**Complete/Selective Refresh Power BI Model from Azure Databricks**

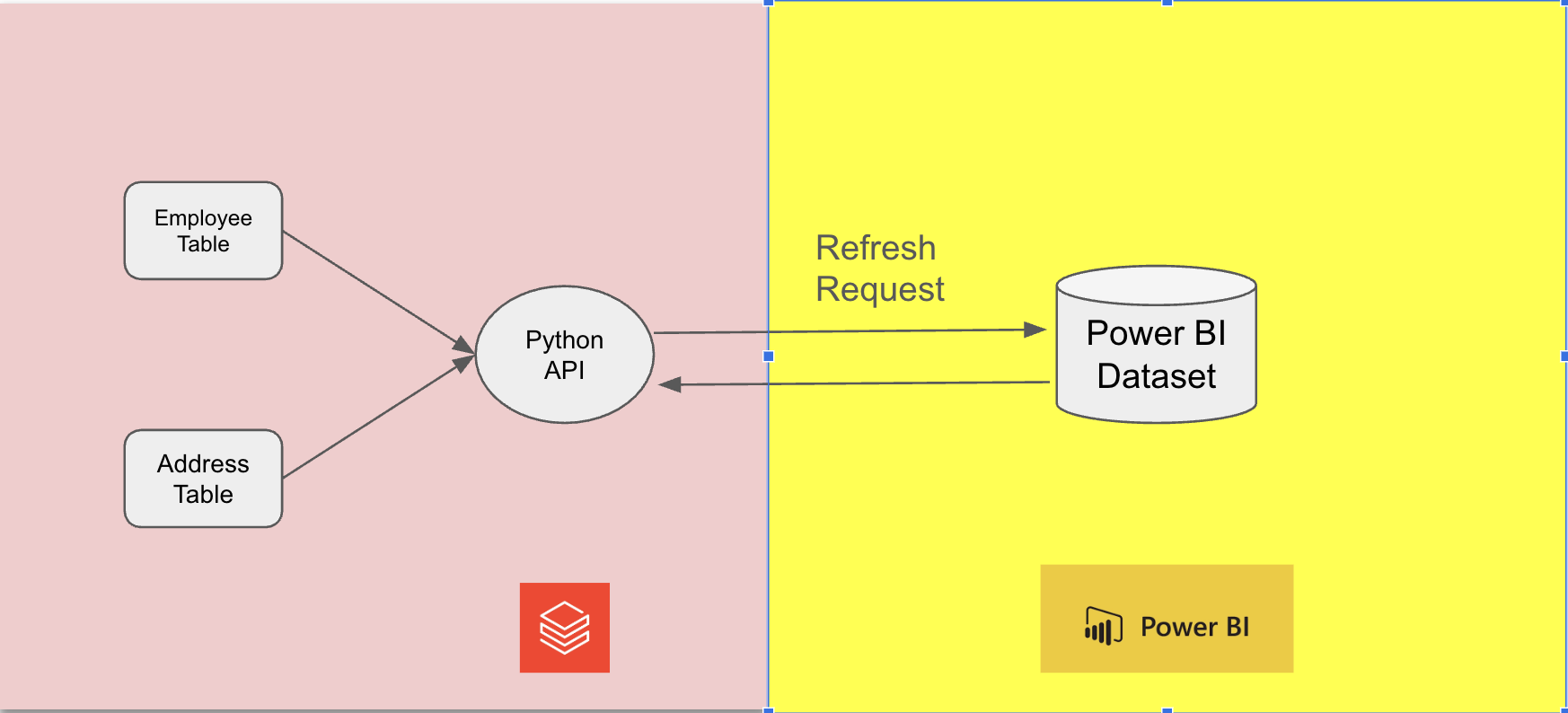
## **🔍 Introduction**

An important aspect of migration to Databricks is ensuring effective data refresh for external BI tools like Power BI or Tableau. This document outlines the method to perform a **complete or selective refresh** of Power BI Import Mode Models using **Power BI REST APIs** from **Azure Databricks**.

One can opt for refreshing the entire model or refresh selective tables in the model through Enhanced Refresh of Power BI with Rest API eliminating timeout issues.

