



**POWER PLATFORM
BOOTCAMP**



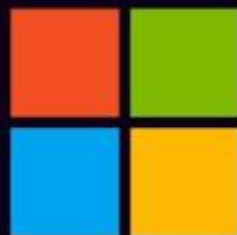
dataMinds

GLOBAL POWER PLATFORM BOOTCAMP 2023 *BELGIUM EDITION*



GeoPostcodes

U2U



Microsoft



SPONSORS



Get started with your own Power BI Monitor!

25-02-2023





Paulien van Eijk



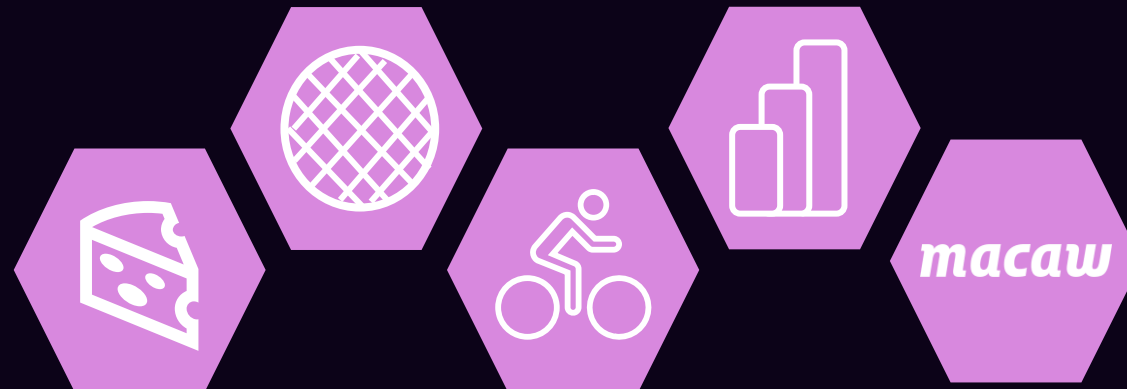
Power BI Solution Architect



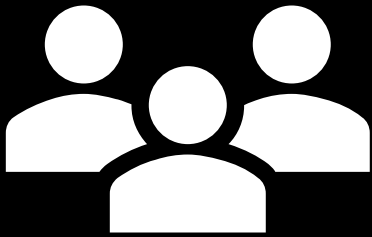
Paulien.van.eijk@macaw.nl



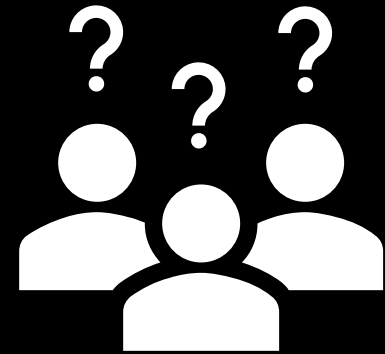
<https://www.linkedin.com/in/paulien-van-eijk>



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.



