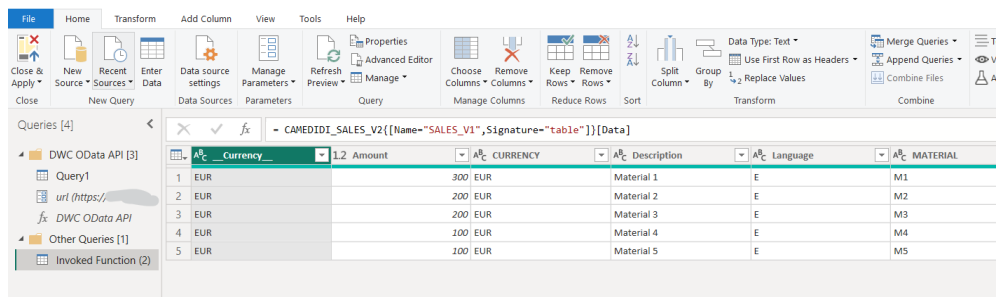


Figure 37: Power BI, step 18

## Further Documentation

By connecting SAP Datasphere with Power BI using the OData service, you can access and analyze the data in Power BI, taking advantage of Power BI's robust data visualization and analytics capabilities. As this runs with a consumer role, you can also give access to SAP Datasphere data to users without credentials.

Please note that the specific steps may vary based on the version and configuration of SAP Datasphere and Power BI, so it's recommended to refer to the official documentation and consult with the relevant product documentation or support teams for the most accurate and up-to-date instructions.

## Contact

**Adalbert Moczygeba** | Head of Profit Center Business Analytics | amoc@camelot-itlab.com  
Camelot Management Consultants, Global Headquarters, Theodor-Heuss-Anlage 12, 68165 Mannheim, Germany

[www.camelot-mc.com](http://www.camelot-mc.com)