



Power BI API Extravaganza

2021-11



Rui Romano

Power BI CAT

Microsoft

 @RuiRomano

 <https://www.linkedin.com/in/ruiromano/>

 <https://ruiromanoblog.wordpress.com>



Lives in Porto, Portugal

♡ Power BI & Data

+15 years working on Microsoft BI & Data Stack



Slides here:

<https://github.com/RuiRomano/sessionslides>

[Recorded Session](#)

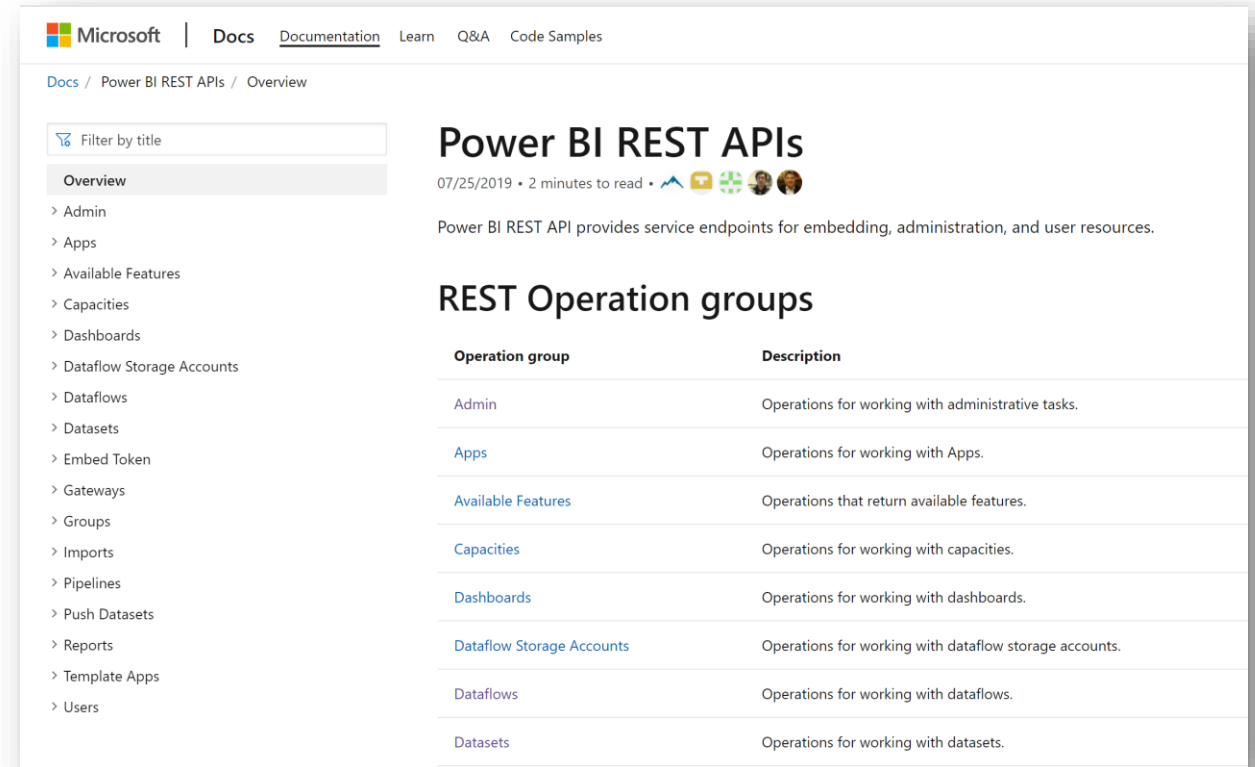
Power BI API Extravaganza

- Not a Power BI Intro!
- 100% focus on the REST API's
- Fast Paced 😊
Focus on the possibilities not details...

Power BI REST API's ?