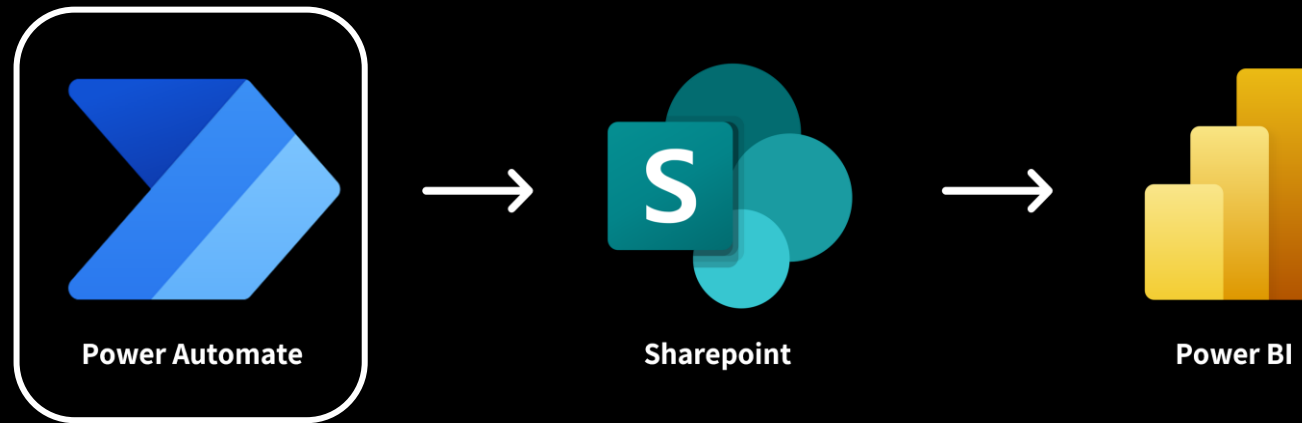
Power Automate



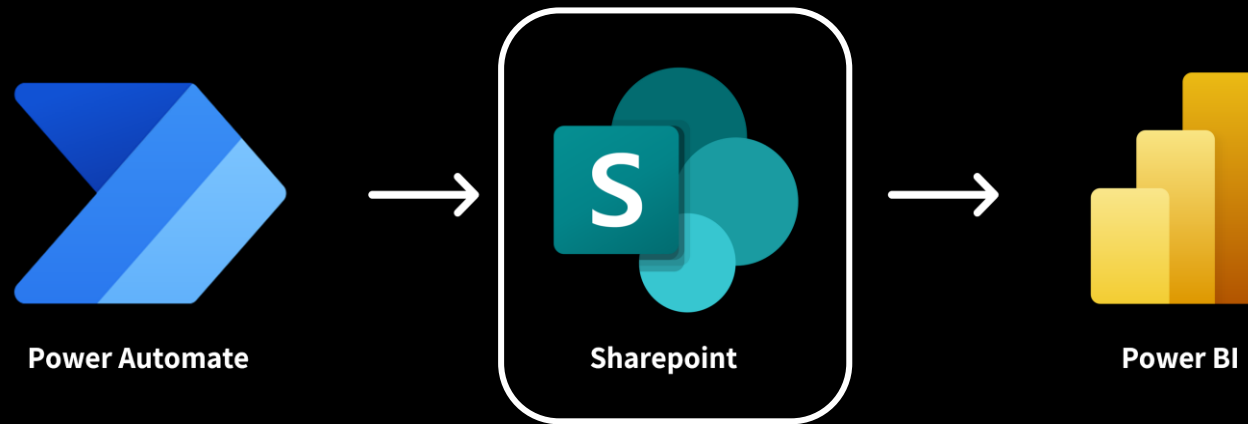
Sharepoint



Power BI



- Call Power BI REST API
- Store results



- Store results



Power Automate

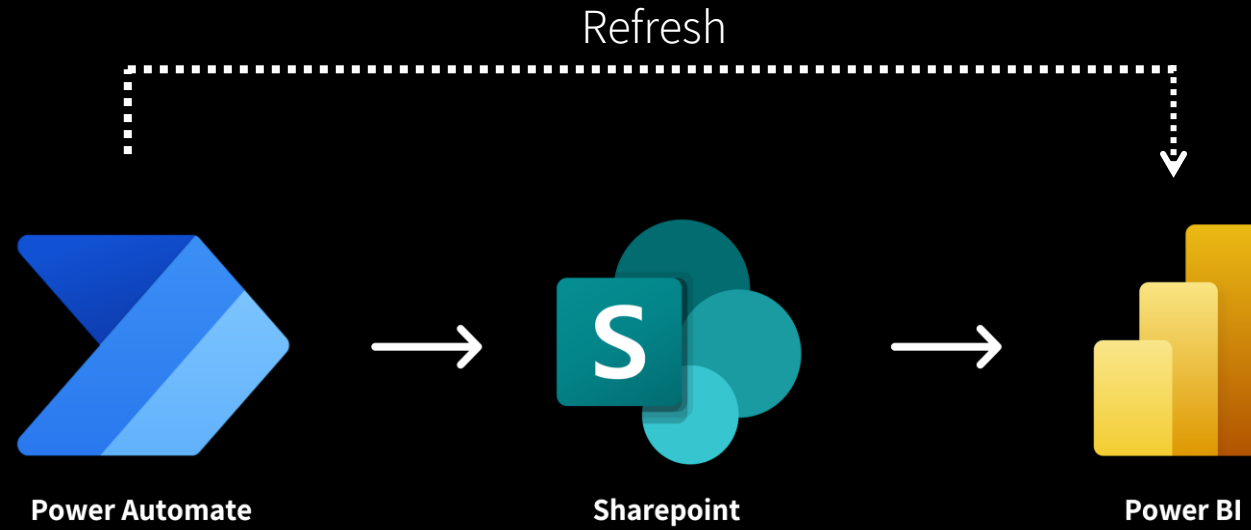


Sharepoint



Power BI

- 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

The screenshot shows the Microsoft Learn documentation page for the 'Groups - Get Groups' API endpoint. The page is dark-themed and includes a left-hand navigation pane, a main content area, and a right-hand 'In this article' sidebar.

Left-hand navigation pane: A search bar at the top is followed by a list of categories: GoalValues (Preview), Goals (Preview), GoalsStatusRules (Preview), Groups (expanded), Imports, Pipelines, Profiles, Push Datasets, Reports, Scorecards (Preview), Template Apps, and Users. Under the 'Groups' category, the following sub-items are listed: Overview, Add Group User, Create Group, Delete Group, Delete User In Group, Get Group Users, **Get Groups** (highlighted), and Update Group User. A 'Download PDF' link is at the bottom of the pane.

