Get started with your own Power BI Monitor!

Paulien van Eijk

11-03-2023

De Power BI Gebruikersdagen worden mede mogelijk gemaakt door:

PBIG
POWER BI GEBRUIKERSGROEP

PLATINUM

delaware

Smart. wortell

ilionx

easydash

GOUD

Nova Silva

creates.

Tabular Editor

EIFFEL
Voorblijven. Niet bijblijven.

KASPAROV FINANCE & BI

macaw

rockfeather

GET RESPONSIVE

ZILVER

VICTA
BUSINESS INTELLIGENCE

Dynamic People

Motion10
AN hso COMPANY

VisionBI
Smart Data Experts

VALID
STAY AHEAD

inergy

valcon

iqbs

Quanto
collective analytics

COMMUNITY

Kimura
data intelligence

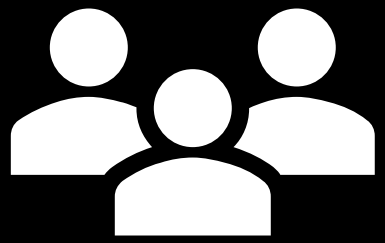
DashData.

volda;
INFORMATIESPECIALISTEN

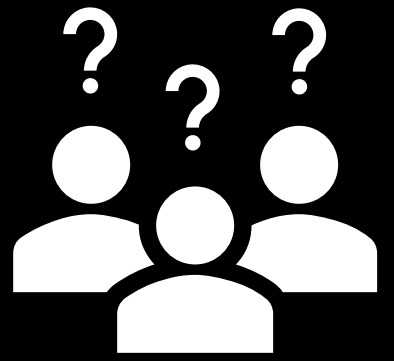
Power BI Connector
by DAVISTA

AZURROFINANCE

Use Case.



Sales & Analytics



Why is the data not as up to date?

Who has access to which reports?

Work smarter, not harder!

What is a Power BI Monitor?

- It helps you get insight and keep track of the existing artifacts in the Power BI Service.
- What can we keep track of?
  - Workspaces: Names, ID, Users
  - Datasets: Allocated Workspace, ID, Refresh schedules, Users, Permissions
  - Reports: Name, ID, Report type, URL, Users , Permissions
  - And the list continues

It answers the what, where, when, who questions of the Power BI artifacts in the Power BI Service.

End product.

- An overview with workspaces
- The datasets within the above workspaces
- The refresh schedules of the datasets within the workspaces

As an additional bonus:

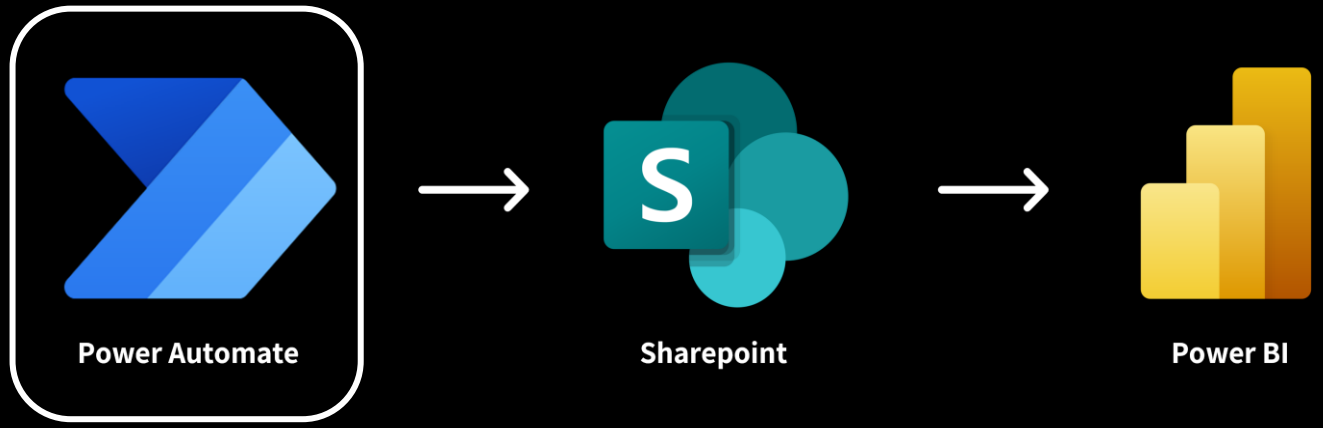
- Automatically retrieve data and refresh dataset

Starting point.

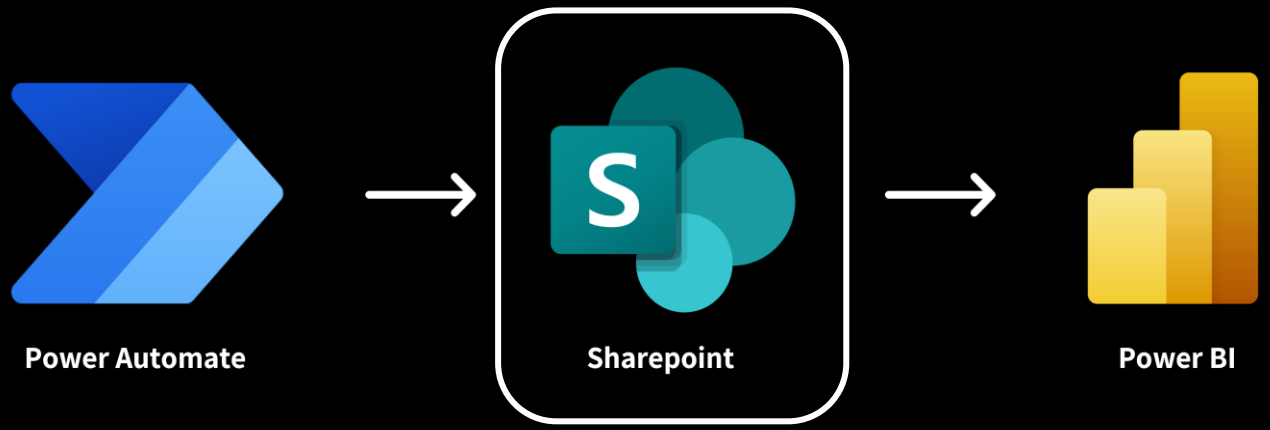
- Citizen developer friendly design (using the Power Platform as much as we can)
- Assuming we, the user, do not have administrator rights to the tenant
- Full end-to-end solution

Solution Design.

