

# Get started with your own Power BI Monitor!

Paulien van Eijk

11-03-2023

De Power BI Gebruikersdagen worden mede mogelijk gemaakt door:



PLATINUM

delaware



ilionx



GOUD



Voorblijven. Niet bijblijven.

KASPAROV  
FINANCE & BI

macaw



GET  
RESPONSIVE

ZILVER

VICTA  
BUSINESS INTELLIGENCE



Motion10  
AN HSO COMPANY



VALID  
STAY AHEAD



valcon

iqb

Quanto  
collective analytics

COMMUNITY



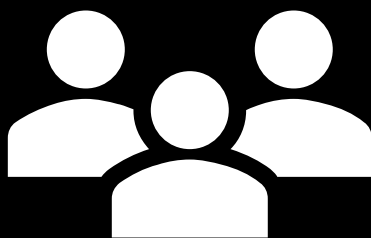
DashData.

volda;  
INFORMATIESPECIALISTEN

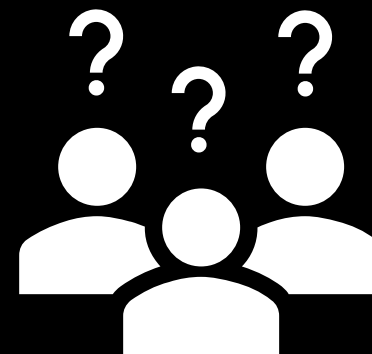
Power BI  
Connector by DAVISTA

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

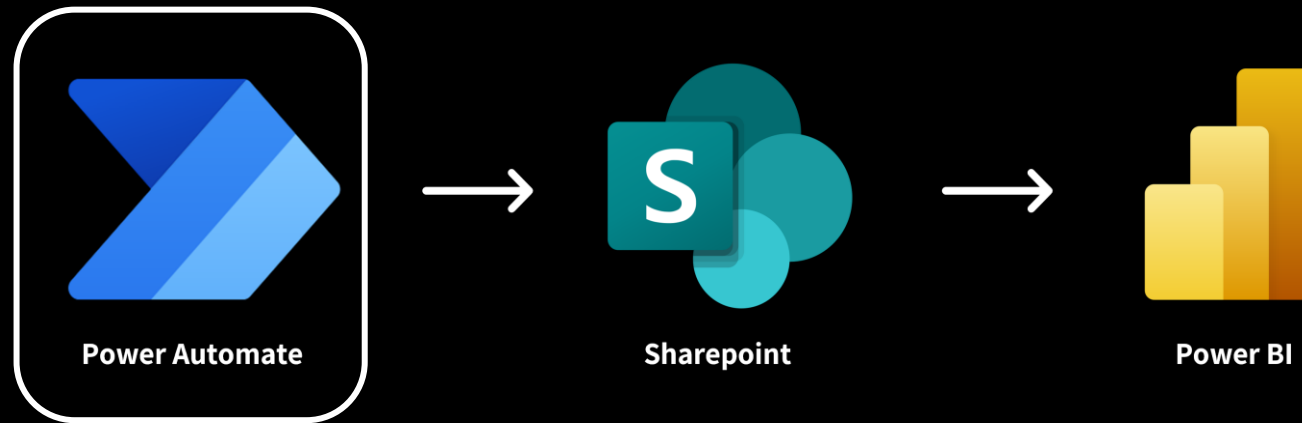


Sharepoint

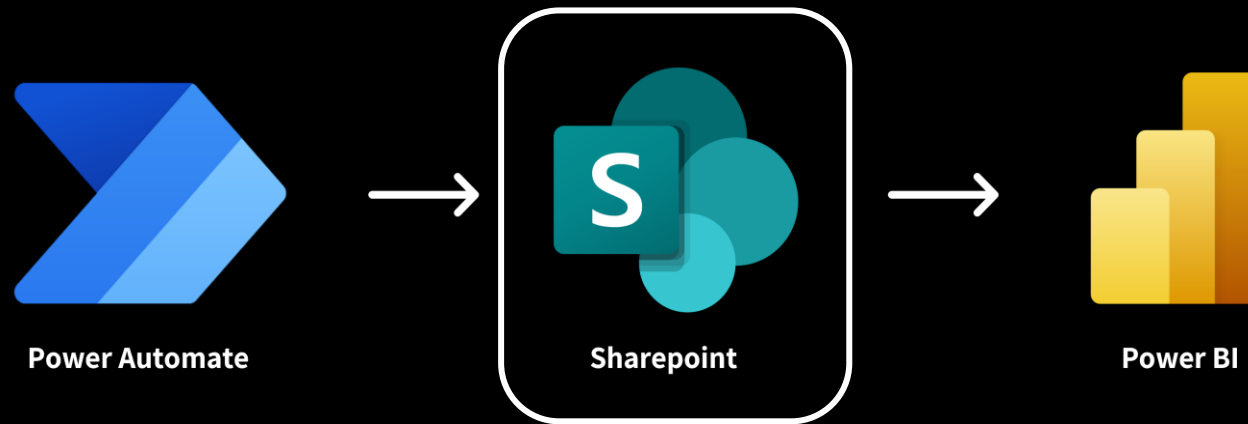


Power BI

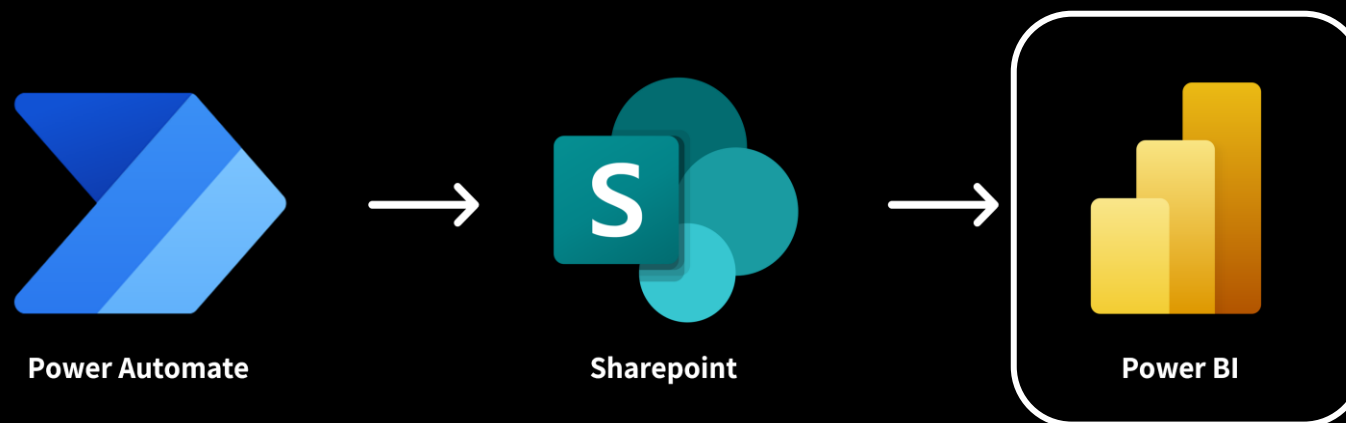




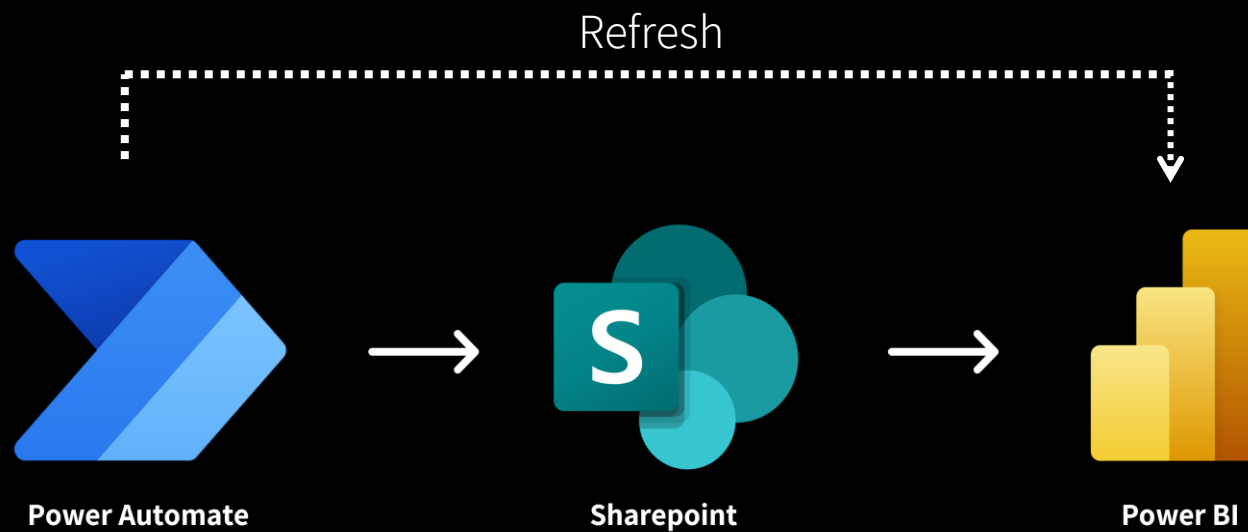
- Call Power BI REST API
- Store results