## **🧰 Pre-requisites**

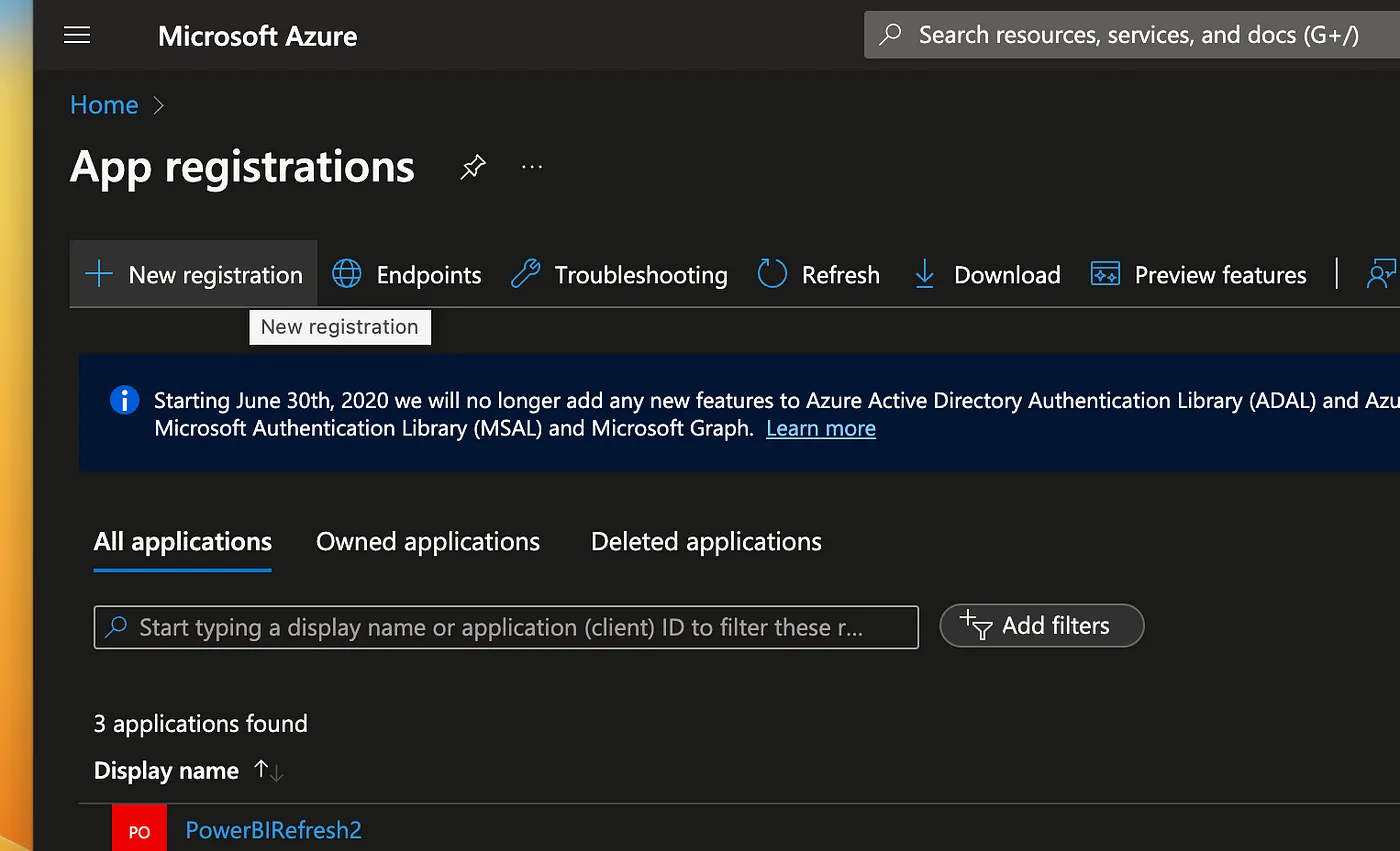
* Azure Admin and Power BI Admin privileges.
* Azure Service Principal (SPN) with necessary API permissions.
* Power BI Tenant configured to allow service principals.