Collection of REST API's for:

- Administration
- Monitoring
- Productivity & Automation
- Real-Time
- Embedding



The screenshot shows the Microsoft Docs page for Power BI REST APIs. The page has a navigation bar with links to Docs, Documentation, Learn, Q&A, and Code Samples. Below the navigation bar, there is a breadcrumb trail: Docs / Power BI REST APIs / Overview. A search bar is present with the text "Filter by title". On the left side, there is a table of contents with links to various sections: Overview, Admin, Apps, Available Features, Capacities, Dashboards, Dataflow Storage Accounts, Dataflows, Datasets, Embed Token, Gateways, Groups, Imports, Pipelines, Push Datasets, Reports, Template Apps, and Users. The main content area is titled "Power BI REST APIs" and includes a sub-header "REST Operation groups". Below this, there is a table with two columns: "Operation group" and "Description". The table lists various operation groups and their descriptions.

Operation group	Description
Admin	Operations for working with administrative tasks.
Apps	Operations for working with Apps.
Available Features	Operations that return available features.
Capacities	Operations for working with capacities.
Dashboards	Operations for working with dashboards.
Dataflow Storage Accounts	Operations for working with dataflow storage accounts.
Dataflows	Operations for working with dataflows.
Datasets	Operations for working with datasets.

[Power BI REST APIs - Power BI REST API | Microsoft Docs](#)

How can I “play” with the API’s?

- Go to the [Docs](#) and “Try it” with your Power BI tenant data

Contents

Groups - Get Groups

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

Returns a list of workspaces the user has access to.

Note: Users that have been recently added to a group may not have their new group immediately available, see [Refresh user permissions](#).

Required scope: Workspace.Read.All or Workspace.ReadWrite.All
To set the permissions scope, see [Register an app](#).

HTTP

Copy

Try It

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

With optional parameters:

HTTP

Copy

GET [https://api.powerbi.com/v1.0/myorg/groups?\\$filter={filter}&\\$top={\\$top}&\\$skip={\\$skip}](https://api.powerbi.com/v1.0/myorg/groups?$filter={filter}&$top={$top}&$skip={$skip})

URI Parameters

Name	In	Required	Type	Description
\$filter	query		string	Filters the results, based on a boolean condition
\$skip	query		integer int32	Skips the first n results

requestid: 983f107e-a730-498f-b342-a5a210de322e

Body

JSON

Copy

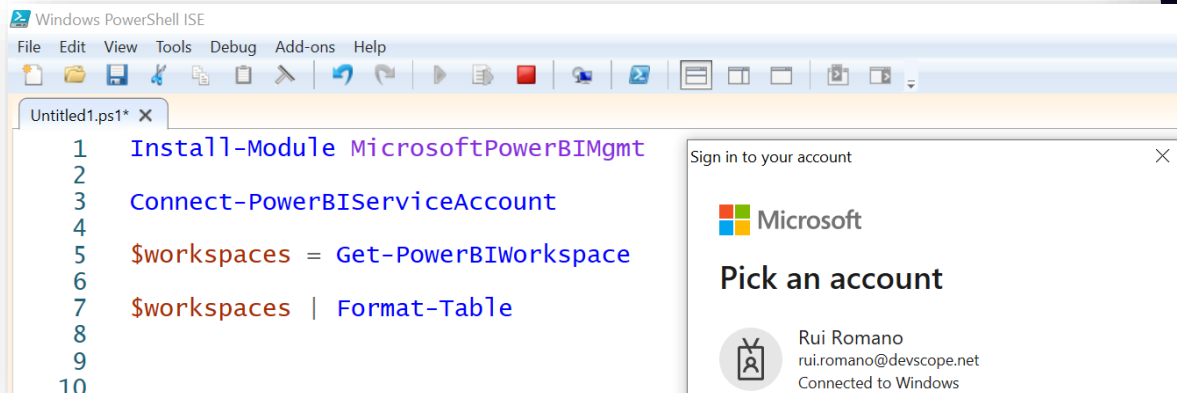
```
{
  "@odata.context": "http://wabi-west-europe-redirect.analysis.windows.net/v1.0/myorg/$metadata#groups",
  "@odata.count": 110,
  "value": [
    {
      "id": "43504e54-e44d-4a5a-bc2f-d3246fc5568a",
      "isReadOnly": false,
      "isOnDedicatedCapacity": false,
      "type": "Group",
      "name": "pbtiptstricksdata"
    },
    {
      "id": "29d49eaf-4016-4f67-91f1-7ba111cdf086",
      "isReadOnly": false,
      "isOnDedicatedCapacity": false,
      "type": "Group",
      "name": "DVS Sharepoint Play"
    },
    {
      "id": "a36d64a4-9cf7-438a-92af-87db0b54e562",
      "isReadOnly": false,
      "isOnDedicatedCapacity": false,
      "type": "Group",
      "name": "Lisbon Data8ricks"
    },
    {
      "id": "e4790749-58d6-4608-855a-c9124bd9eab7",
      "isReadOnly": false,
      "isOnDedicatedCapacity": false,
      "type": "Group",
      "name": "PBI Embed"
    },
    {
      "id": "ca6dc9f6-62eb-43fb-9244-4c2ab87d3803",
      "isReadOnly": false,
      "isOnDedicatedCapacity": false,
      "type": "Group",
      "name": "WMI Sales Home"
    }
  ]
}
```