- 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

The screenshot shows the Microsoft Learn documentation page for the 'Groups - Get Groups' API endpoint. The page is dark-themed with a sidebar on the left containing a navigation menu. The main content area includes a breadcrumb trail, a title, a reference section, a description, a permissions section, a required scope section, and a code block for the HTTP request. A right-hand sidebar lists 'In this article' topics.

**Learn / Power BI REST APIs / Groups /**

## Groups - Get Groups

Reference [Feedback](#)

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

Returns a list of workspaces the user has access to.

When user permissions to a workspace have been recently updated, the new permissions might not be immediately available through API calls. To refresh user permissions, use the [Refresh User Permissions](#) API call.

### 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

Workspace.Read.All or Workspace.ReadWrite.All

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

```
GET https://api.powerbi.com/v1.0/myorg/groups
```

**In this article**

- Permissions
- Required Scope
- URI Parameters
- Responses

[Show more](#) ▾

[Download PDF](#)

# ▶ 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 displays the 'HTTP' tab in a request builder tool. The configuration is as follows:

- \* Method:** GET
- \* URI:** Enter request URL. A red error message below the field states: **'URI' is required.**
- Headers:** A table with two columns: 'Enter key' and 'Enter value'. A plus icon is on the right.
- Queries:** A table with two columns: 'Enter key' and 'Enter value'. A plus icon is on the right.
- Body:** Enter request content
- Cookie:** Enter HTTP cookie
- \* Authentication:** Active Directory OAuth
- Authority:** Default: https://login.windows.net/
- \* Tenant:** Enter tenant. A red error message below the field states: **'Tenant' is required.**
- \* Audience:** Enter audience. A red error message below the field states: **'Audience' is required.**
- \* Client ID:** Enter client ID. A red error message below the field states: **'Client ID' is required.**
- \* Credential Type:** Secret
- \* Secret:** Enter secret as plain text or use a secure parameter. A red error message below the field states: **'Secret' is required.**

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

# & **Create and assign service principal.**

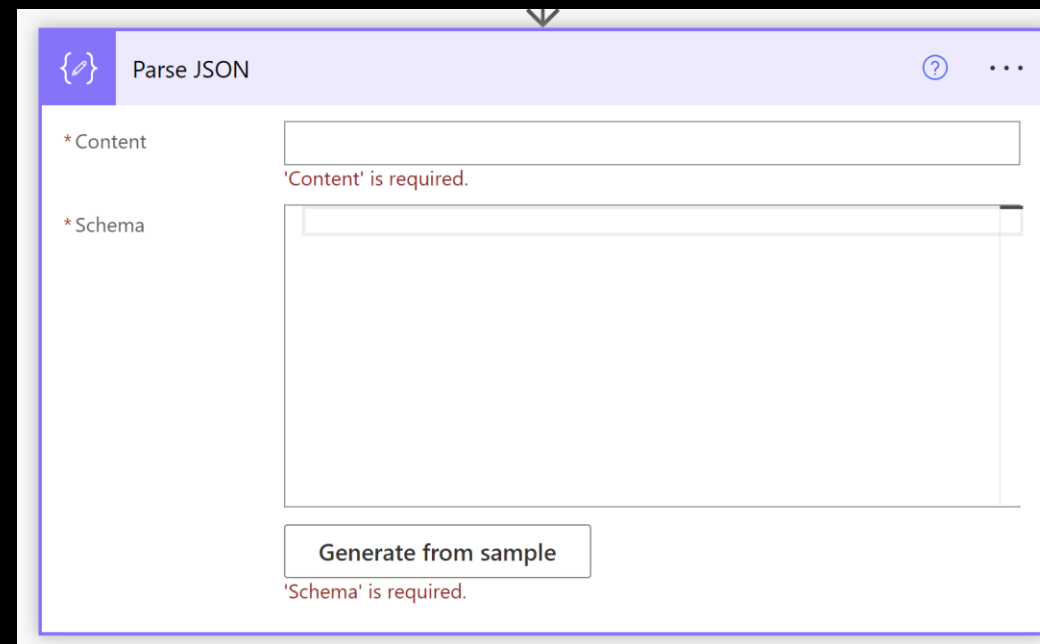
- Create Azure App Registration
- Grant API Permissions
  1. Add a permission
  2. Power BI Service
  3. Delegated Permissions
  4. Workspace.Read.All
  5. Grant admin consent for MST
  6. Add service principal to workspace