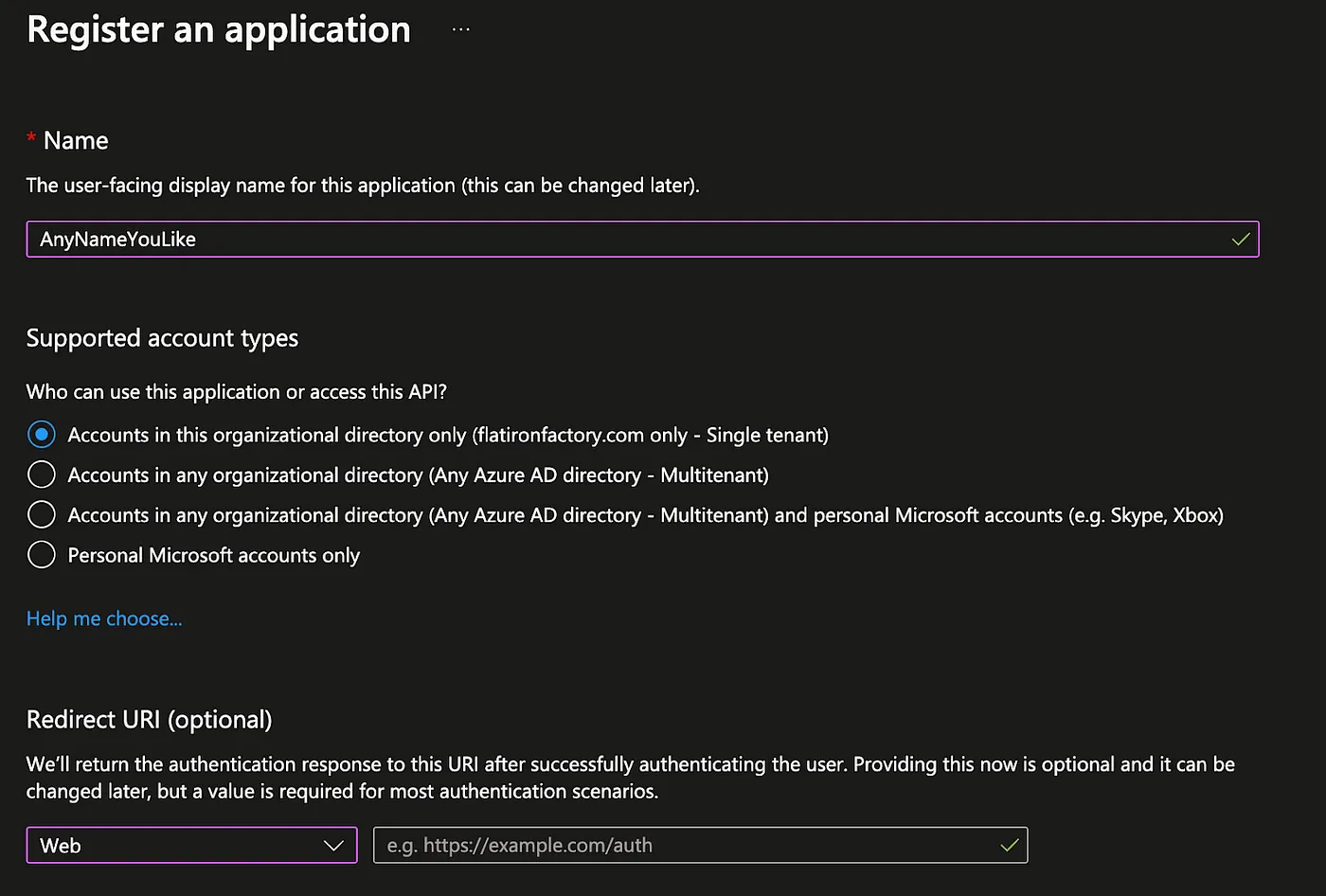


**🔐 Step 1: Service Principal Setup:**

Let’s set up the SPN and its necessary permissions needed to trigger a refresh in PowerBI Workspace.

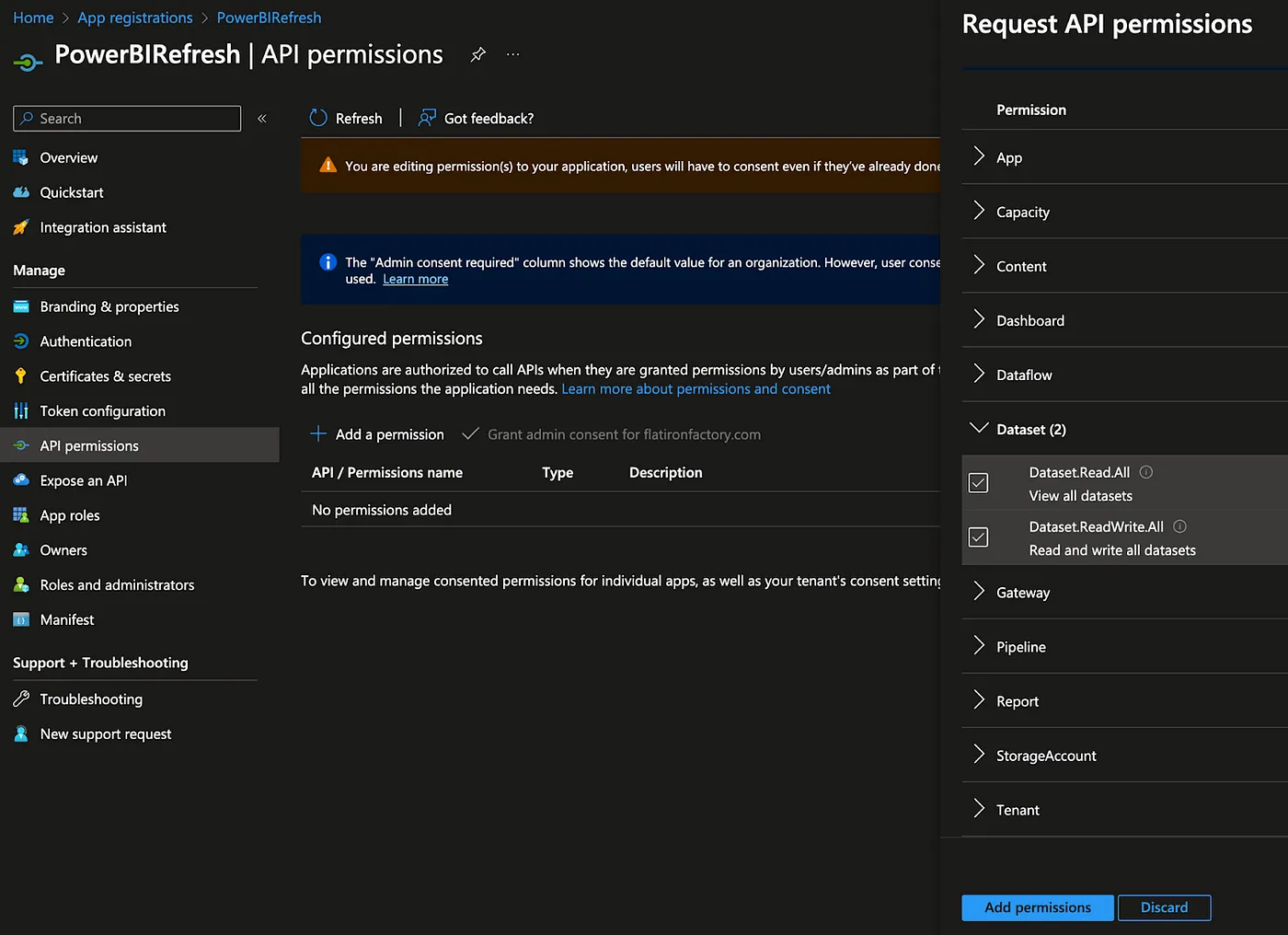
1. Create a SPN In Azure App Registrations.