Main content area: The title 'Groups - Get Groups' is prominently displayed. Below it, the breadcrumb 'Learn / Power BI REST APIs / Groups /' is shown. A 'Reference' section indicates the 'Service: Power BI REST APIs' and 'API Version: v1.0'. A 'Feedback' link is present. The description states: 'Returns a list of workspaces the user has access to.' A note mentions that user permissions might not be immediately available through API calls and suggests using the 'Refresh User Permissions' API call. The 'Permissions' section states: 'This API call can be called by a service principal profile. For more information see: Service principal profiles in Power BI Embedded.' The 'Required Scope' section lists 'Workspace.Read.All or Workspace.ReadWrite.All'. At the bottom, a code block shows the HTTP method 'GET' and the URL 'https://api.powerbi.com/v1.0/myorg/groups', with 'Copy' and 'Try It' buttons.

Right-hand sidebar: Titled 'In this article', it contains a list of links: Permissions, Required Scope, URI Parameters, and Responses. A 'Show more' link with a downward arrow is at the bottom.

HTTP Request

- **Method:** GET
- **URI:** <https://api.powerbi.com/v1.0/myorg/groups>
- **Authentication:** Active Directory OAuth
- **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)

The screenshot shows an HTTP client interface with the following fields and values:

- Method:** GET
- URI:** Enter request URL (with a red error message: 'URI' is required.)
- Headers:** Enter key / Enter value
- Queries:** Enter key / Enter value
- Body:** Enter request content
- Cookie:** Enter HTTP cookie
- Authentication:** Active Directory OAuth
- Authority:** Default: https://login.windows.net/
- Tenant:** Enter tenant (with a red error message: 'Tenant' is required.)
- Audience:** Enter audience (with a red error message: 'Audience' is required.)
- Client ID:** Enter client ID (with a red error message: 'Client ID' is required.)
- Credential Type:** Secret
- Secret:** Enter secret as plain text or use a secure parameter (with a red error message: 'Secret' is required.)

At the bottom, there is a link: [Hide advanced options](#) with an upward arrow icon.

& Create and assign service principal.

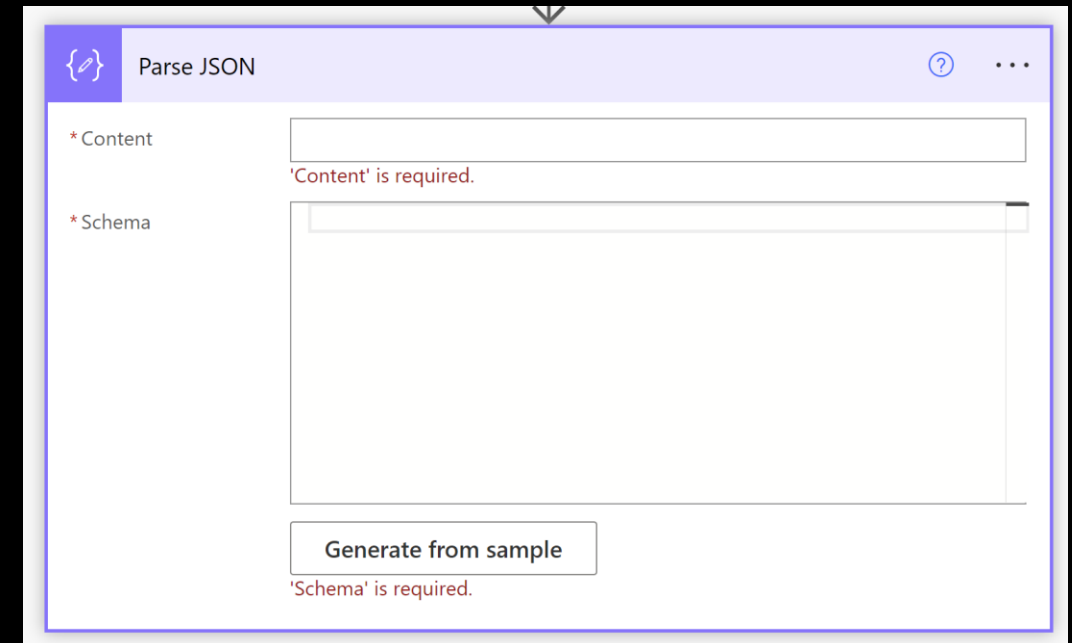
- Create Azure App Registration
- Grant API Permissions
 1. Add a permission
 2. Power BI Service
 3. Delegated Permissions
 4. Workspace.Read.All
 5. Grant admin consent for MST
 6. Add service principal to workspace