## 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

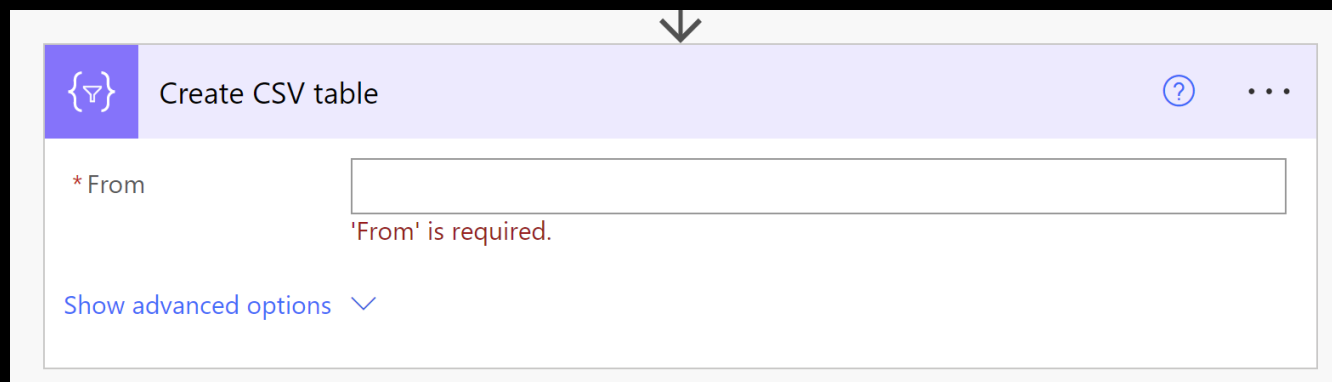


The screenshot shows a web application titled "Parse JSON". It has a sidebar on the left with two main sections: "\* Content" and "\* Schema". The "\* Content" section has a text input field, and the "\* Schema" section has a larger text area. Both fields have red error messages below them: "'Content' is required." and "'Schema' is required." respectively. At the bottom of the form 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

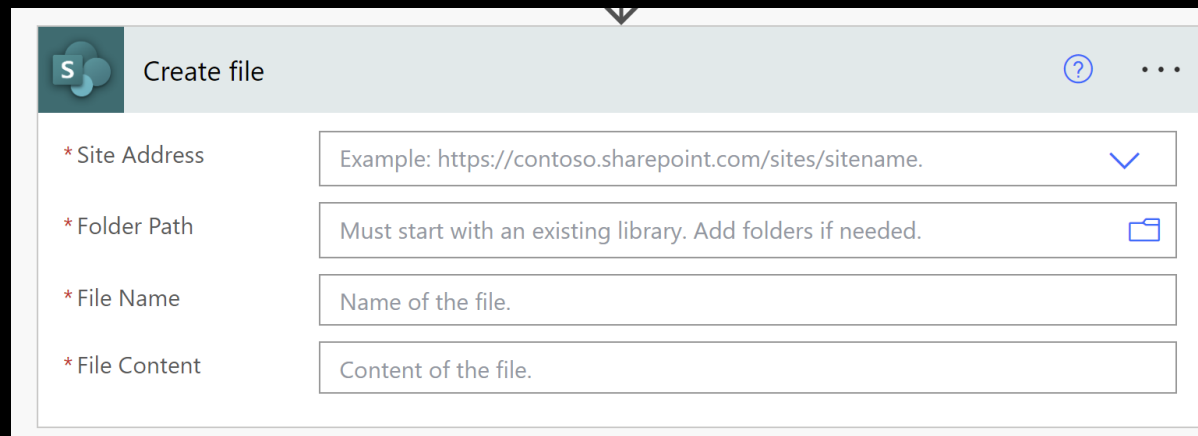


The screenshot shows a web form titled "Create CSV table". It features a purple header bar with a table icon, the title, a help icon, and a menu icon. Below the header, there is a label "\*From" followed by an empty text input field. A red error message "'From' is required." is displayed below the input field. At the bottom of the form, there is a link "Show advanced options" with a downward-pointing chevron icon. A black arrow points down to the top of the form.