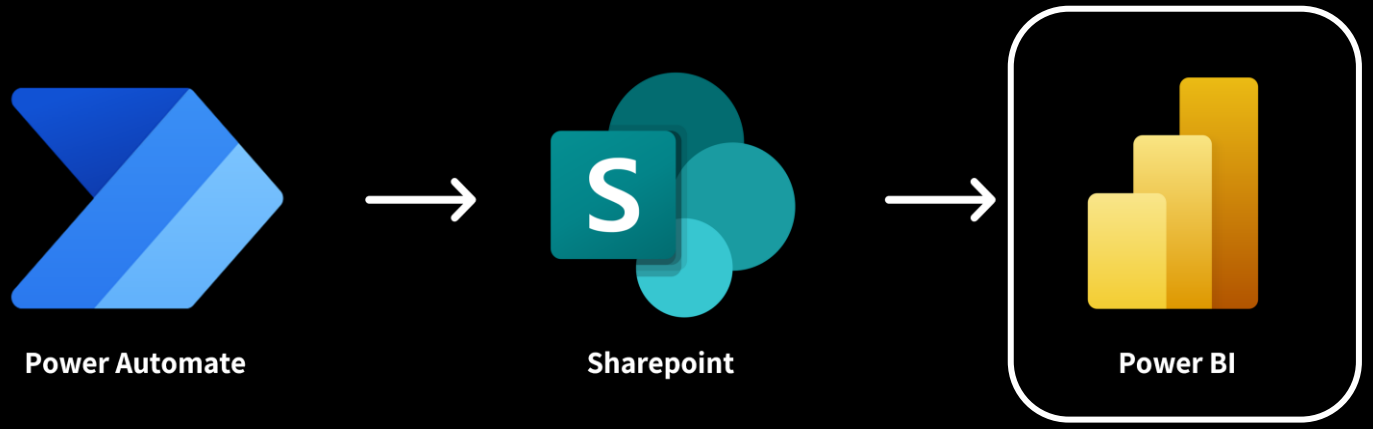
Power Automate → Sharepoint → Power BI



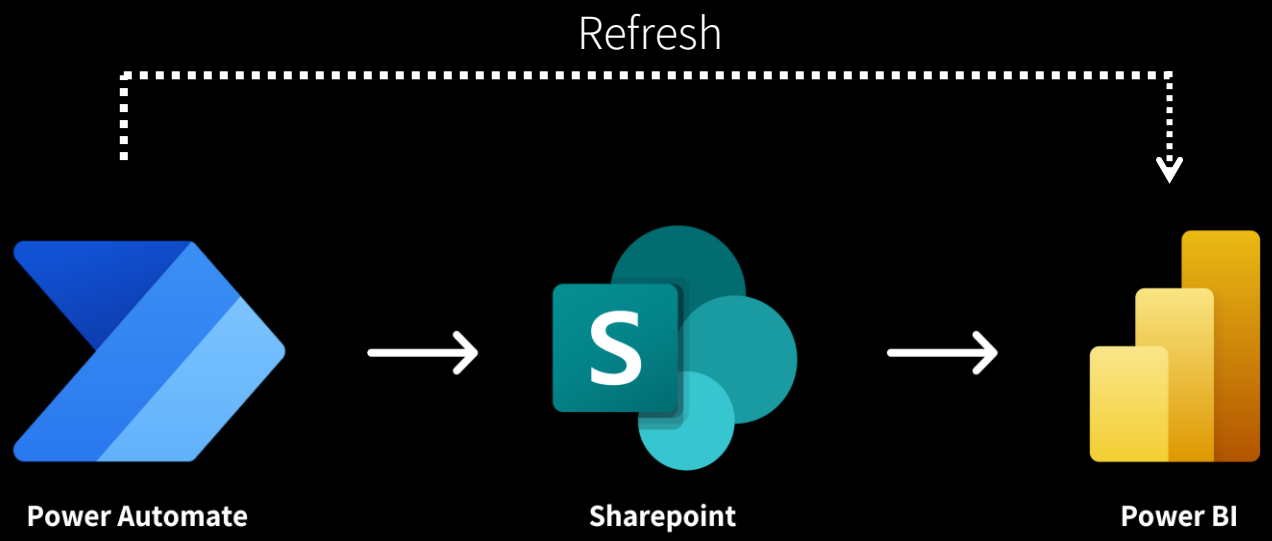
- Call Power BI REST API
- Store results



- Store results



- Retrieve results from sharepoint
- Model results
- Visualize results
- Publish to Power BI Service



- Refresh Power BI dataset with Power Automate

Let’s get started!

What do you need to get started?

- Power Automate license
- Power BI license
- Sharepoint site

Power Automate.

- Create a new Power Automate flow to retrieve the workspaces
  - Create a trigger to trigger the flow
  - Call the Power BI REST API
  - Parse the results of the call
  - Create a table to store the results in
  - Store data in a file on sharepoint

Power Automate.

- Create a new Power Automate flow to retrieve the workspaces
  - Create a trigger to trigger the flow > Manual Trigger
  - Call the Power BI REST API > HTTP Request
  - Parse the results of the call > Parse JSON
  - Create a table to store the results in > Create CSV table
  - Store the file > Create file on sharepoint

Create flow with manual trigger.

- Create a new flow
- Add manual trigger

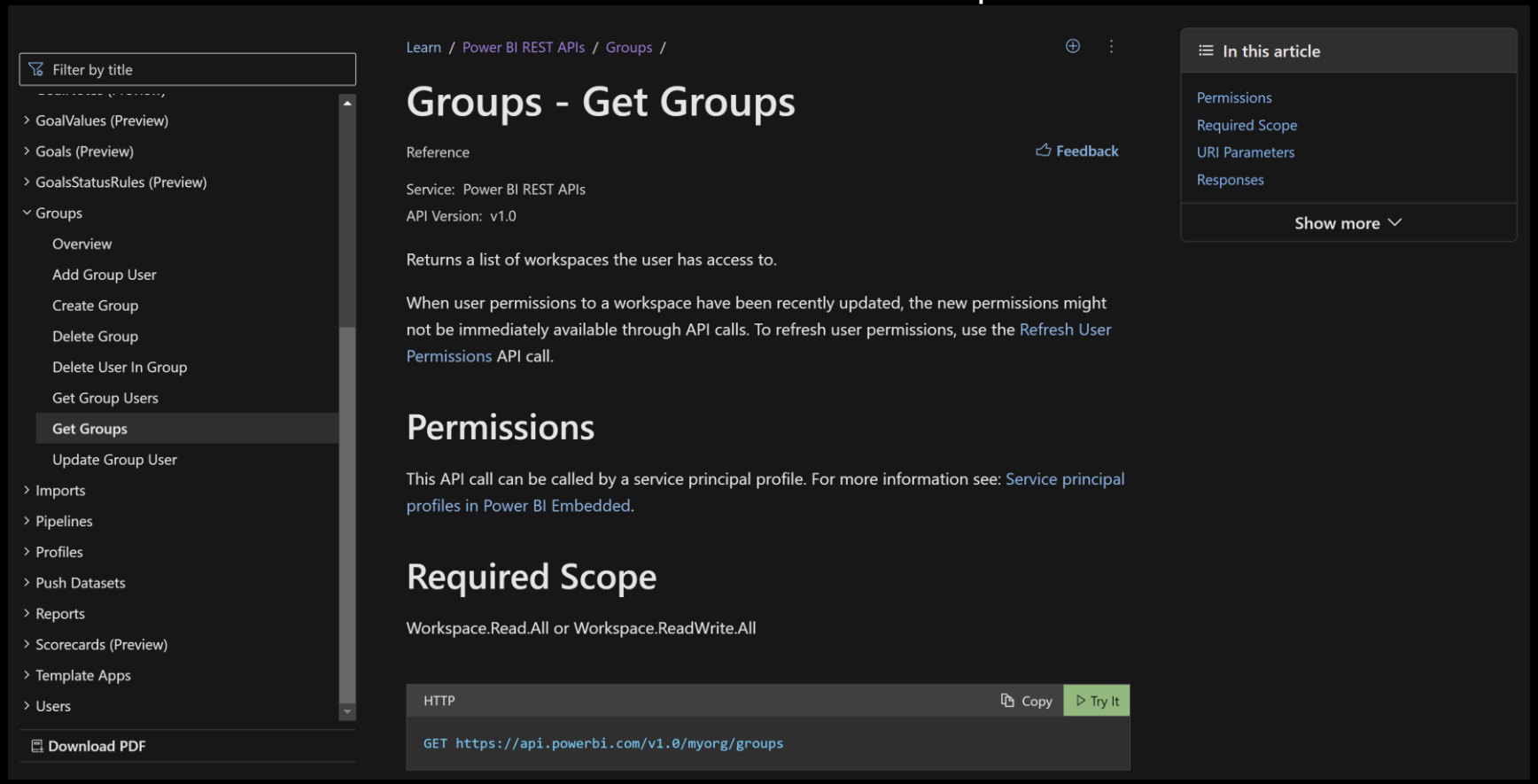
Power BI REST API

- *“The Power BI REST APIs provide service endpoints for embedding, administration, governance and user resources.”*

- With Power BI REST APIs you can do the following:
  - Manage Power BI content
  - Perform admin operations
  - Embed Power BI Content