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

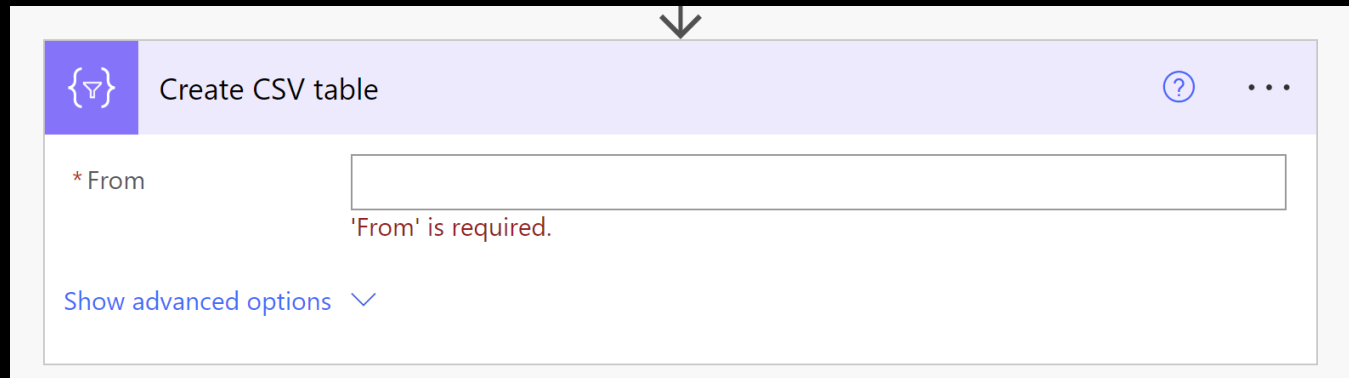


The screenshot shows a web application titled "Parse JSON". It features two input fields: "Content" and "Schema". The "Content" field has a red error message below it: "'Content' is required." The "Schema" field has a red error message below it: "'Schema' is required." Below these fields is a button labeled "Generate from sample".

➤ Create CSV Table.

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

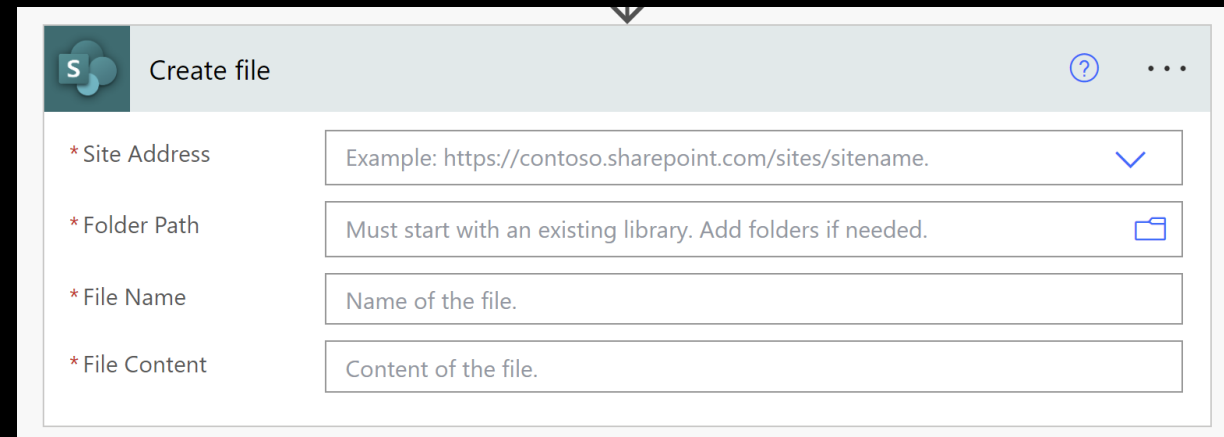
- From:** value



The screenshot shows a web form titled "Create CSV table". The form has a header bar with a blue icon on the left, the title "Create CSV table" in the center, and a help icon and a menu icon on the right. Below the header, there is a field labeled "*From" with a red asterisk indicating it is required. The field is empty, and a red error message "'From' is required." is displayed below it. At the bottom of the form, there is a link "Show advanced options" with a downward arrow.

➤ Create file.

- Creates file on given sharepoint location with the specified contents
- **Site Address:** Siteaddress here
- **Folder Path:** /Shared Documents/General
- **File Name:** GetGroups.csv
- **File Content:** Output
- **Settings:**
 - Allow chunking: Off



The screenshot shows the 'Create file' dialog box in a SharePoint interface. The dialog has a title bar with the SharePoint logo and the text 'Create file'. Below the title bar, there are four input fields, each with a red asterisk indicating a required field. The first field is 'Site Address' with a placeholder text 'Example: https://contoso.sharepoint.com/sites/sitename.' and a dropdown arrow. The second field is 'Folder Path' with a placeholder text 'Must start with an existing library. Add folders if needed.' and a folder icon. The third field is 'File Name' with a placeholder text 'Name of the file.'. The fourth field is 'File Content' with a placeholder text 'Content of the file.'.

Field	Placeholder/Value
* Site Address	Example: https://contoso.sharepoint.com/sites/sitename.
* Folder Path	Must start with an existing library. Add folders if needed.
* File Name	Name of the file.
* File Content	Content of the file.