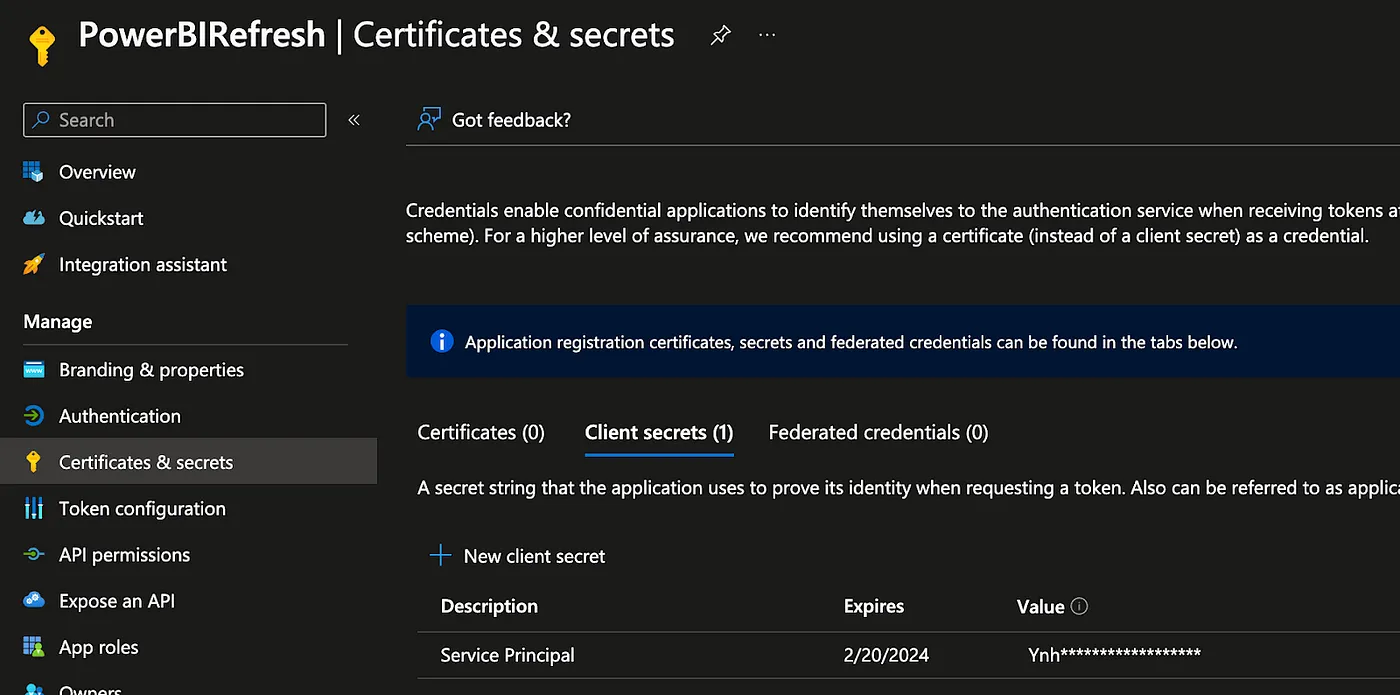


1. Go to API permissions of the App created above and add the below permissions

* DatasetReadWrite.All

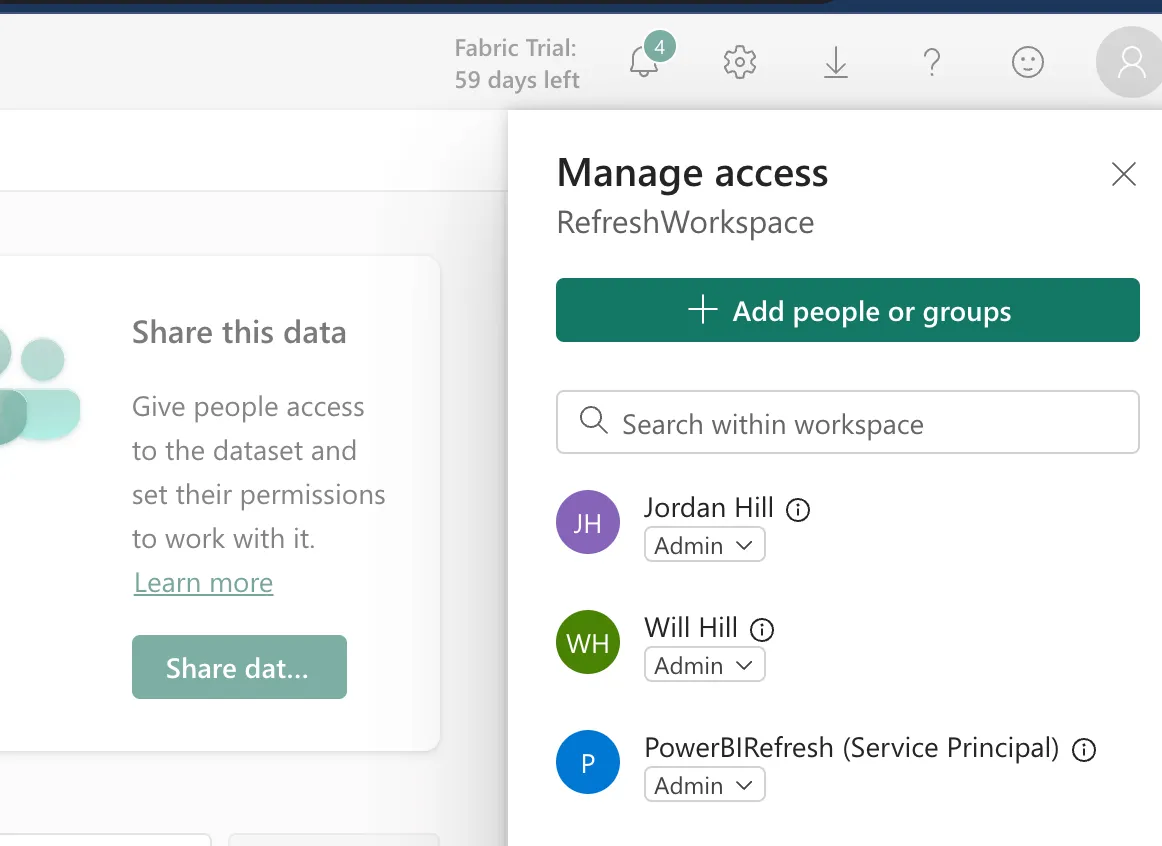


1. Then create a secret and a key. We need to use this secret and its key in our code to trigger the refresh.

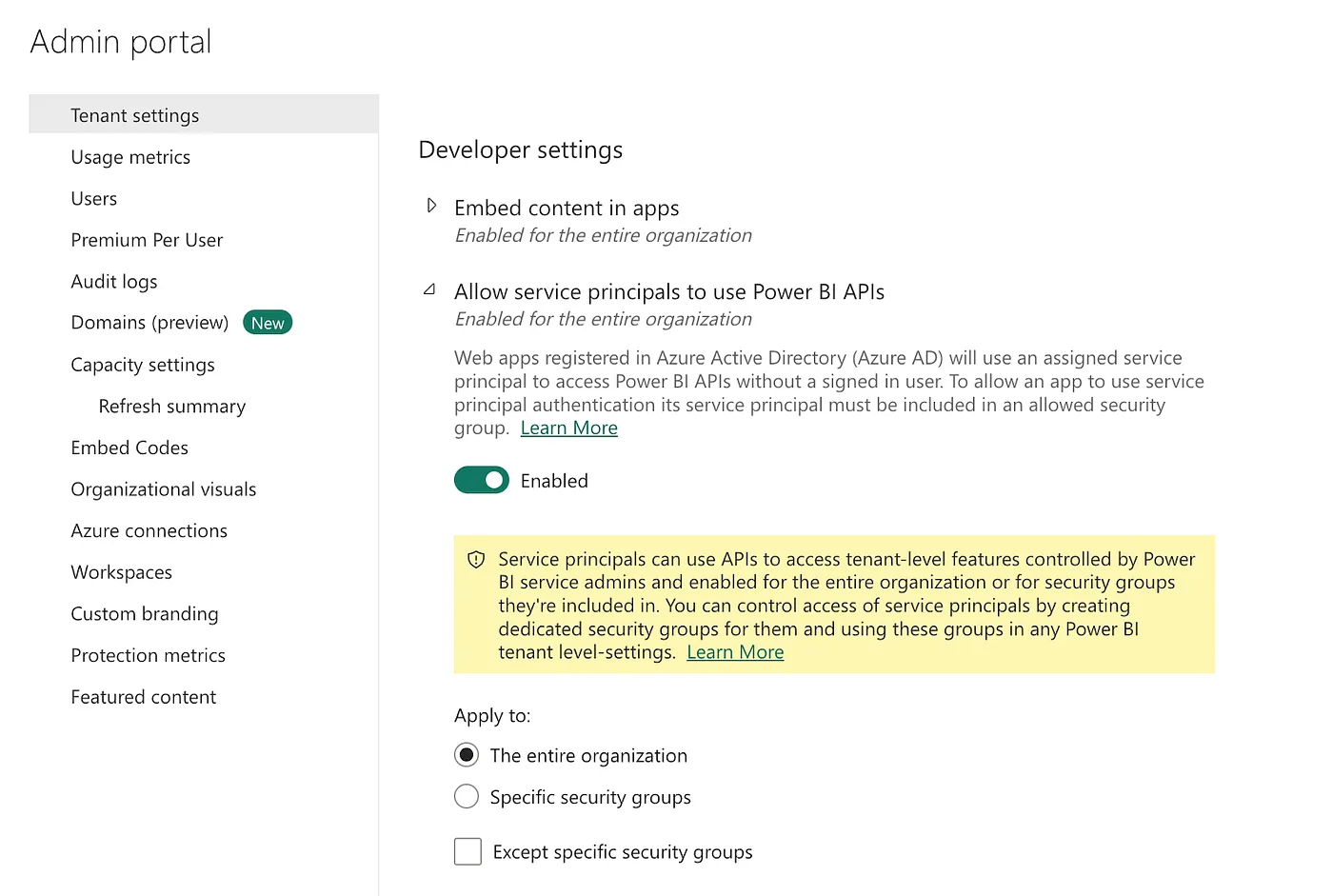