How can I work with the API's?

- You need to be comfortable with code/script/automation (power automate)
Any language that can make an HTTP request is compatible: .Net, PowerShell, Python,...
- There are multiple libraries/packages available:
PowerShell – [MicrosoftPowerBIMgmt](#), [PowerBIPS](#) (♥)
Python – [python-power-bi](#)
.Net – [Microsoft.PowerBI.Api](#)
- If possible, use a Service Principal

Demo

MicrosoftPowerBIMgmt



The screenshot shows the Windows PowerShell ISE interface. The main window displays a PowerShell script with the following commands:

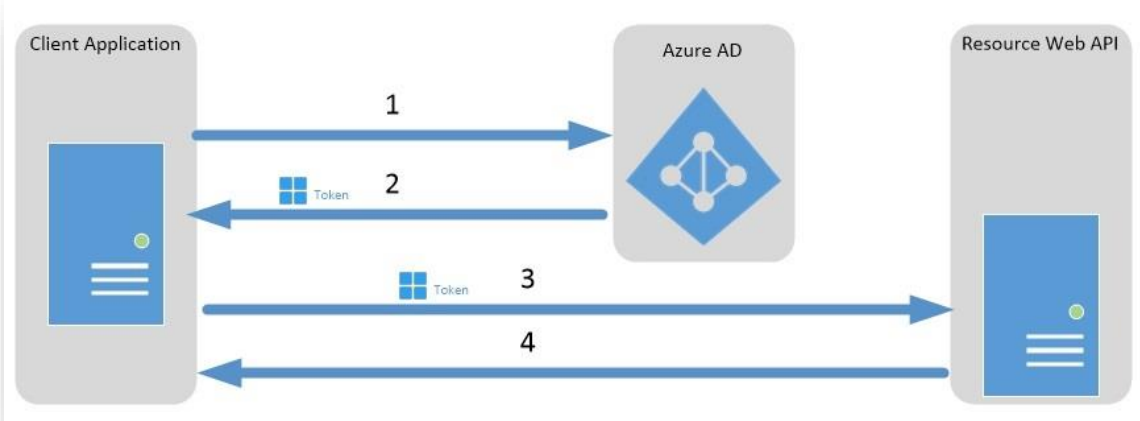
```
1 Install-Module MicrosoftPowerBIMgmt
2
3 Connect-PowerBIServiceAccount
4
5 $workspaces = Get-PowerBIWorkspace
6
7 $workspaces | Format-Table
8
9
10
```