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:** Site address here

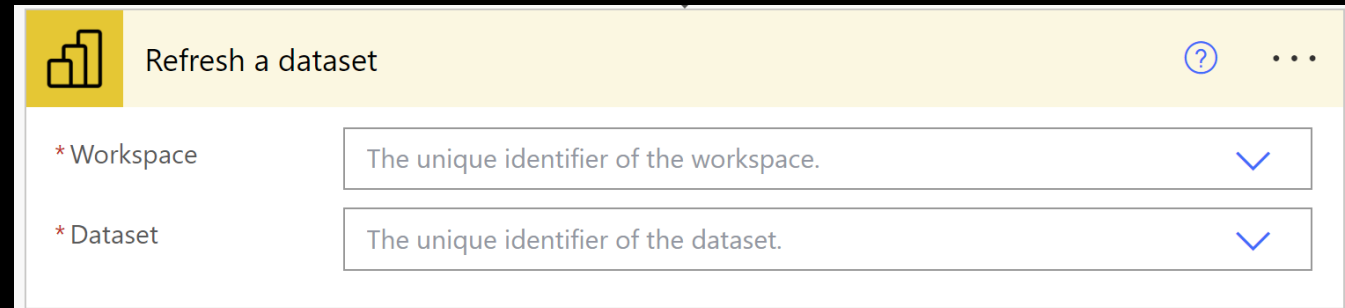


Publish report to Power BI service.

- Create workspace to store the Power BI Monitor
- Assign service 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



Refresh a dataset	
* Workspace	The unique identifier of the workspace.
* Dataset	The unique identifier of the dataset.

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

Learn / Power BI REST APIs / Datasets /

Datasets - Get Datasets In Group

Reference [Feedback](#)

Service: Power BI REST APIs
API Version: v1.0

Returns a list of datasets from the specified workspace.

Permissions

This API call can be called by a service principal profile. For more information see: [Service principal profiles in Power BI Embedded](#).

Required Scope

Dataset.ReadWrite.All or Dataset.Read.All

HTTP [Copy](#) [Try It](#)

```
GET https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets
```

➤ 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

The screenshot displays the configuration for a REST API connector in a workflow. At the top, a 'For each workspace' loop is established, with a selection of 'value' from a previous step. Below this, the 'Get all datasets' action is configured with the following details:

- Method:** GET
- URI:** `https://api.powerbi.com/v1.0/myorg/groups/{id}/datasets` (where {id} is a dynamic content placeholder)
- Headers:** Two input fields for 'Enter key' and 'Enter value'.
- Queries:** Two input fields for 'Enter key' and 'Enter value'.
- Body:** 'Enter request content'
- Cookie:** 'Enter HTTP cookie'
- Authentication:** Active Directory OAuth
- Authority:** Default: `https://login.windows.net/`
- Tenant:** `871b2d46-9092-4b60-b275-770640f94111`
- Audience:** `https://analysis.windows.net/powerbi/api`
- Client ID:** `a5b006a-2330-44a3-976d-21a0b38b412b`
- Credential Type:** Secret
- Secret:** `f1c0c3-8b6a1m1447-f71q8g4z00b6-go-8enM4ids`

A 'Hide advanced options' link is visible at the bottom of the configuration panel.

Parse Json.

- Parse results again (with example)