🏢 **Step 2: Power BI Tenant Configuration:**

1. Open Power BI Tenant and add the SPN as Admin in the Manage Access Tab.



1. Once Done, go to the Power BI Admin Portal and Enable “Allow Service Principals to use Power BI APIs” option. You can enable it for an entire organization or specific security groups based on your security needs.



**🧾 Step 3: Code for Refresh Operations:**

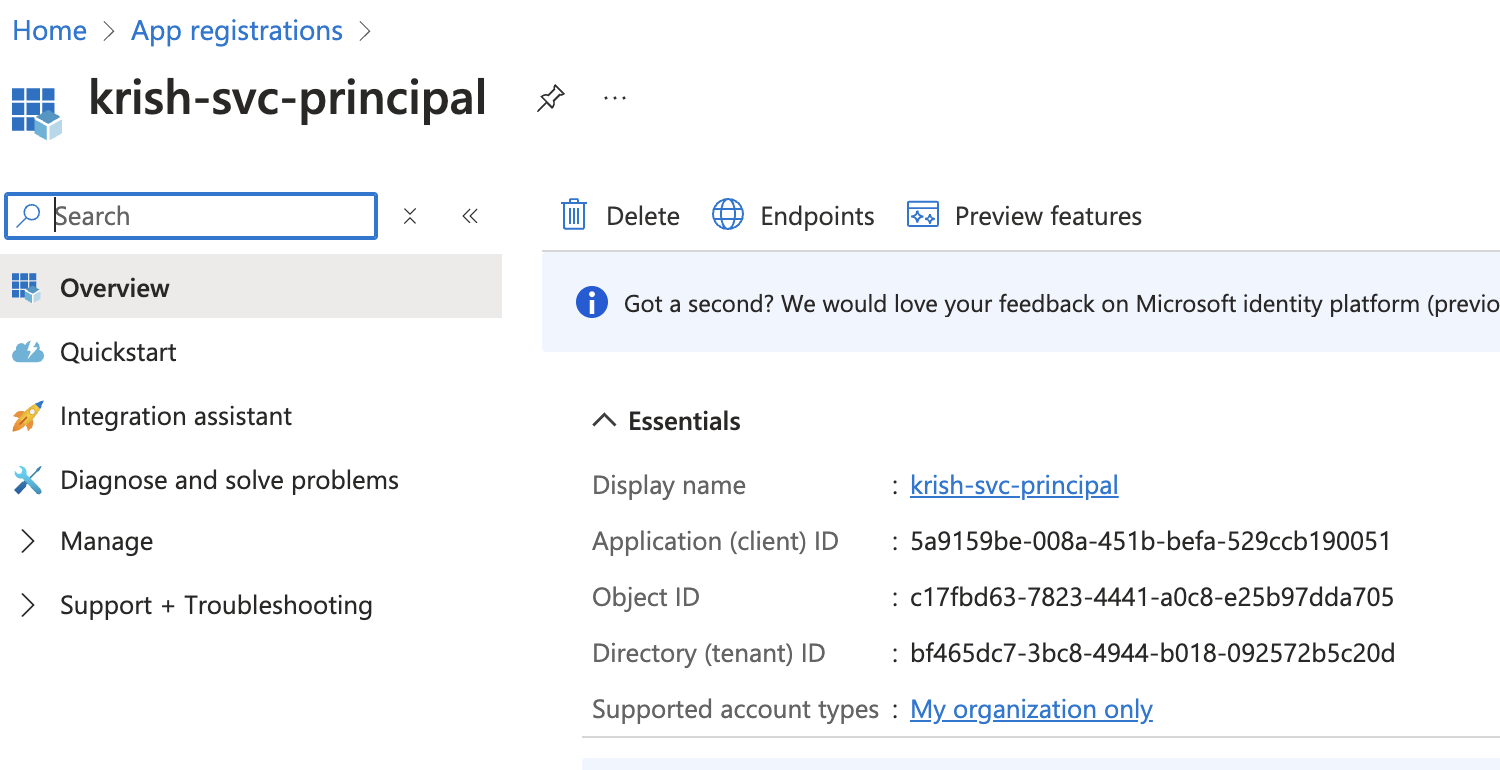
**Pre-requisites:**

Client Secret = Secret Key of the SPN Created Above

Client ID = App ID Of the SPN Created above

Tenant ID = Azure Tenant Name

Reference of how to get above details are mentioned below :