On the right side, a Microsoft account sign-in dialog is open, titled "Sign in to your account". It displays the Microsoft logo and the text "Pick an account". Below this, a user profile is shown with a circular icon, the name "Rui Romano", the email address "rui.romano@devscope.net", and the status "Connected to Windows".

```
mirror_mod = modifier_ob.  
Set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES --  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

API Authentication

- OAuth 2.0
- Authentication Flows:



Authentication Flow	Interactive	Requirements
Client Credentials (Recommended for non interactive)	No	Azure AD Service Principal
Auth Code	Yes	Power BI Account

```
POST /token HTTP/1.1
Host: authorization-server.com

grant_type=client_credentials
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```



```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVh",
  "scope": "create"
}
```


Service Principal – Manual Step by Step

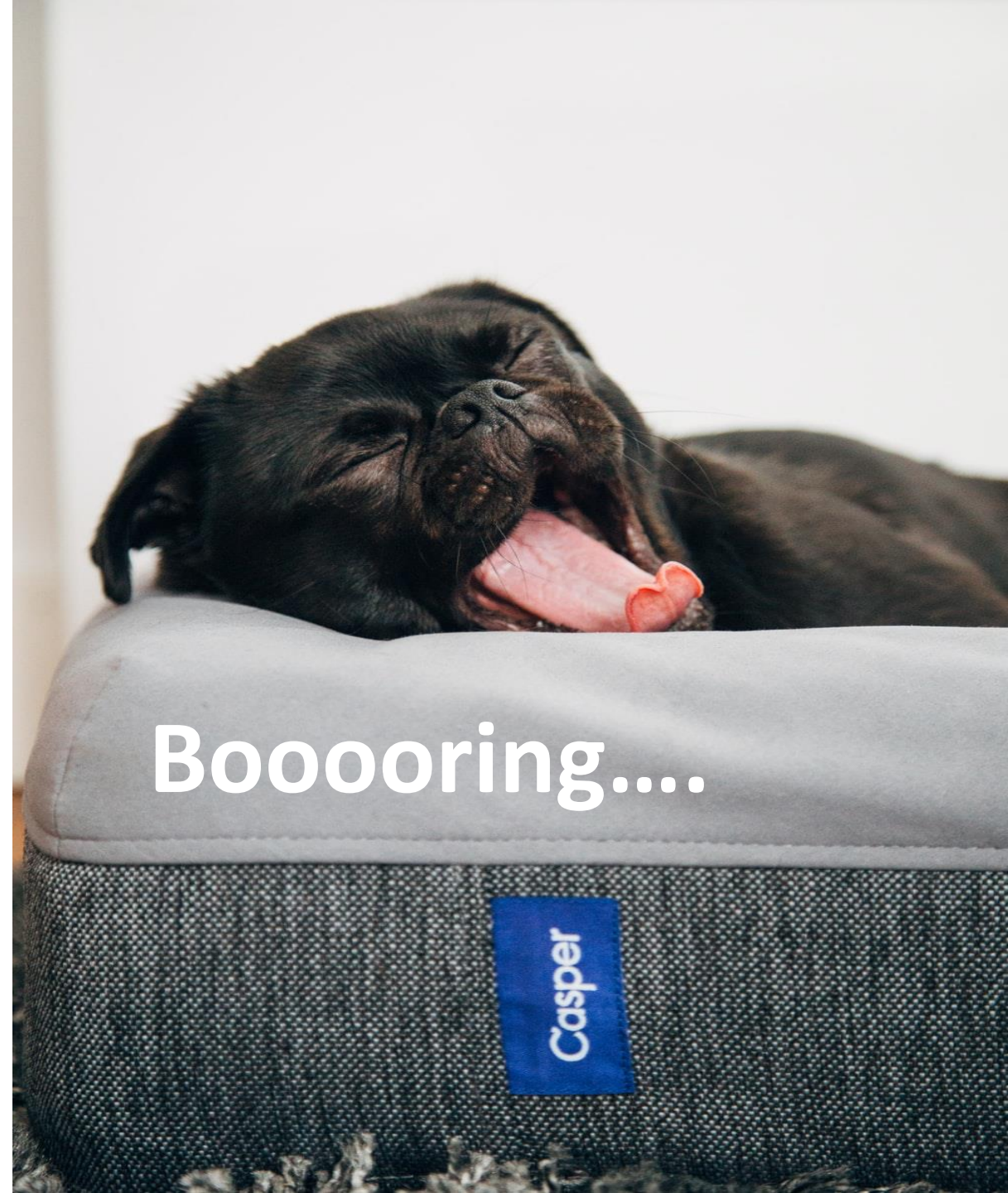
1. Go to [Azure AD Active Directory](#)
2. Go to [App Registrations](#) and create a new App (leave defaults)
3. Generate a new App Secret
4. Save the App Id, App Secret & Tenant Id
5. [Create/Reuse an Azure AD Security Group](#)
6. Add the Service Principal to the Security Group as a member
7. Authorize the Security Group in Power BI Admin Portal Tenant Settings:
 - Allow service principals to use Power BI APIs
 - Allow service principals to use read-only Power BI admin APIs – if admin calls are necessary (ex: monitoring)
 - Allow XMLA endpoints and Analyze in Excel with on-premises datasets (execute Queries)