## Create file.

- Creates file on given sharepoint location with the specified contents
- **Site Address:** Siteaddress
- **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 dropdown menu showing 'Example: https://contoso.sharepoint.com/sites/sitename.' and a blue checkmark icon. The second field is 'Folder Path' with a text input containing 'Must start with an existing library. Add folders if needed.' and a blue folder icon. The third field is 'File Name' with a text input containing 'Name of the file.'. The fourth field is 'File Content' with a text input containing 'Content of the file.'. There are also help and more options icons in the top right corner of the dialog.

Field	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:** Siteadress

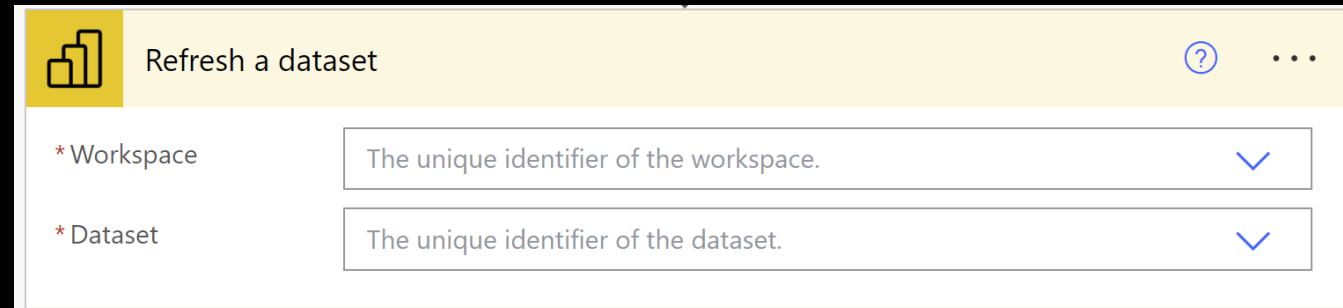


# **Publish report to Power BI service.**

- 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 an HTTP request in a workflow tool. At the top, a 'For each workspace' loop is established, with a 'value' output from a previous step selected. The main configuration is for the 'Get all datasets' action, which is set to use the GET method. The URI is configured as `https://api.powerbi.com/v1.0/myorg/groups/{id}/datasets`, where '{id}' is a dynamic content placeholder. The authentication is set to 'Active Directory OAuth' with the following details: Authority is `https://login.windows.net/`, Tenant is `871b2406-9092-4b60-b275-770640f94111`, Audience is `https://analysis.windows.net/powerbi/api`, Client ID is `a5b006a-2330-44a3-976d-21a0b38b412b`, Credential Type is 'Secret', and the Secret is `f1c0c3-8b6a1m1447-f71q8g4z00b6-go-8enM4ids`. A 'Hide advanced options' link is visible at the bottom.

For each workspace

\* Select an output from previous steps

{value} x

Get all datasets

\* Method: GET

\* URI: `https://api.powerbi.com/v1.0/myorg/groups/{id}/datasets`

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: `871b2406-9092-4b60-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`

Hide advanced options

# Parse Json.

- Parse results again (with example)