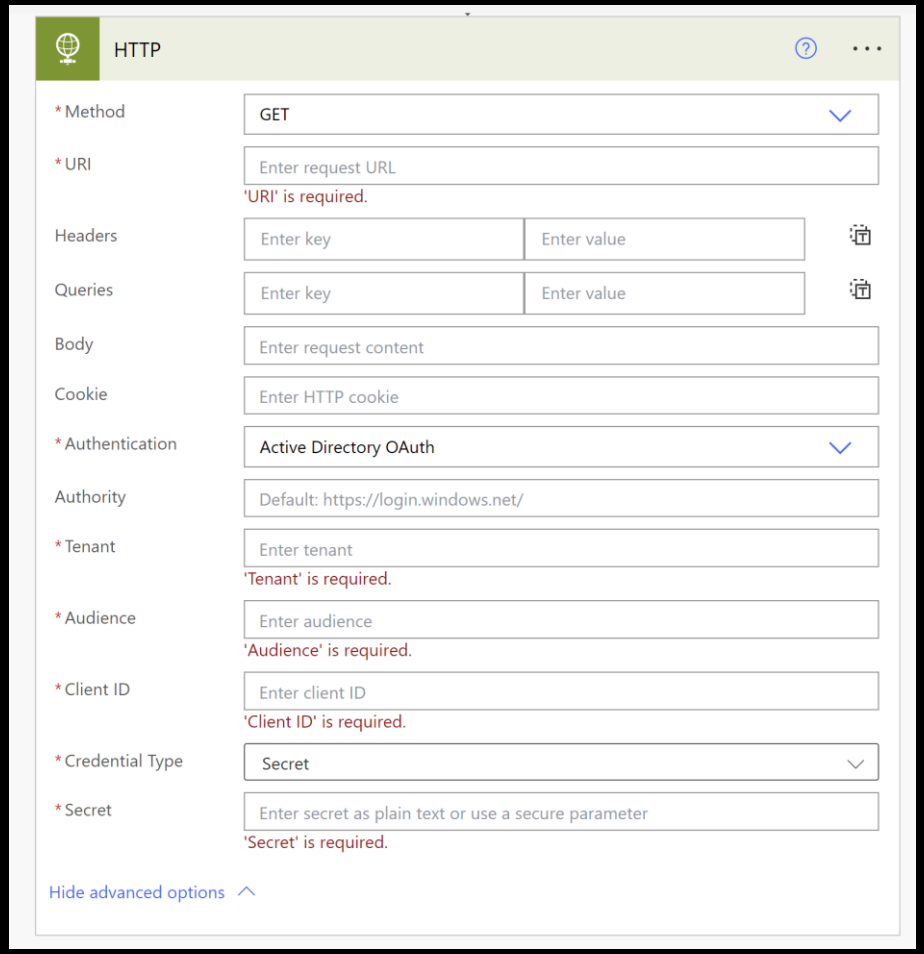
Power BI REST API

- Use this API to retrieve the metadata of the workspaces on the Power BI tenant



HTTP Request

- **Method**: GET
- **URI**: https://api.powerbi.com/v1.0/myorg/groups
- **Authentication**: Active Directory Oautch
- **Tenant**: Tenant ID (retrieved from App registration)
- **Audience**: https://analysis.windows.net/powerbi/api
- **Client ID**: Client ID (retrieved from App registration)
- **Credential Type**: Secret
- **Secret**: Secret value (retrieved from App registration)



& Create and assign service principal.

- Create Azure App Registration
- Grant API Permissions
    1. Add a permission
    2. Power BI Service
    3. Delegated Permissons
    4. Workspace.Read.All
    5. Grant admin consent for MST
    6. Add service principal to workspace

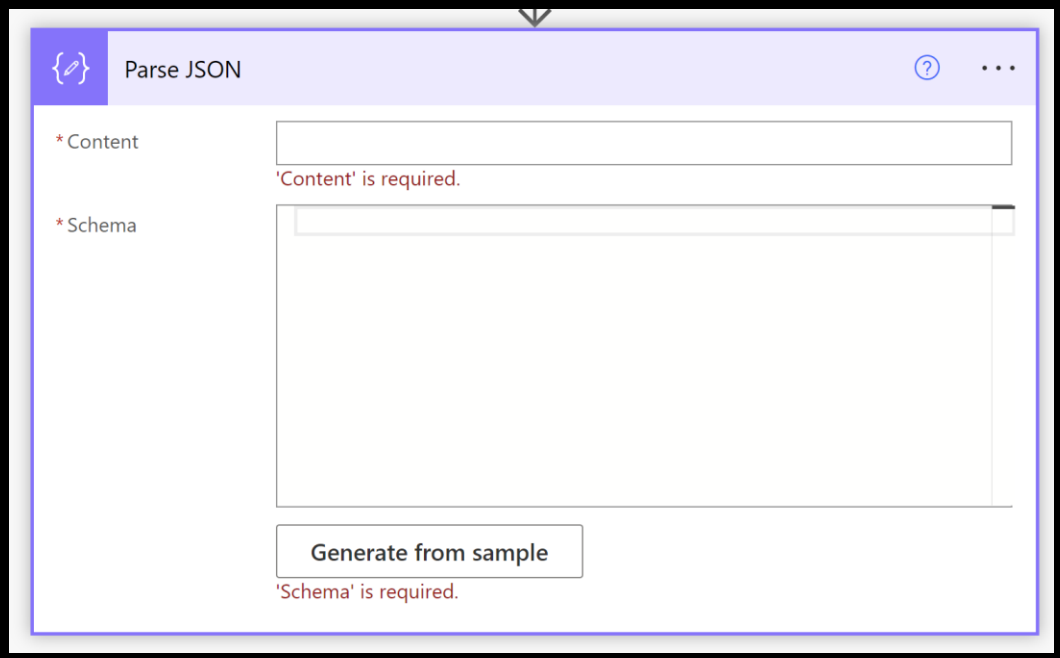
Double check.

- Save the flow
- Let’s check if we get our first results ☺

Parse JSON.

- Parse JSON makes data easier to access and process data later in the process
- Easily reference keys within the body

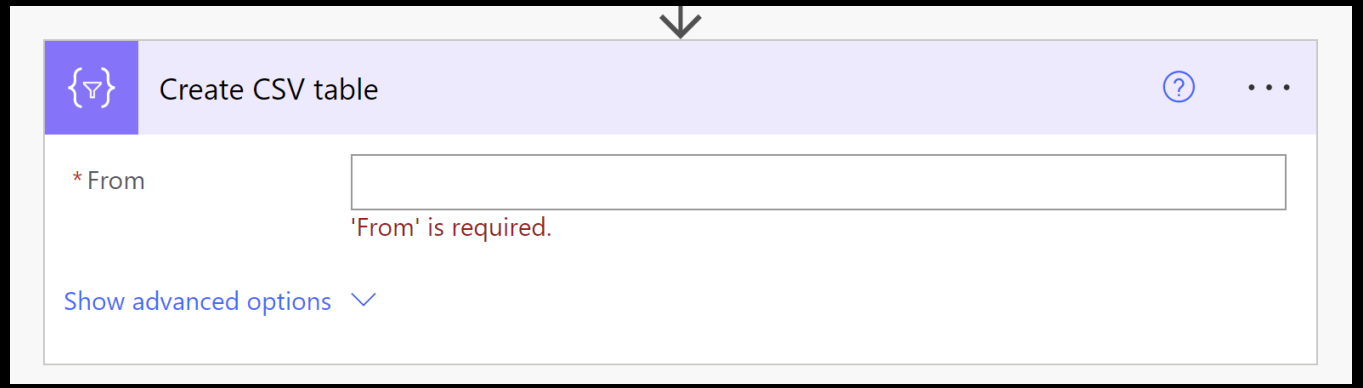
- **Content**: Body
- **Schema**:
  - Generate from sample
  - Paste sample from Microsoft documentation



Create CSV Table.