Note: You don't need to add any Power BI API Permissions

More info: [link](#)

Scenarios / Demos

- DAX Queries w/ REST API
- Monitoring
- Productivity / DevOps
- Realtime



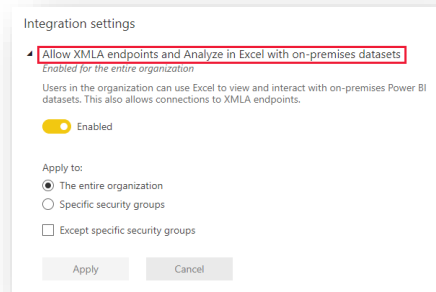
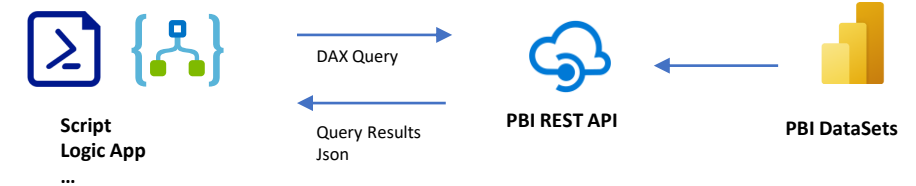
Power BI Execute Queries API

```
mirror_mod = modifier_ob.  
Set mirror object to mirror  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES --  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

REST API - Dataset Execute Queries

- The coolest API! 😊
- Useful Scenarios:
 - Data Integration (*limited)
 - Alerting System using Power BI Data
 - App/Website integrations
- Works on Premium & **Shared (PRO)**
- Requirements
 - “Build” permission on the dataset
 - Enable Tenant Setting “Allow XMLA Endpoints and Analyze in Excel”
- Main Limitations
 - Result sets capped to 100k rows
 - No parallel queries

Script: [here](#)



```
Dataset-REST API DAX Query.ps1 > ...
$Query = "EVALUATE"

VAR p_currentDate = dt"$($date.ToString("yyyy-MM-dd"))"