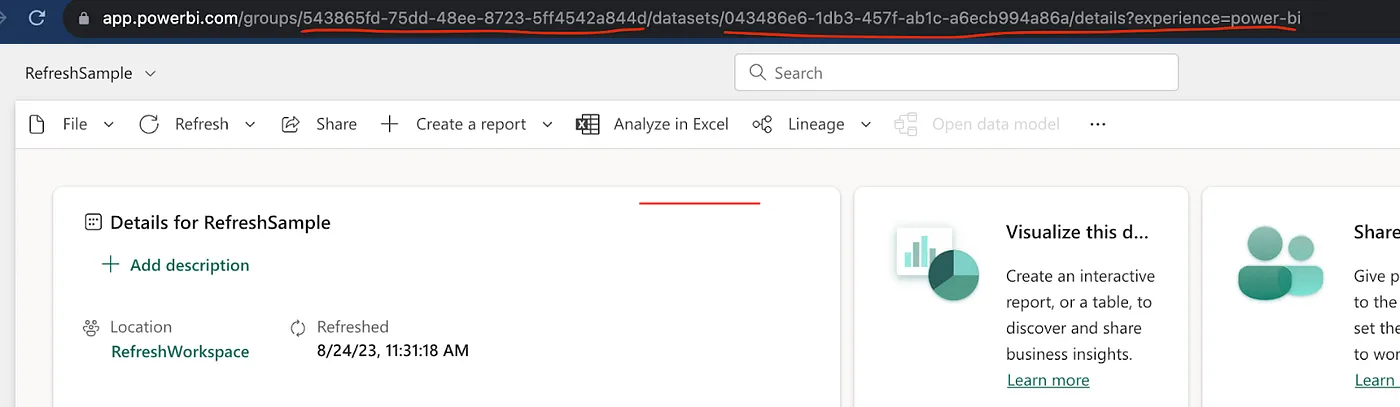


For maintaining standard security practices, store the client\_secret, client\_id and the tenant\_id in a databricks scope and use the scope keys in the refresh code.

Workspace Id : Group Id of the Power BI Workspace.

Dataset ID : Dataset ID of the dataset in Power BI Workspace.

We can get the workspace id and dataset id details from Power BI Model:



**Below Code does the following:**

1. Gets the authentication token from azure
2. Triggers the Refresh API (Selective/Complete)
3. Fetches the Refresh Completion Start Time, End Time, Total Time Elapsed

import requests

import json

import msal

import time

from datetime import datetime

session = requests.Session()

# === Utility to acquire token ===

def get\_access\_token(client\_id, client\_secret, tenant\_id):

scope = ['https://analysis.windows.net/powerbi/api/.default']

app = msal.ConfidentialClientApplication(

client\_id=client\_id,

client\_credential=client\_secret,

authority=f'https://login.microsoftonline.com/{tenant\_id}'

)

result = app.acquire\_token\_for\_client(scopes=scope)

if 'access\_token' not in result:

raise Exception("Failed to acquire access token")

return result['access\_token']

# === Method 1: Trigger Refresh ===

def trigger\_refresh(workspace\_id, dataset\_id, client\_id, client\_secret, tenant\_id):

access\_token = get\_access\_token(client\_id, client\_secret, tenant\_id)

refresh\_url = f"https://api.powerbi.com/v1.0/myorg/groups/{workspace\_id}/datasets/{dataset\_id}/refreshes"

headers = {

'Authorization': f'Bearer {access\_token}',

'Content-Type': 'application/json'

}

payload = {

"type": "Full",

"commitMode": "transactional",

"maxParallelism": 2,

"retryCount": 2,

"timeout": "02:00:00",

"objects": [

{"table": "DimCustomer", "partition": "DimCustomer"},

{"table": "DimDate"}

]

}

response = requests.post(refresh\_url, headers=headers, data=json.dumps(payload))

print(f"Trigger Refresh Response: {response.status\_code} - {response.text}")

if response.status\_code != 202:

raise Exception(f"Failed to trigger refresh: {response.status\_code} - {response.text}")

return True # Just acknowledge it triggered

# === Method 2: Check Refresh Status Until Done ===

def wait\_for\_refresh\_completion(workspace\_id, dataset\_id, client\_id, client\_secret, tenant\_id, poll\_interval=10):

access\_token = get\_access\_token(client\_id, client\_secret, tenant\_id)

status\_url = f"https://api.powerbi.com/v1.0/myorg/groups/{workspace\_id}/datasets/{dataset\_id}/refreshes?$top=1"

headers = {

'Authorization': f'Bearer {access\_token}',

'Content-Type': 'application/json'