- Take all the key/value pairs from the array and turn them into individual columns.

- **From**: value

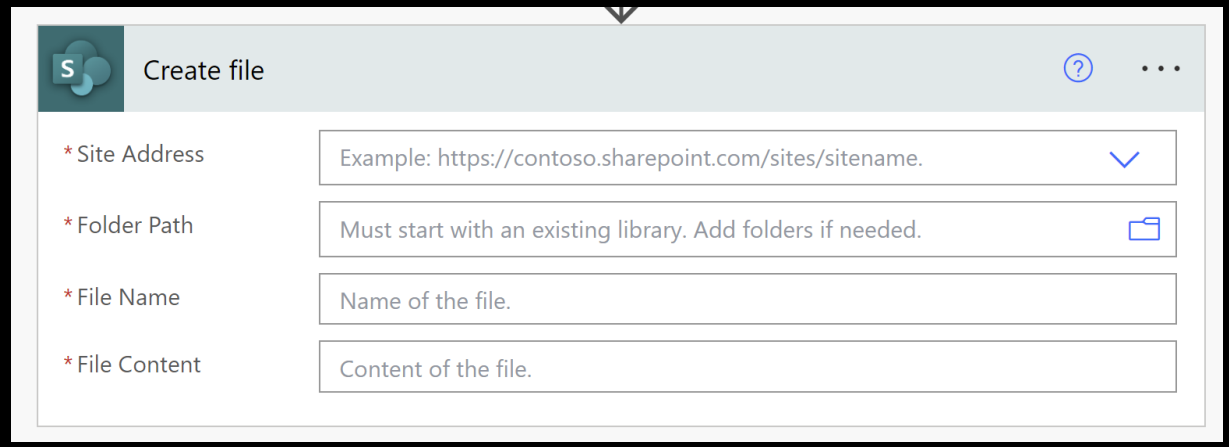


Create file.

- Creates file on given sharepoint location with the specified contents

- **Site Address**: Siteadress
- **Folder Path:** /Shared Documents/General
- **File Name:** GetGroups.csv
- **File Content:** Output

- **Settings:**
  - Allow chunking: Off



Double check.

- Save the flow
- Test the flow
- Check if the results are indeed stored correctly on sharepoint

Create Power BI dataset and report.

- Create a Power BI dataset with this csv file as a source.
  - Retrieve data from the file GetGroups.csv in Sharepoint folder
  - Rename columns
- Create a visualization to show all the workspaces.

- **Data source:** Sharepoint folder
- **Site Address:** Siteadress

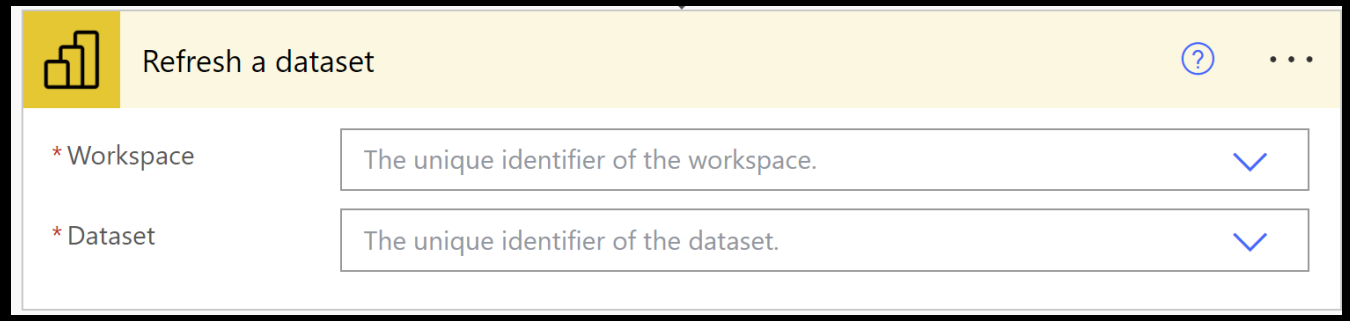
Publish report to Power BI service.

- Create workspace to store the Power BI Monitor
- Assign serice principal
- Publish dataset and report to this workspace
- Edit credentials

Refresh a dataset.

- Refreshes the Power BI dataset

- **Workspace:** Power BI Monitor
- **Dataset:** Power BI Monitor



We did it!

(at least I hope the demo succeeded)

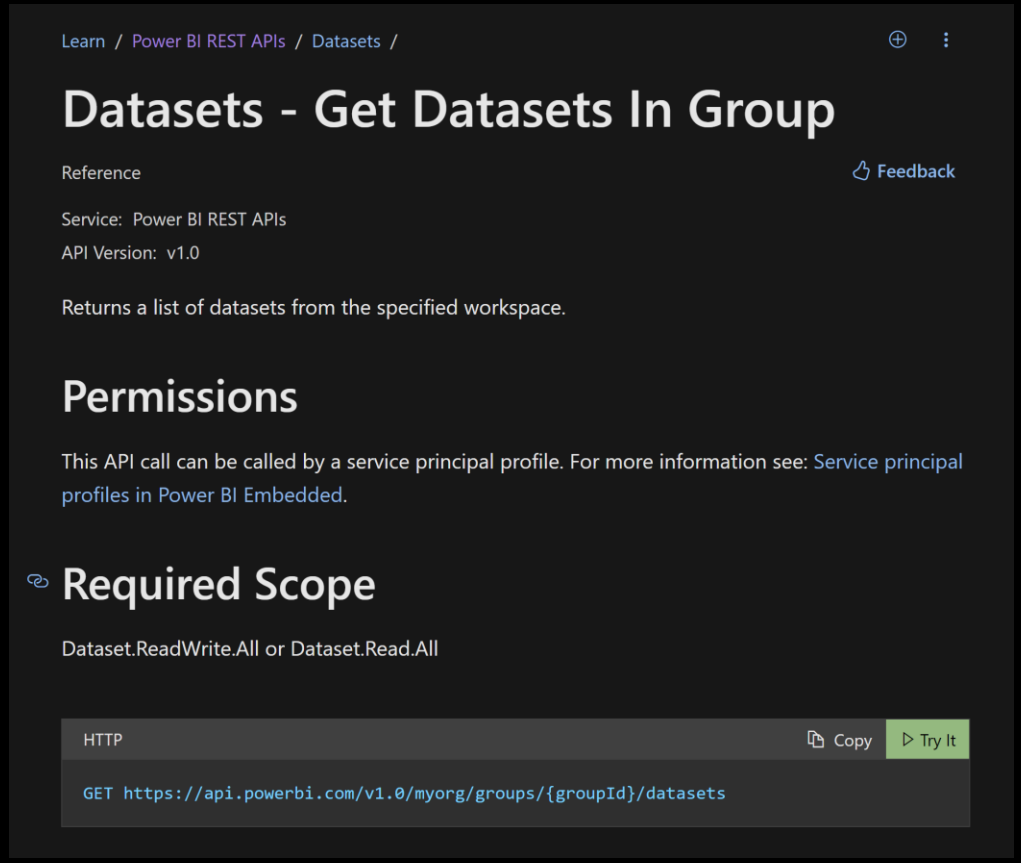
Next steps.