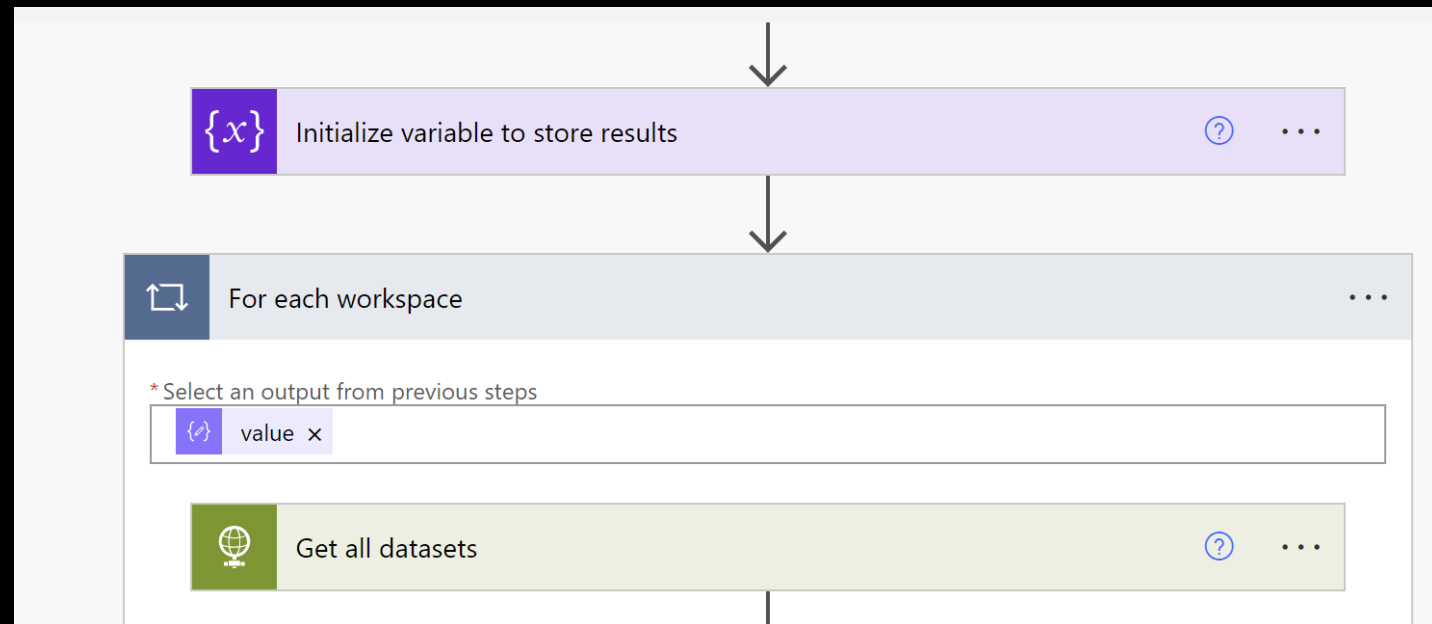
The screenshot shows a web interface with two main sections. The top section, titled "Get all datasets" with a globe icon, has a light green header. An arrow points down to the second section, "Parse dataset results", which has a light purple header and a curly brace icon. Below the header, there are two tabs: "Content" and "Schema". The "Schema" tab is active, showing a JSON schema in a text editor. The schema defines a type of "object" with a "properties" object containing a "value" property of type "array". This array has "items" of type "object", which in turn has a "properties" object with an "id" property of type "string". A "Generate from sample" button is located at the bottom of the schema editor.

```
{
  "type": "object",
  "properties": {
    "value": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

Generate from sample

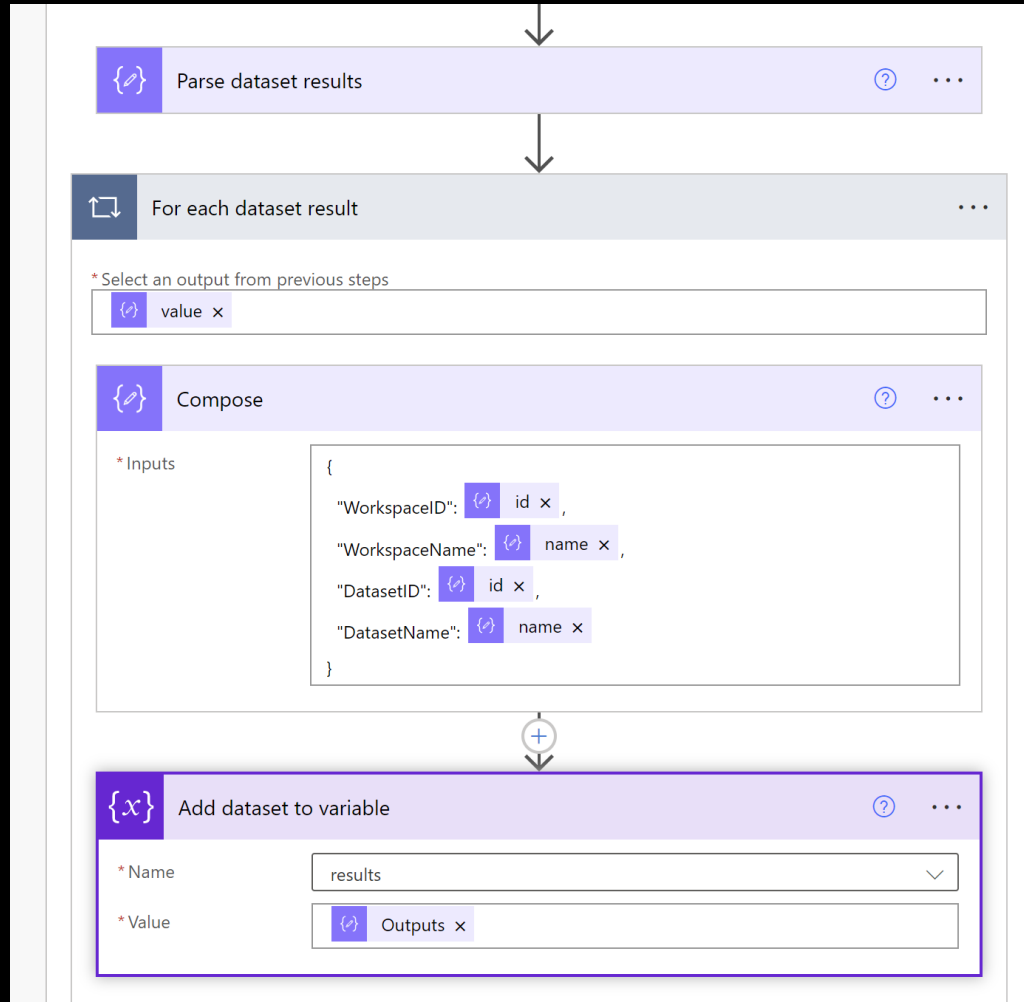
➤ Initialize variable.

- 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

The screenshot illustrates a two-step process in Power BI Desktop for storing results on SharePoint. The first step, 'Create CSV table', shows a variable named 'results' being used as the source for the CSV data. The second step, 'Create file', shows the file being saved to a SharePoint site. The file name is 'GetDatasets.csv' and the content is the 'Output' of the previous step. The SharePoint site address is 'Power BI Monitor Solutions' and the folder path is '/Shared Documents/General'.

Create CSV table

* From: {x} results x

Show advanced options ▾

Create file

* Site Address: Power BI Monitor Solutions ▾