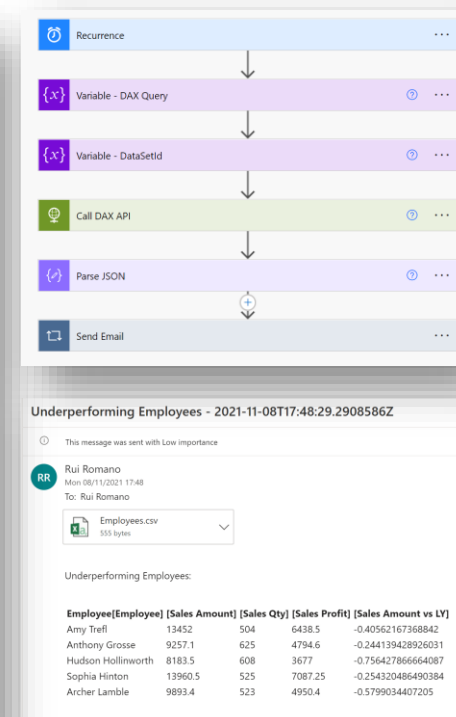
return
FILTER(
    SUMMARIZECOLUMNS(
        'Employee'[Employee],
        TREATAS({p_currentDate}, 'Calendar'[Date]),
        ""Sales Amount"", [Sales Amount],
        ""Sales Qty"", [Sales Qty],
        ""Sales Profit"", [Sales Profit],
        ""Sales Amount vs LY"", [% Sales Amount vs ly]
    ),
    [Sales Amount vs LY] < 0
)

Connect-PowerBIServiceAccount

$body = @{
    "queries" = @(
        @{
            "query" = $Query
        }
    )
    ; "includeNulls" = $false
}

$bodyStr = $body | ConvertTo-Json

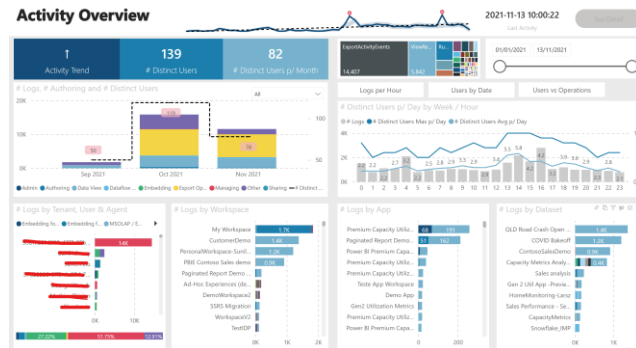
$result = Invoke-PowerBIRestMethod -url "datasets/$datasetId/executeQueries" -body $bodyStr
```



Power BI Monitoring

Full session on the topic
[link](#)

Code & Slides
[link](#)



```
mirror_mod = modifier_ob.  
set mirror object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

OPERATOR CLASSES

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

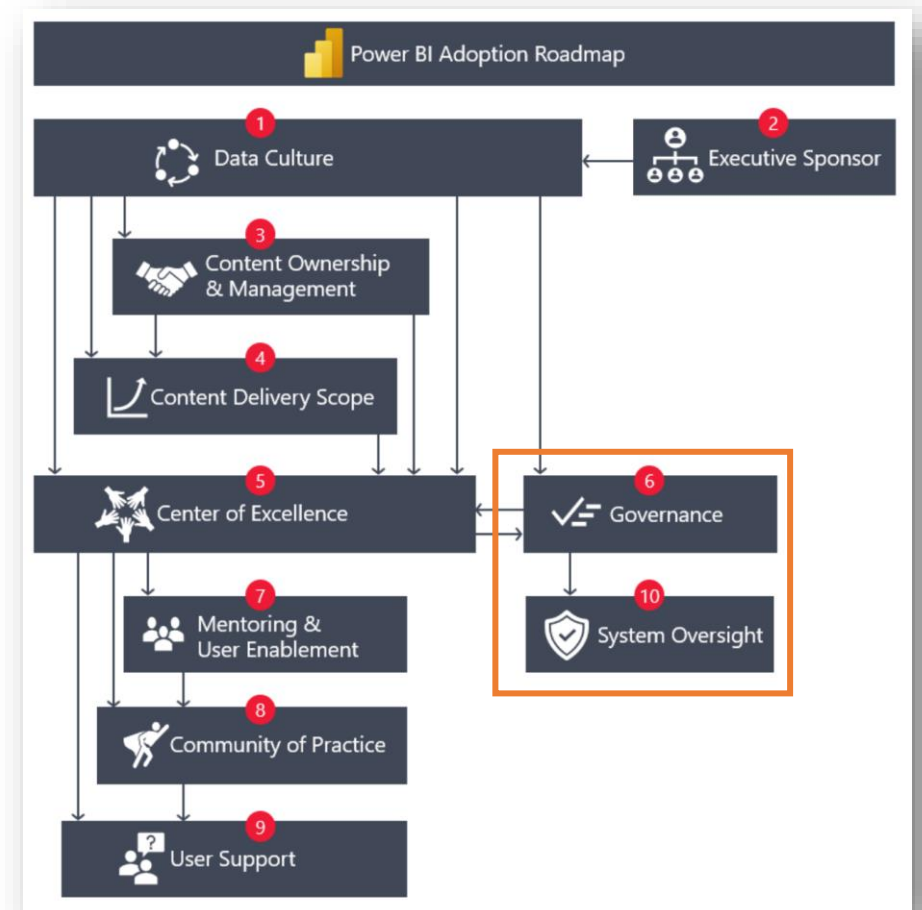
```
context):  
context.active_object is not
```