In similar fashion, we can:

- Retrieve all datasets within the workspace
- Retrieve refresh schedules of these datasets

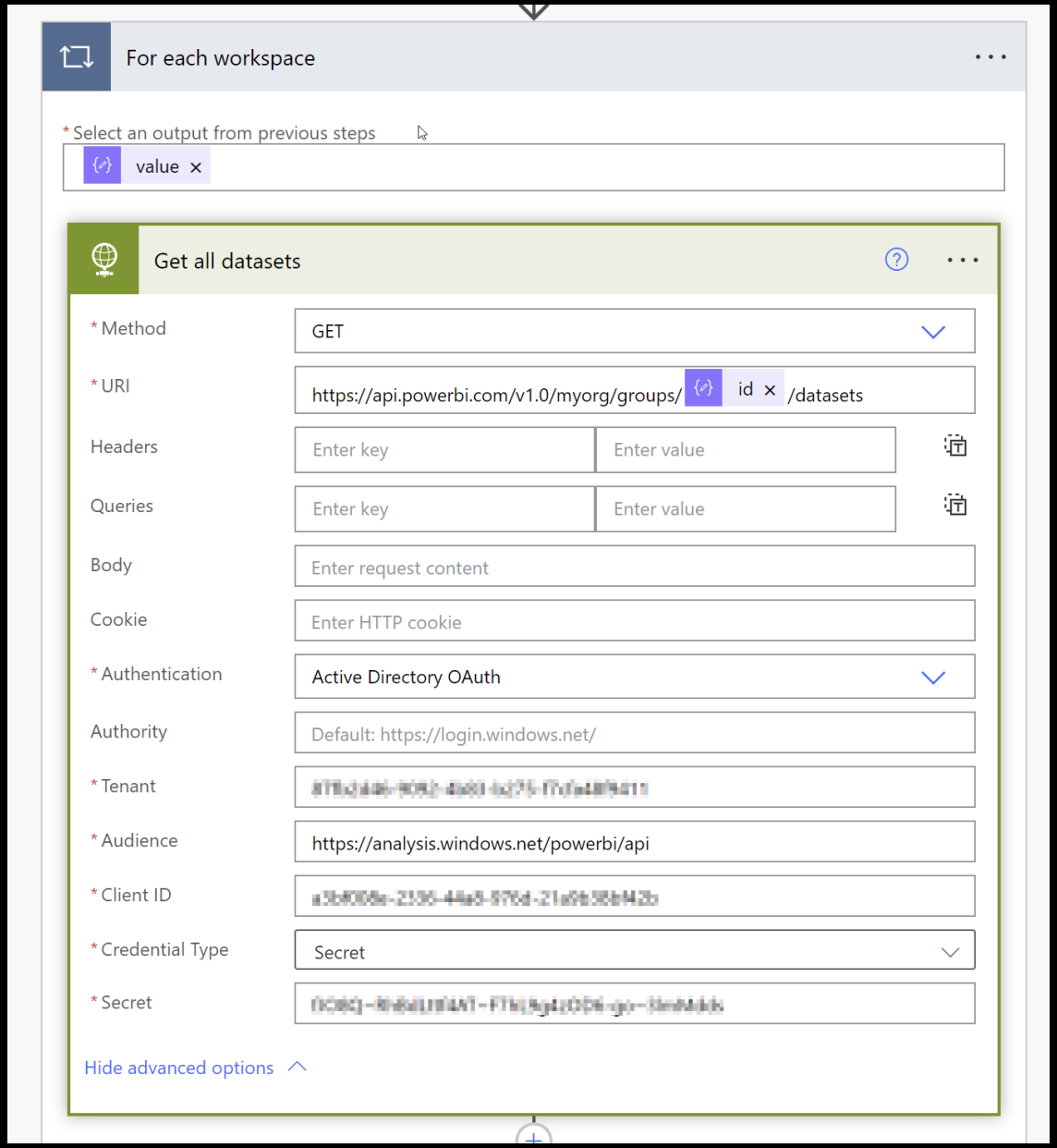
Retrieve Datasets.

- We have to call this API to retrieve the datasets per workspace



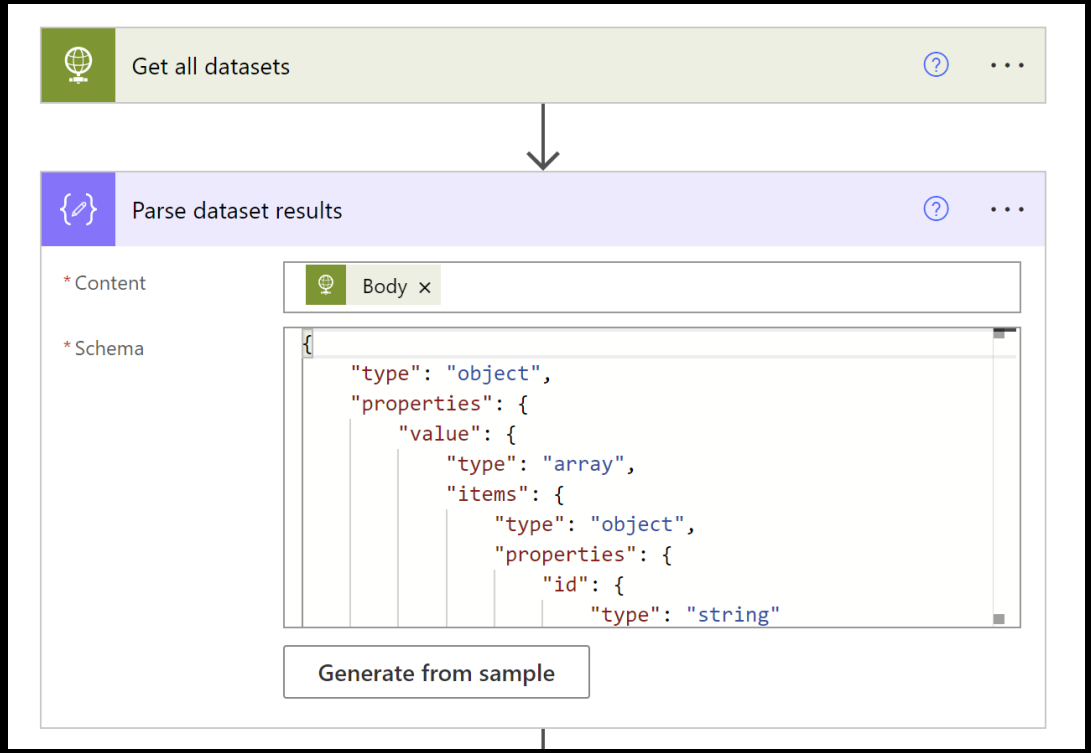
Retrieve Datasets.

- Create HTTP request for each workspace
- Use “Get Datasets in Group” API with “id” as dynamic content
- Automatically creates a “for each” loop
- Authentication is the same as before



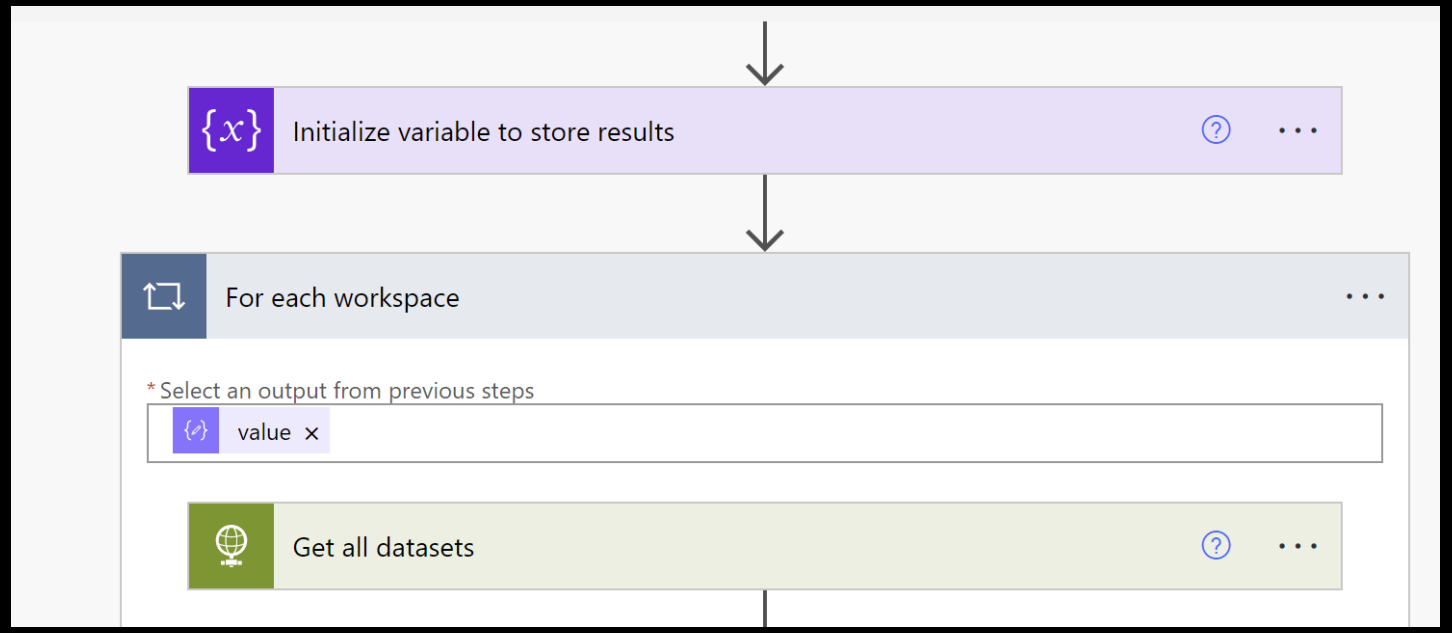
Parse Json.