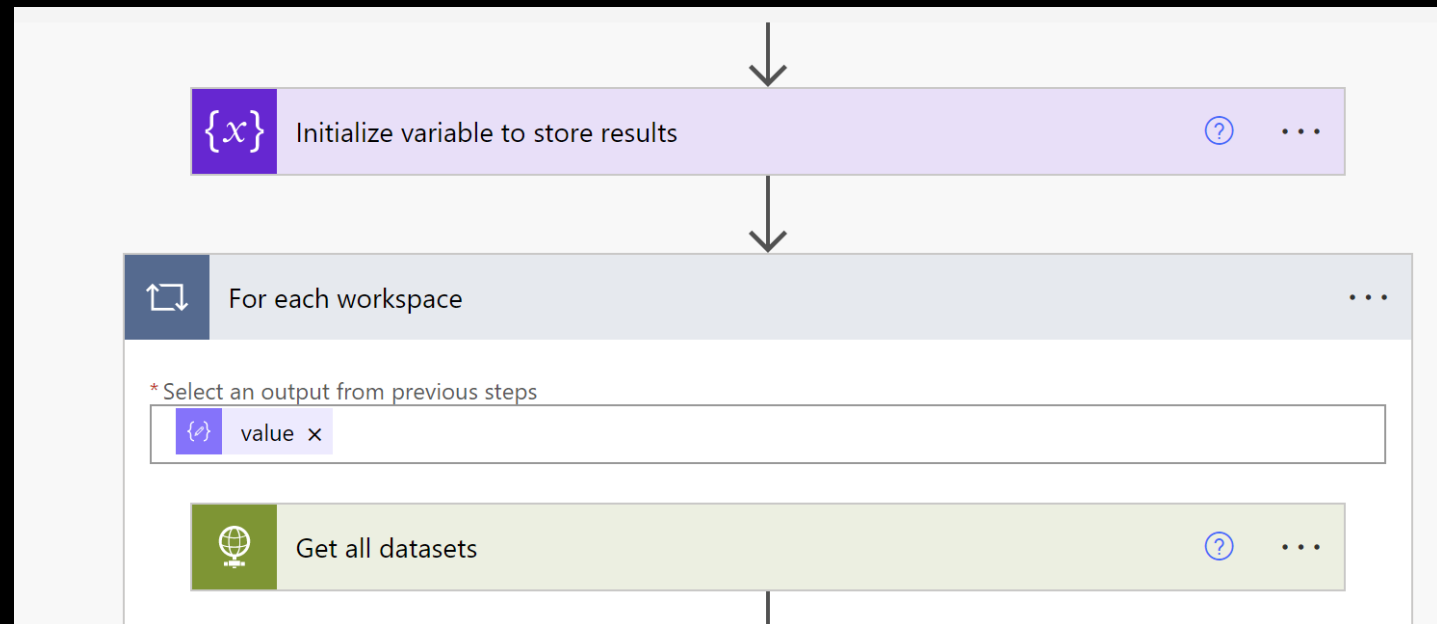
The screenshot shows a web interface with two main sections. The top section, titled "Get all datasets", has a green header bar with a globe icon, a question mark icon, and three dots. An arrow points down from this section to the "Parse dataset results" section. The "Parse dataset results" section has a purple header bar with a curly brace icon, a question mark icon, and three dots. Below the header, there are two tabs: "Content" and "Schema". The "Content" tab is active, showing a text input field with the placeholder "Body x". The "Schema" tab is also visible, showing a JSON schema definition. Below the schema definition, there is a button labeled "Generate from sample".

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

Generate from sample

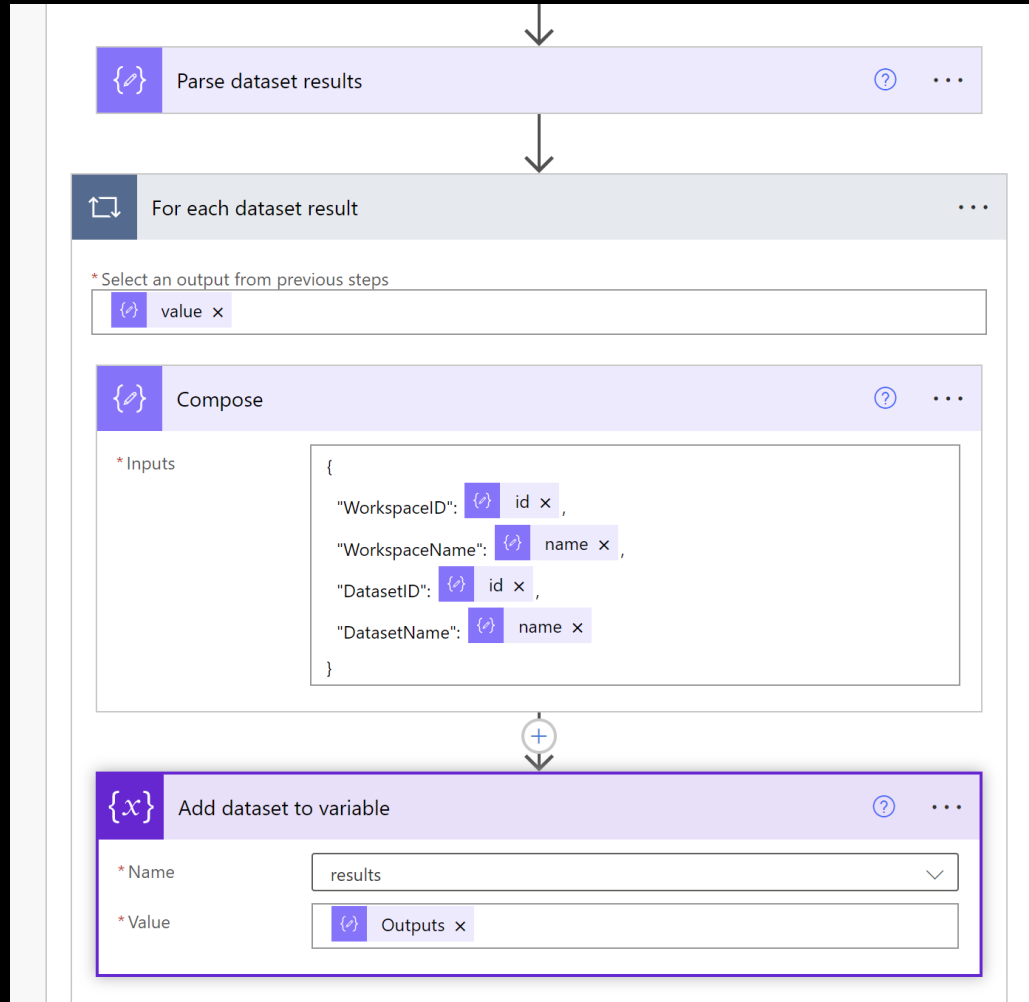
## ➤ Initialize variable.