}

start\_time = datetime.now()

print(f"Started polling at {start\_time.strftime('%Y-%m-%d %H:%M:%S')}")

while True:

response = requests.get(status\_url, headers=headers)

if response.status\_code != 200:

raise Exception(f"Failed to fetch refresh status: {response.status\_code} - {response.text}")

refresh\_data = response.json()

latest\_refresh = refresh\_data.get('value', [])[0]

status = latest\_refresh.get('status', 'Unknown')

print(f"Current Refresh Status: {status} at {datetime.now().strftime('%H:%M:%S')}")

if status.lower() not in ["unknown", "inprogress"]:

# If status is not 'InProgress' or 'Unknown', we're done

end\_time = datetime.now()

total\_time = (end\_time - start\_time).total\_seconds()

print(f"Finished at {end\_time.strftime('%Y-%m-%d %H:%M:%S')}")

print(f"Total Refresh Time: {total\_time:.2f} seconds")

print(f"Final Status: {status}")

return status, start\_time, end\_time, total\_time

# Sleep and check again

time.sleep(poll\_interval)

# === Example Usage ===

# == Credentials ==

#client\_secret = dbutils.secrets.get(scope="",key="")

client\_id = ""

client\_secret = ""

tenant\_id = ""

workspace\_id = ""

dataset\_id = ""

# == Step 1: Trigger the refresh ==

trigger\_refresh(workspace\_id, dataset\_id, client\_id, client\_secret, tenant\_id)

# == Step 2: Keep polling until it's complete ==

status, start\_time, end\_time, total\_time = wait\_for\_refresh\_completion(

workspace\_id, dataset\_id, client\_id, client\_secret, tenant\_id, poll\_interval=10

)

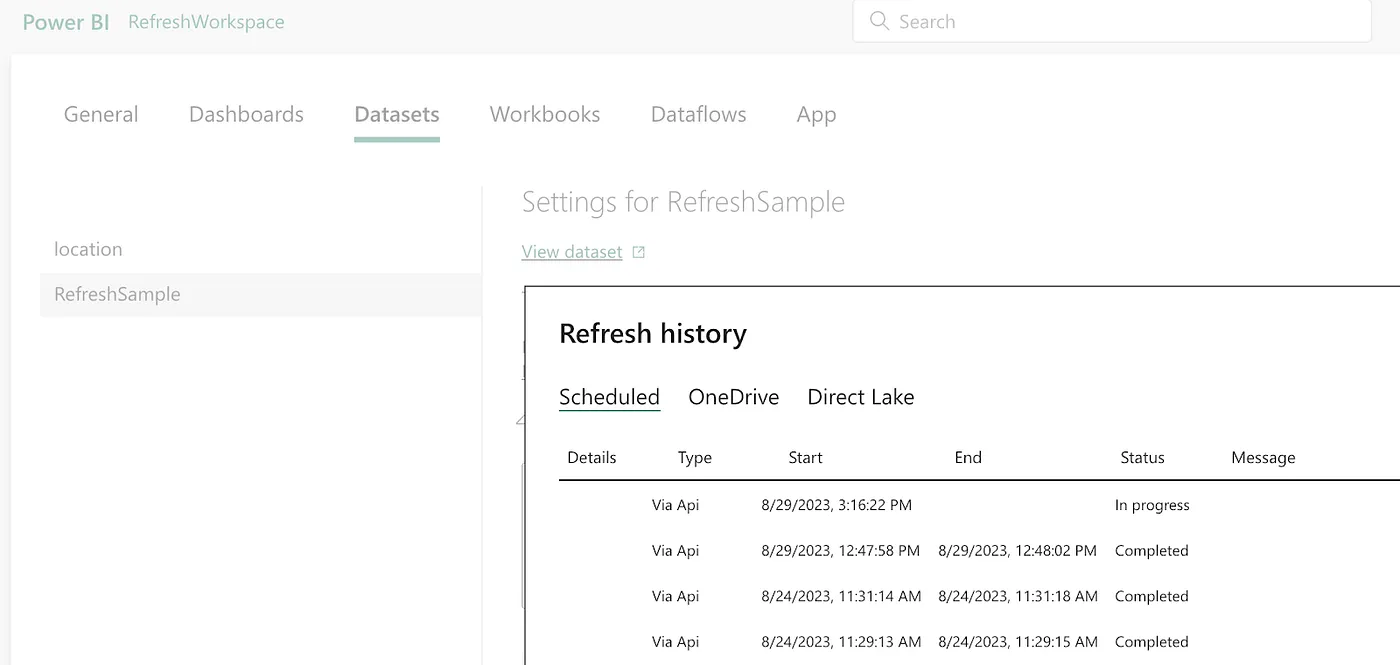
print(f"\nRefresh Completed with Status: {status}")

print(f"Started at: {start\_time}")

print(f"Ended at: {end\_time}")

print(f"Total Duration: {total\_time} seconds")

**Once triggered we can verify the refresh operation in Power BI:**



To perform a **selective refresh**, you can use the provided code with a payload that specifies the tables (or partitions) you want to refresh. This allows for more granular control, enabling refresh operations at the **table** or **partition level** instead of refreshing the entire model.

If you wish to **refresh the entire dataset**, simply omit the payload from the POST request—this will trigger a full model refresh by default

##response = requests.post(api\_url, headers=headers)