- Parse results again (with example)



Initialize variable.

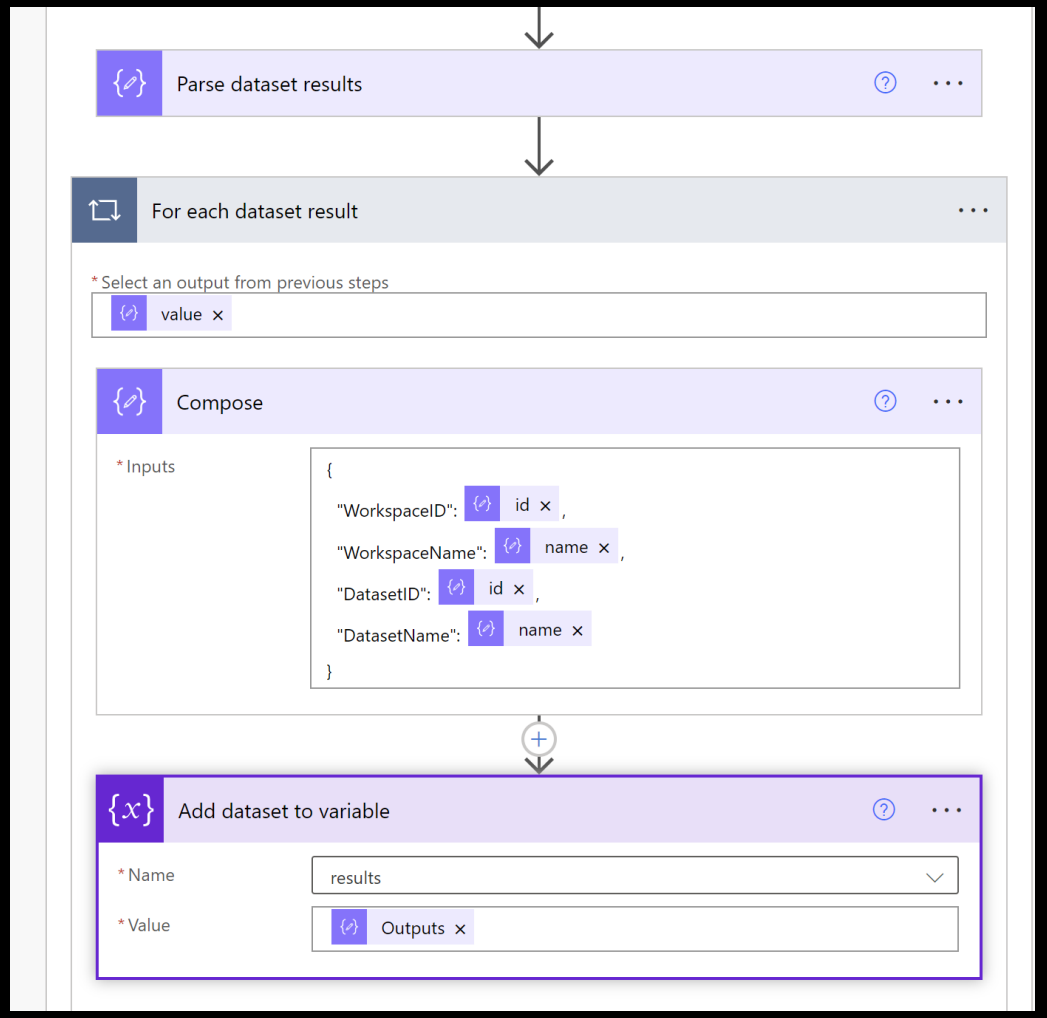
- Initialize variable before calling the API to store the details of each workspace – dataset combination



Save results in variable.

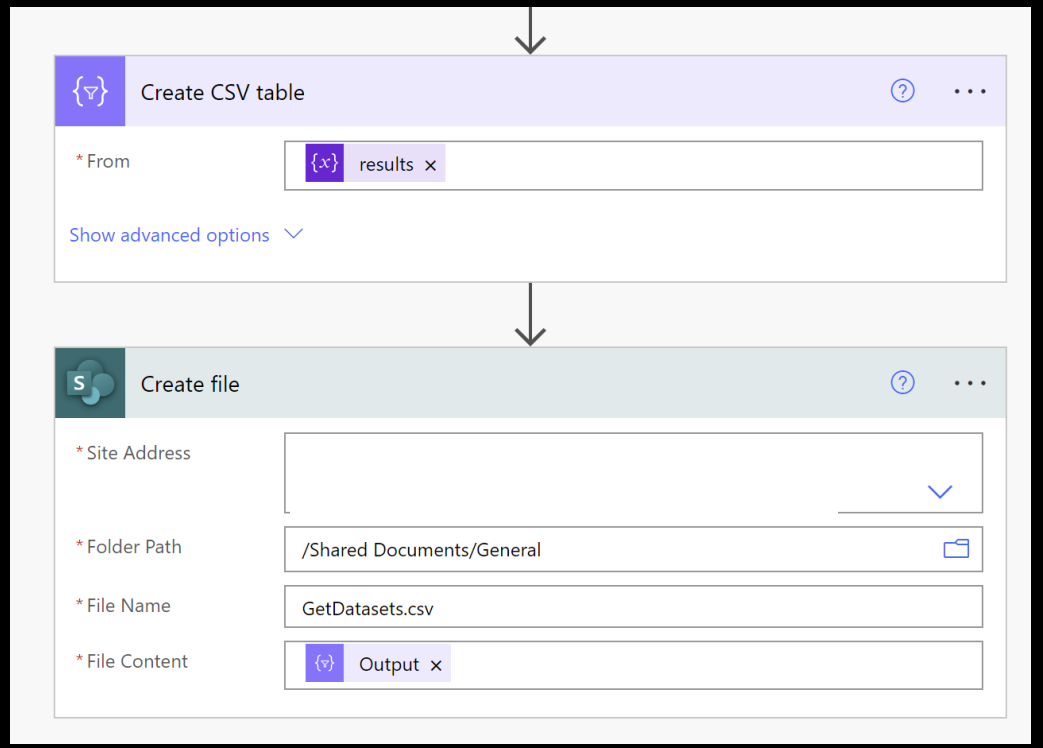
- Define what needs to be saved
- For each combination of below entities:
  - WorkspaceID
  - WorkspaceName
  - DatasetID
  - DatasetName

- Append to variable “results”



Store results on Sharepoint.