Why Monitoring?

Power BI Monitoring is one of the pillars for a good Power BI Governance, allows you to:

- Focus on what really matters
- Detect Patterns & react to them
- Get Insights & Anticipate issues (ex: wrong sharing, non authorized export)
- Enforce guidelines (ex: naming convention)

Power BI Adoption Roadmap



How? – Simplest (Easiest)

Data Sources



PBI REST API

- Metadata / Async
- Activity (**30 days**)



Graph API

- Users
- Licenses

Extract



PBI - FetchCatalog.ps1
PBI - FetchDataSetRefresh.ps1



PBI - FetchActivity.ps1



PBI - FetchGraph.ps1



Config.json

Store



Local Folders

- Metadata
- Activity
- Graph

Analyze



Power BI DataSet & Report

How? – Azure (Recommended)

Data Sources



PBI REST API

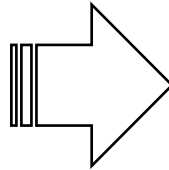
- Metadata / Async
- Activity (**30 days**)



Graph API

- Users
- Licenses

Extract



Azure Function



Key Vault



App Insights

Store



Azure Storage

\Metadata
\Activity
\Graph

Analyze



Power BI Report

API's Overview

Scope	Resource	API
Activity	Power BI Activity Logs	Admin API - Activity Events
Power BI Metadata	Workspaces DataSets Reports Dashboards Permissions Schema & Lineage Apps	<ol style="list-style-type: none"> 1. Admin Scan API – GetModifiedWorkspaces 2. Admin Scan API – PostWorkspaceInfo 3. Admin Scan API – GetScanStatus (loop) 4. Admin Scan API – GetScanResult Admin API - GetAppsAsAdmin
	RefreshHistory	Admin API - GetGroupsAsAdmin + Expand DataSets Dataset API - Get Refresh History
Users & Licenses	Users & Licenses	Graph API – Users
	Licenses Details	Graph API – SubscribedSKUs

Productivity

- Rebinding
- Premium Capacity Switch
- Devops

```
mirror_mod = modifier_ob.  
Set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
@selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

--- OPERATOR CLASSES ---

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