- 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 shows a Power Automate workflow with two steps connected by a downward arrow. The first step is 'Create CSV table', which has a 'From' field set to a variable named 'results'. The second step is 'Create file', which has four required fields: 'Site Address' (empty), 'Folder Path' (set to '/Shared Documents/General'), 'File Name' (set to 'GetDatasets.csv'), and 'File Content' (set to an 'Output' variable).

**Create CSV table**

\* From: {x} results x

Show advanced options ▾

**Create file**

\* Site Address: [Empty field]

\* 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

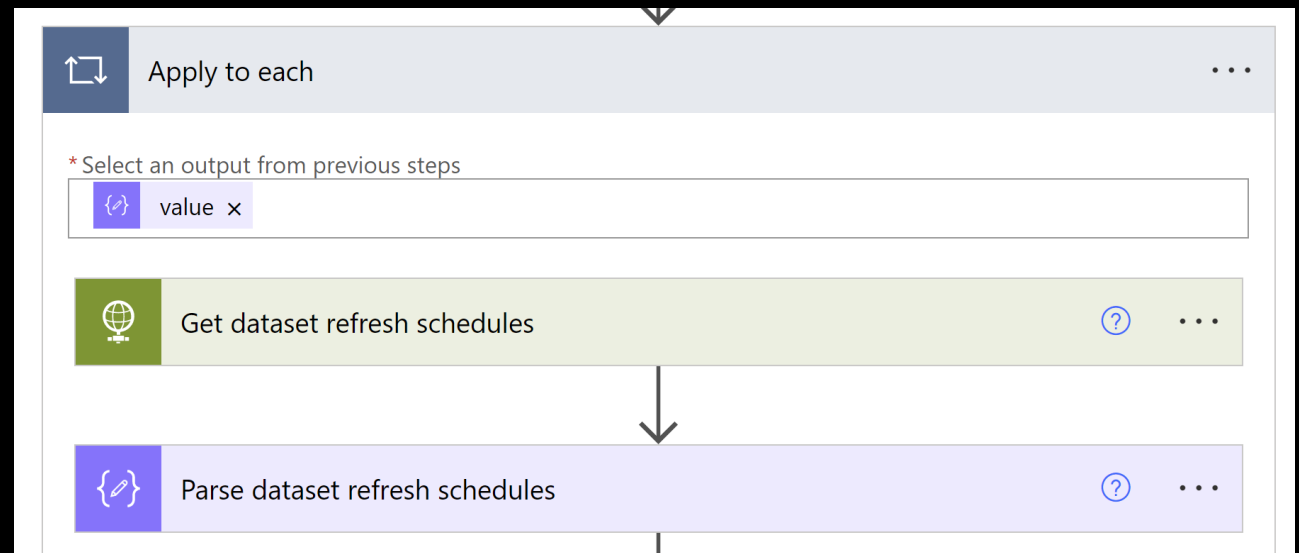
[Copy](#)