- Create CSV table with contents of variable
- Create and store a file out of CSV table

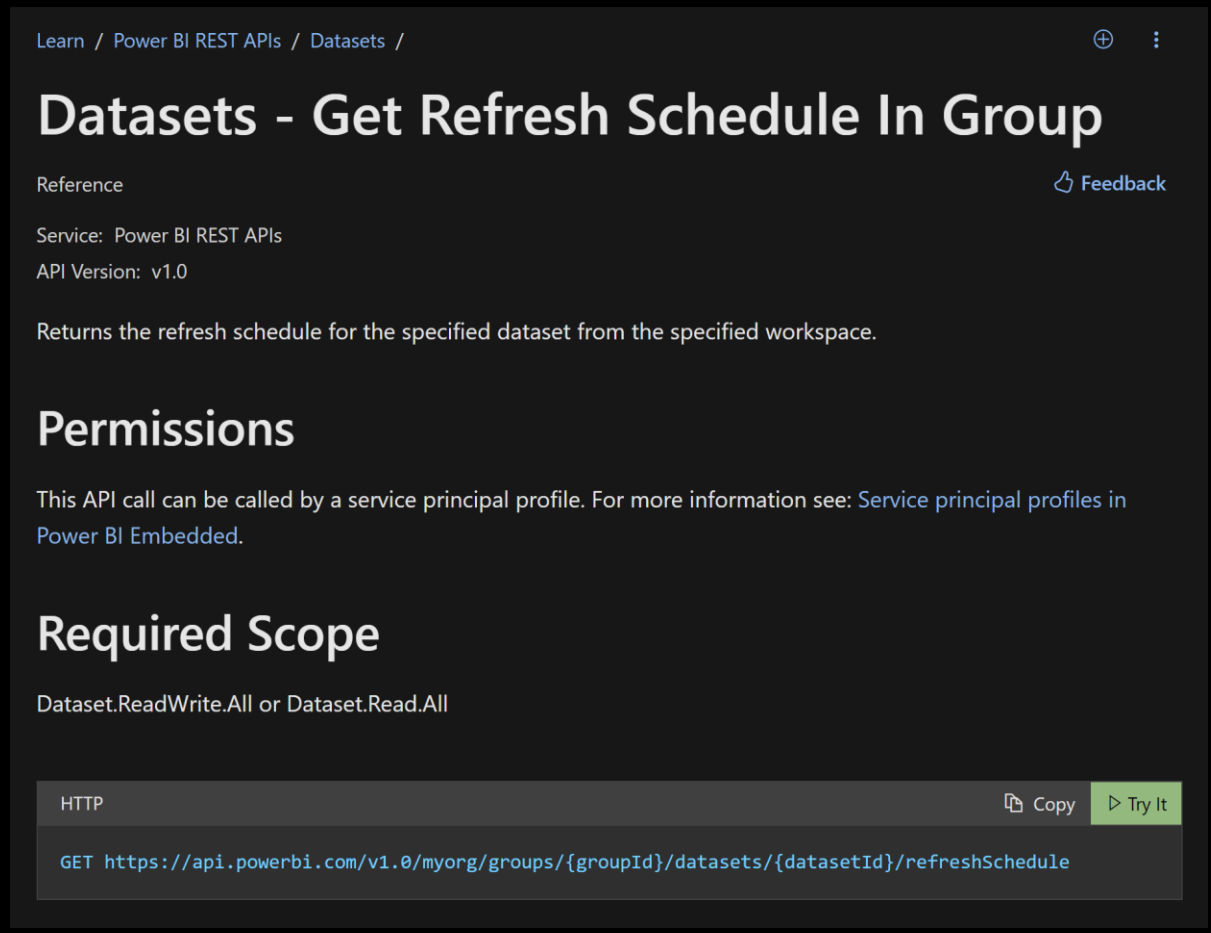


File content.

WorkspaceID	WorkspaceName	DatasetID	DatasetName
578d0ef4-0870-4902-8fdc-d808f07d29c2	Sales	9bbfe1f3-0dc5-4d67-ba43-8da299155673	CRP
578d0ef4-0870-4902-8fdc-d808f07d29c2	Sales	f787198d-4156-4470-9435-2b848833ab46	Sales & Returns
578d0ef4-0870-4902-8fdc-d808f07d29c2	Sales	35e34da7-807a-47ba-8b8b-307019563326	Sales Categories
578d0ef4-0870-4902-8fdc-d808f07d29c2	Sales	9ec61a3d-8c8a-4445-ac17-b55a471368c6	Sales
6922dd40-f9fb-49e4-bce7-f3d032541cb7	Finance	2ee77242-95b2-4858-ba5f-2b132a12ad29	Pricing
6922dd40-f9fb-49e4-bce7-f3d032541cb7	Finance	dbbaf9c94-8f8f-4084-b382-71cebef36de7	Costs
f5413b66-d11e-412a-afc8-176a58e0fee7	HR	42826845-767b-4a02-9d14-56075846c434	Recruitment
62066cd5-ff9f-45e3-ac79-11f1b838cbfa	Finance [DEV]	e3b14201-f923-4bd6-b2de-58a6c40fca0b	Pricing
62066cd5-ff9f-45e3-ac79-11f1b838cbfa	Finance [DEV]	c01aaaaa-780d-4272-b15f-451de764c700	Costs
5d776429-dd57-4767-ae32-c8fc5f77f23e	Finance [TST]	b8c5e907-d1bf-4261-b6d0-338885cb97b6	Costs
5d776429-dd57-4767-ae32-c8fc5f77f23e	Finance [TST]	5fe45052-2c99-47b2-aa07-63dfceb8123c	Pricing

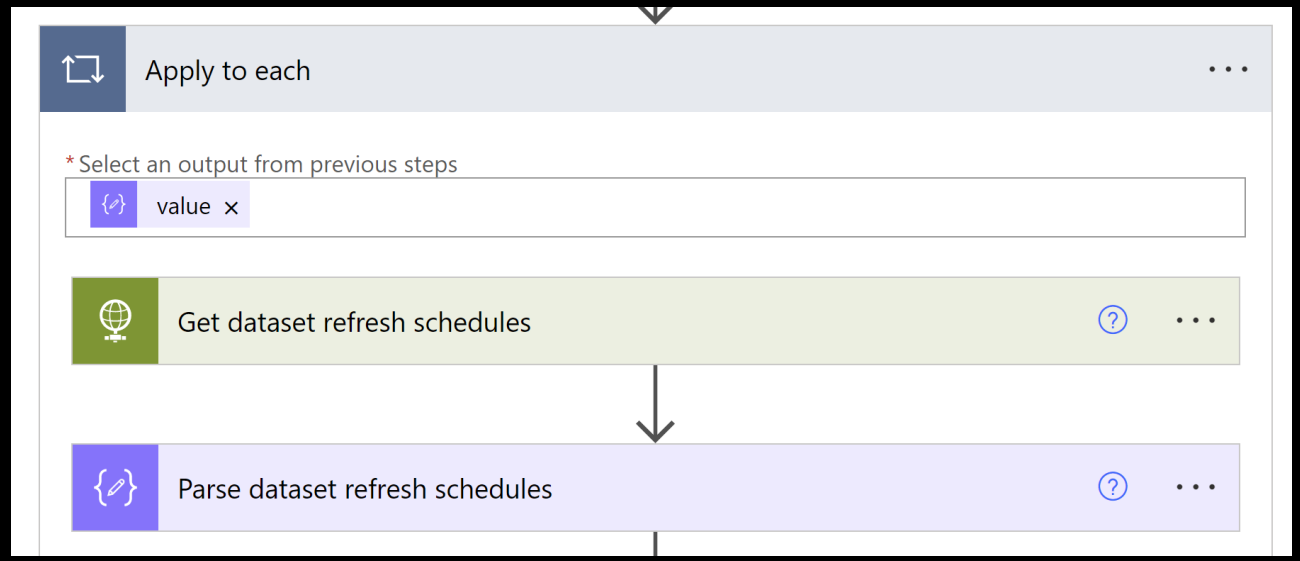
Retrieve Refresh Schedules.

