MICRAN DATA

Power Automate & PBI Datasets

Date: 31.10.2023

OUR AGENDA

The usage of Power Automate in combination with the Power BI dataset, can open up the endless possibilities including personalized notifications and alerts.

This will be a 2-part series, starting from a basic examples and then having a dynamic flow.

01

Basic Data Extraction

Learn how to generate the code directly using Power BI Desktop.

02

Power Automate Flow

See the possible flows that we can create including their respective triggers.

03

Customized CSV

Learn how to customize the column names and avoid using the default generated ones.

04

Combining Queries

This aims at using a query result, as the input for a second query.

Flow Order

This would be the order in which it is recommended to create the flow.

Prepare Queries

This is done in Power BI Desktop and will require a bit of tweaking in the code.

Automate Flow Creation

It is important to think about the trigger type and possible variables to be used.

(Customize) - Optional

In the case of tables, the column headers can be customized.

Share

This can be done either by creating a file, sending out an e-mail, or through Teams.

Key Limitations

MICRAN DATA

Run Query

This is limited to 100k rows, or 1,000,000 values. A possible workaround would be to paginate the extraction.

Monitor the capacity, if the Power BI workspace is backed by a Premium capacity.

Consider the Paginated Reports as well as a possible customized solution.

File Creation

There can be Power Automate errors when saving a file with the same name on a SharePoint site.

A change in the dataset structure can affect as well the Power Automate flow.

If multiple files get created, an array will be used to append all the data together.

Sharing

When sending a file, keep in mind the possible future growth in size. Most tenants also have an attachment size policy.

Files cannot be shared in an individual Teams chat. Alternative would be to save them on a Teams channel (with the help of SharePoint) and get the hyperlink to the file.

Table Extract Query

MICRAN DATA

DEFINE

```
VAR __DS0Core =
    SUMMARIZECOLUMNS(
        'Bookings'[BookingDate],
        'Bookings'[Customer],
        "CountBookingID",
                CALCULATE(COUNTA('Bookings'[BookingID]))
            )
```

This step will summarize the data and is required.

```
VAR __DS0PrimaryWindowed =
    TOPN(
        501,
        __DS0Core,
        'Bookings'[BookingDate],
            1,
        'Bookings'[Customer],
            1
        )
```

This step will be completely removed, as it is automatically generated by Power BI to only load chunks of 500 rows.

EVALUATE

```
    __DS0PrimaryWindowed
```

This step will mention which step will be evaluated

ORDER BY

```
'Bookings'[BookingDate],
'Bookings'[Customer]
```

This is an optional step, which can be kept or removed.

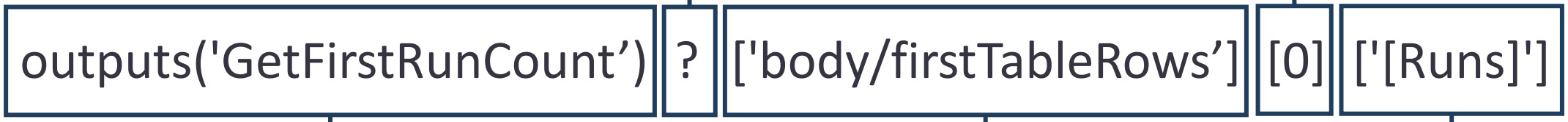
```
DEFINE
    VAR __DS0Core =
        SUMMARIZECOLUMNS(
            'Bookings'[BookingDate],
            'Bookings'[Customer],
            "CountBookingID",
                CALCULATE(COUNTA('Bookings'[BookingID]))
                )

EVALUATE
        __DS0Core

ORDER BY
'Bookings'[BookingDate],
'Bookings'[Customer]
```

Single Value Extraction

MICRAN DATA

- This is called the safe navigation operator.
- It's used to safely navigate through potentially null objects or arrays.
- If the object before the ? is null, the expression will short-circuit and return null, instead of throwing an error.

- This provides the position index that will be extracted.
- It follows a zero-based indexing system, hence it's grabbing the first item.



- This is retrieving the output of an action in Power Automate named "GetFirstRunCount".
- Each action can produce outputs that can be consumed by subsequent actions.
- The **outputs** function allows you to access these outputs.

- This is accessing a property in the output.
- Specifically, it's looking for a property named firstTableRows inside the body of output.
- The use of the forward slash / is a shorthand way to navigate through nested properties.

- This is accessing a property named [Runs] from the object that was retrieved in the previous step
- The inner brackets are coming from the source name

Why Compose Steps

MICRAN DATA

Simplification & Transformation

If the JSON output from the previous step is complex and you only need a subset or a transformed version of that data for subsequent steps, a Compose action can help you reshape or filter the data to just what you need.

Debugging & Logging

If you're troubleshooting your flow, having a Compose step can serve as a checkpoint where you can see exactly what data is being passed at that point in the flow. This can be especially useful when working with dynamic content that might change over time.

Static Content

Compose can be used to create static content which can be combined with dynamic content from previous steps. This can be useful for creating messages, emails, or any other content that needs a mix of static and dynamic values.

Readability & Maintenance

By using a Compose action to store intermediate results, you can make your flow more readable. This can be particularly useful if other team members need to understand or maintain the flow. It clearly delineates where transformations or data extractions occur.

Performance

In some cases, if you're repeatedly using the same part of the JSON output in multiple subsequent actions, using a Compose action might reduce the need to repeatedly parse the JSON. This could potentially lead to minor performance improvements.

Reusability

In some cases, if you're repeatedly using the same part of the JSON output in multiple subsequent actions, using a Compose action might reduce the need to repeatedly parse the JSON. This could potentially lead to minor performance improvements.

Why Variables vs. Compose

MICRAN DATA

Category	Variables	Compose
Purpose	Variables are designed to store and manage values that you might want to change or use multiple times throughout the flow.	The Compose action is designed to hold a value or expression that you want to use later. It's more like a snapshot of data at that specific point in the flow.
Actions	• You'd typically use the "Initialize variable" action to declare a variable. • You'd use the "Set variable" action to change its value. • If you want to increment an integer, you'd use the "Increment variable" action.	• You just have the "Compose" action where you set its value. Once set, it cannot be changed like a variable.
Scope	Once a variable is initialized, it can be accessed anywhere in the flow after its initialization point.	The result of a Compose action can be accessed in subsequent actions after the Compose action.
Flexibility	You can change the value of a variable multiple times in a flow using the "Set variable" action.	The result of a Compose action can be accessed in subsequent actions after the Compose action.
Use Case	**=> Multiple Uses, Potential Update**	**=> Single Use, No Modification**

THANK YOU