[Try It](#)

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

## ➤ Retrieve Refresh Schedules.

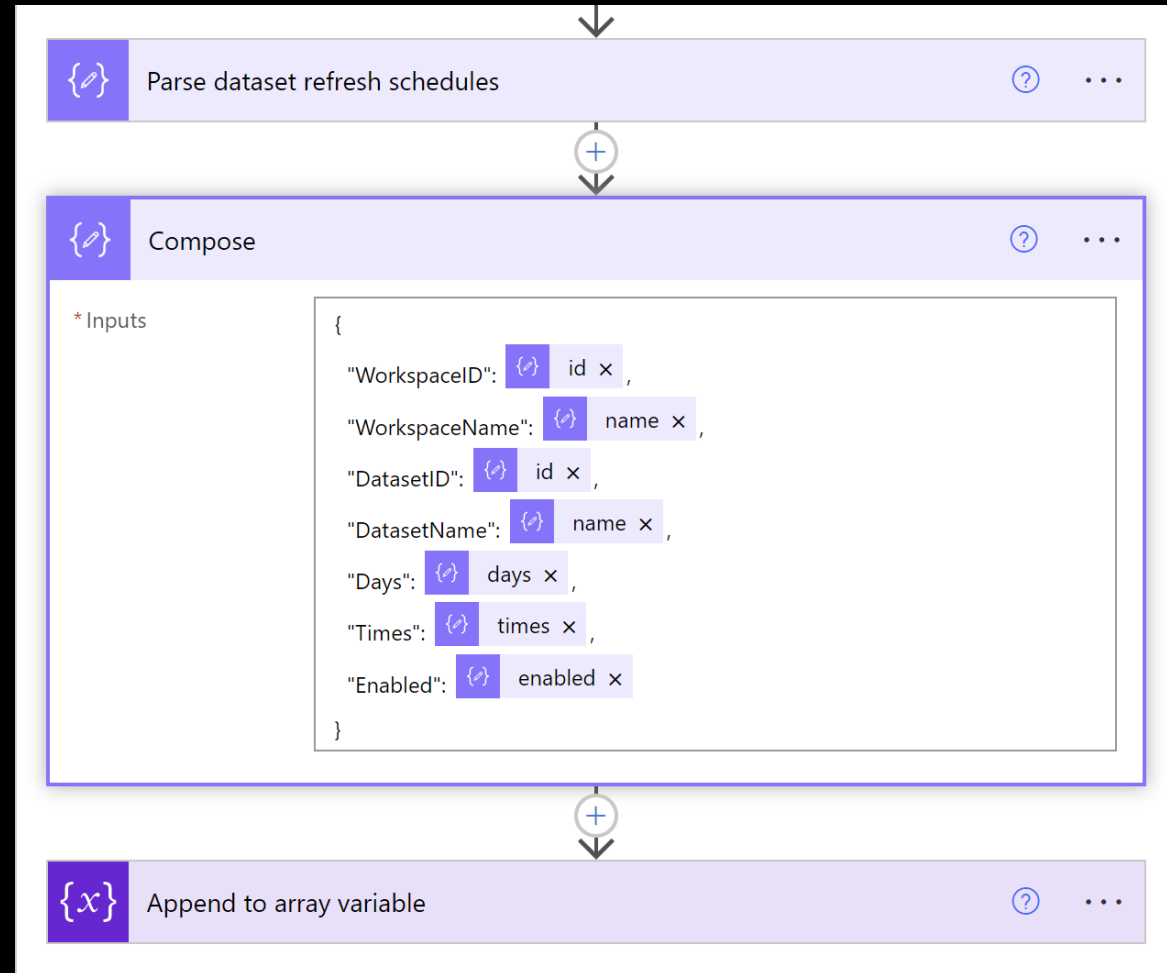
- Retrieve all refresh schedules per dataset with an HTTP request
- Parse results





## ➤ 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 a Power Automate workflow with two actions connected by a connector. The first action, 'Create CSV table', has its 'From' property set to a variable named 'results'. The second action, 'Create file', is configured with the following properties: 'Site Address' is empty, 'Folder Path' is '/Shared Documents/General', 'File Name' is 'GetDatasetsRefreshSchedules.csv', and 'File Content' is set to a variable named 'Output'.

**Create CSV table**

\* From: {x} results x

Show advanced options ▾

**Create file**

\* Site Address: [Empty field]

\* 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?