- We have to call this API to retrieve the refresh schedules of the datasets



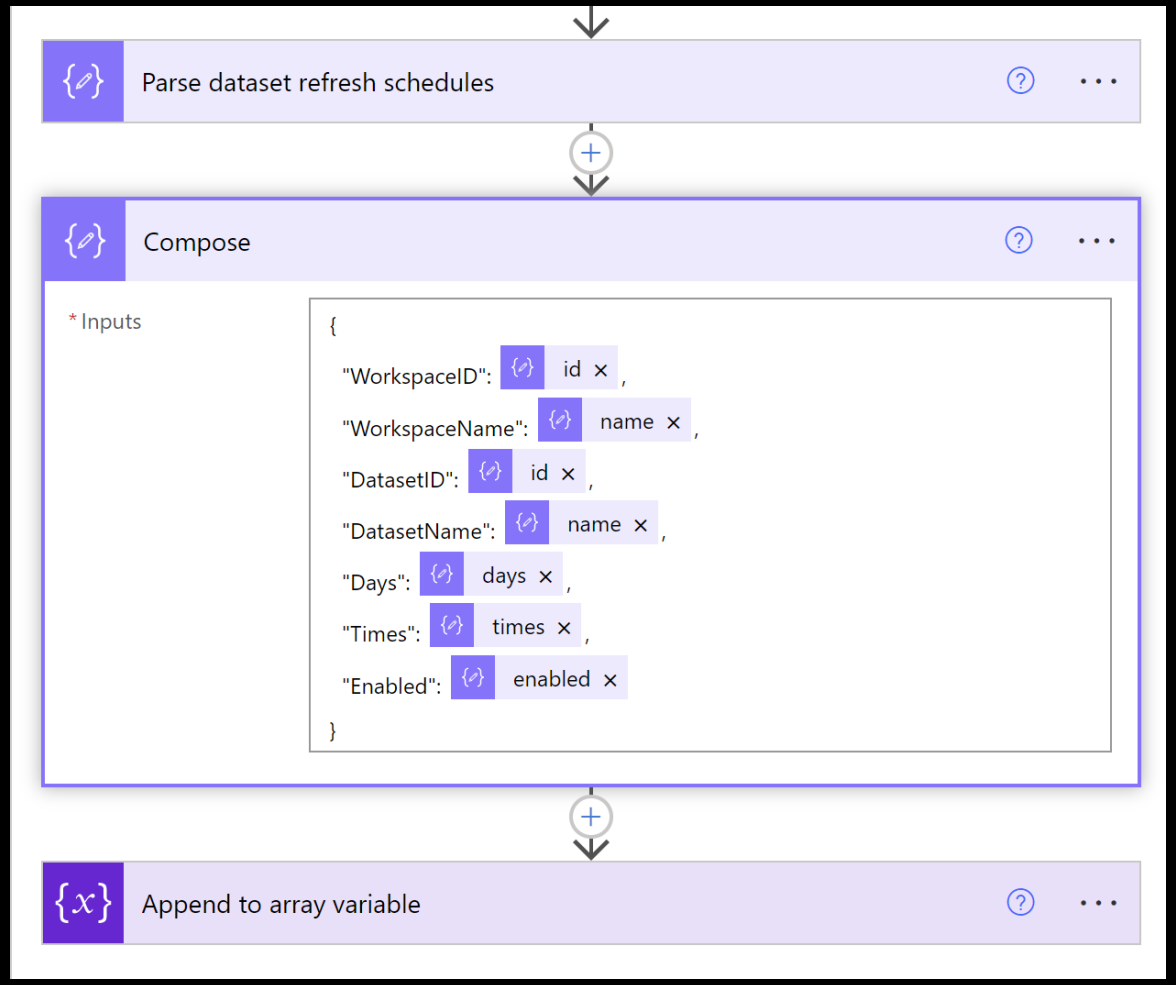
Retrieve Refresh Schedules.

- Retrieve all refresh schedules per dataset with an HTTP request
- Parse results



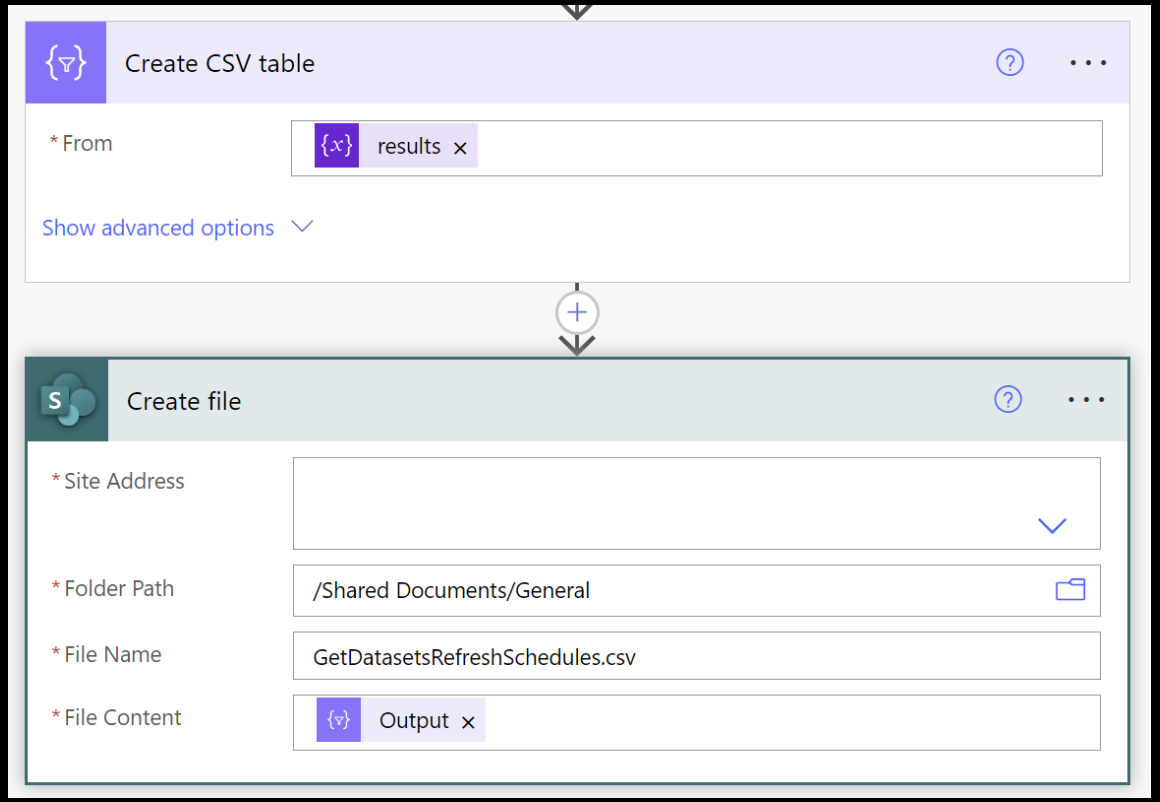
Save results.

- Define what needs to be saved
- For each combination of below entities:
  - Workspace
  - Dataset
  - Refresh Schedule
- Append to variable “results”



Store results on Sharepoint.

- Save the array “results” in CSV table
- Store CSV table on sharepoint



Expand Power BI Report.

- Expand Power BI report with new information
  - Datasets per workspace
  - Refresh schedules per dataset

Improvemens and variations.

- Store credentials in Azure Key Vault
- Instead of using Power Automate and Sharepoint, you can choose between many tools to retrieve and store the results, e.g.:
  - Logic Apps and Log Analytics
  - Databricks

Wrap up.

- End-to-end solution to monitor your Power BI environment
  - Retrieve workspaces
  - Retrieve datasets
  - Retrieve refresh schedules

- Power Automate – Sharepoint – Power BI

- Play around and have fun!

Thank you for your attention!

Questions?