* Folder Path: /Shared Documents/General ▾

* File Name: GetDatasets.csv

* File Content: {v} Output x



File content.

WorkspaceID	WorkspaceName	DatasetID	DatasetName
578d0ef4-0870-4902-8f6c-d808f07d29c2	Sales	9b6fe1f3-0dc5-4d67-ba43-8da299135673	CRP
578d0ef4-0870-4902-8f6c-d808f07d29c2	Sales	f787196d-4156-4476-9435-2b848833ab46	Sales & Returns
578d0ef4-0870-4902-8f6c-d808f07d29c2	Sales	35e34da7-807a-47ba-8b8b-307019563326	Sales Categories
578d0ef4-0870-4902-8f6c-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	dbba9c94-8f8f-4084-b382-71cebef36de7	Costs
f5413b66-d11e-412a-afc8-176a58e0fee7	HR	42826843-767b-4a02-9d14-5607584bc434	Recruitment
62066cd5-f99f-45e3-ac79-11f1b838cbfa	Finance [DEV]	e3b14201-f923-4bd6-b2de-58a6c40fca0b	Pricing
62066cd5-f99f-45e3-ac79-11f1b838cbfa	Finance [DEV]	c01aaaa-780d-4272-b15f-451de764c700	Costs
5d776429-dd57-4767-ae32-c8fc5f777f23e	Finance [TST]	b8c5e907-d1bf-4261-b6d0-338885cb97b6	Costs
5d776429-dd57-4767-ae32-c8fc5f777f23e	Finance [TST]	5fe45052-2c99-47b2-aa07-63dfcebb8123c	Pricing



Retrieve Refresh Schedules.

- We have to call [this API](#) to retrieve the refresh schedules of the datasets

Learn / Power BI REST APIs / Datasets /⊕ ⋮

Datasets - Get Refresh Schedule In Group

Reference👍 Feedback

Service: Power BI REST APIs
API Version: v1.0

Returns the refresh schedule for the specified dataset from the specified workspace.

Permissions

This API call can be called by a service principal profile. For more information see: [Service principal profiles in Power BI Embedded](#).

Required Scope

Dataset.ReadWrite.All or Dataset.Read.All

HTTP

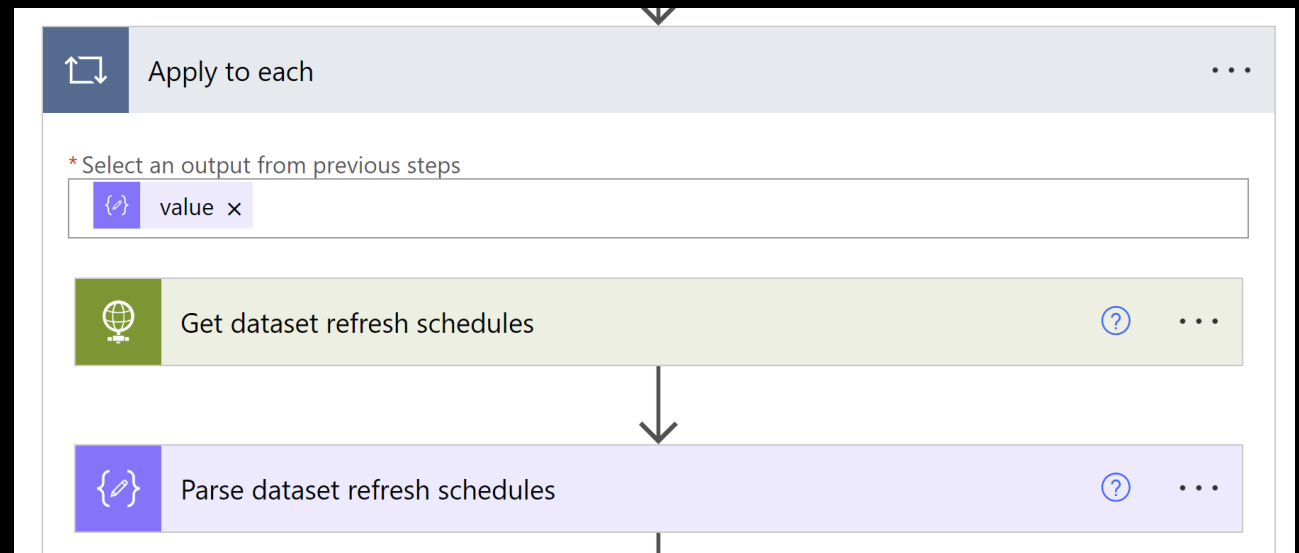
Copy

Try It

```
GET https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshSchedule
```

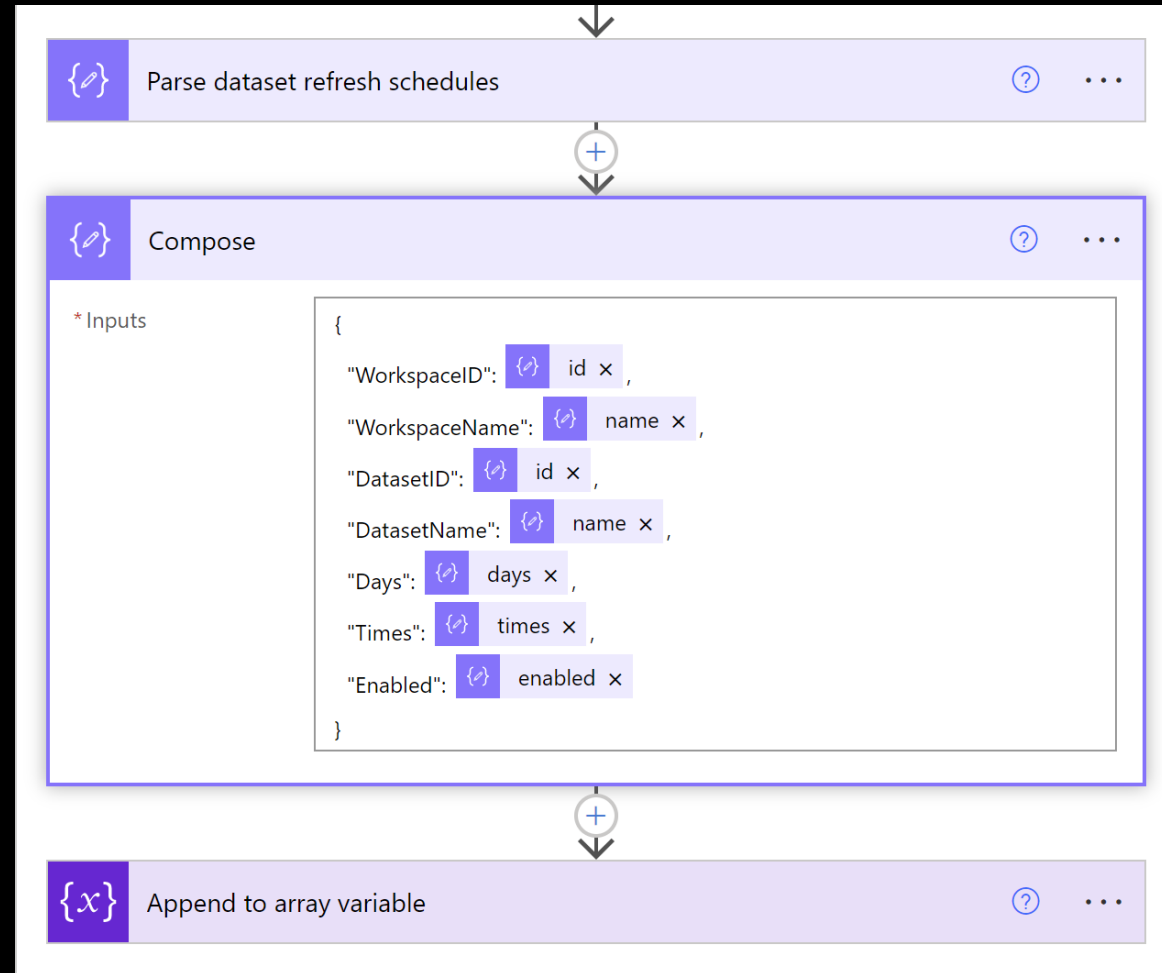
➤ Retrieve Refresh Schedules.

- 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

The screenshot displays two steps in a Power BI Desktop sequence, connected by a downward arrow with a plus sign.

Step 1: Create CSV table

- From:** {x} results x
- Show advanced options:** ▾

Step 2: Create file

- * Site Address:** Power BI Monitor Solutions - ns ▾
- * Folder Path:** /Shared Documents/General ▾
- * File Name:** GetDatasetsRefreshSchedules.csv
- * File Content:** {v} Output x



Expand Power BI Report.

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

Improvements 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?

We'd  to hear your feedback!



SCAN ME

FEEDBACK



THANK YOU!



POWER PLATFORM
BOOTCAMP



u2u



Microsoft



dataMinds