```
context):  
context.active_object is not
```

Power BI & DevOps – A simple approach

Avoid multiple development files

Declare your deployment strategy

Automate the deployment using REST API's

You can make mistakes, the computer don't 😊

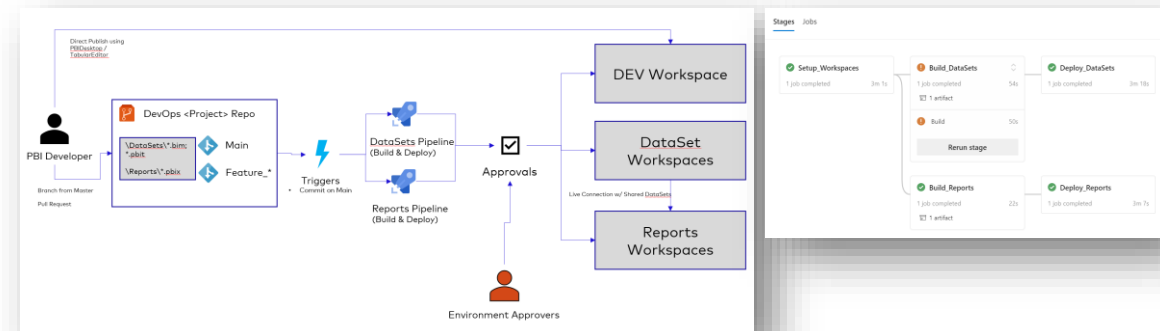
If you have Premium

Deployment Pipelines

Must Watch: [Daniel Otykier Session](#)

- Azure DevOps
- Tabular Editor / bim file vs pbix
- REST API's

```
26
27 Connect-PowerBIServiceAccount
28
29 if ($workspaces)
30 {
31     Publish-PBIWorkspaces -configPath $configPath
32 }
33
34 if ($datasets)
35 {
36     Publish-PBIDataSets -configPath $configPath -path "$path\DataSets"
37 }
38
39 if ($reports)
40 {
41     # README - The live connected PBIX reports need to be binded to an existent Datas
42     Publish-PBIReports -configPath $configPath -path "$path\Reports"
43 }
44
45 if ($paginatedReports)
46 {
47     Publish-PBIReports -configPath $configPath -path "$path\PaginatedReports"
48 }
49
```



Workspace Premium Capacity Switch

Why?

Need to reassign all workspaces to another capacity:

- P Sku <-> A Sku <-> PPU
- Ex: Gen 1 to Gen 2 migration

If there are many workspaces, please don't do it manually 😊

API's:

- [AssignWorkspacesToCapacity](#)

Download: [here](#)

```
ice-Premium Capacity Switch.ps1 | Report-Rebind-Manual.ps1

#Requires -Modules @{ ModuleName="MicrosoftPowerBIMgmt"; ModuleVersion="1.2.1026" }

cls

$sourceCapacityId = "BEBF8A28-B230-4187-AD24-92FE2ECEAD53" #rrpbiembedtestgen2
$targetCapacityId = "B841DB73-7A03-4349-BE78-2B81C32EC60F" #Premium Per User

$currentPath = (Split-Path $MyInvocation.MyCommand.Definition -Parent)

Connect-PowerBIServiceAccount

Write-Host "Getting workspaces"

# If your tenant has more than 5000 workspaces
$premiumWorkspaces = Invoke-PowerBIRestMethod -url "admin/groups?`$stop=5000&`$filter=isonDedicated"

$sourcePremiumWorkspaces = @($premiumWorkspaces |? {$_.capacityId -eq $sourceCapacityId})

if ($sourcePremiumWorkspaces.Count -gt 0)
{
    Write-Host "Assigning $($sourcePremiumWorkspaces.Count) workspaces to new capacity '$targetCapacityId'"

    $workspaceIds = @($sourcePremiumWorkspaces.id)

    # Unassign workspaces

    $body = @{
        capacityMigrationAssignments= @( @{
            targetCapacityObjectId = $targetCapacityId;
            workspacesToAssign = $workspaceIds
        } )
    }

    $bodyStr = ConvertTo-Json $body -Depth 3

    Invoke-PowerBIRestMethod -url "admin/capacities/AssignWorkspaces" -method Post -body $bodyStr
}
else
{
    Write-Host "No workspaces on source capacity: '$sourceCapacityId'"
}
```


REST API - Rebind Reports

Why?

Dataset issue/corruption

Minimize user impact on a dataset change / refresh

Azure AS to Premium Migrations

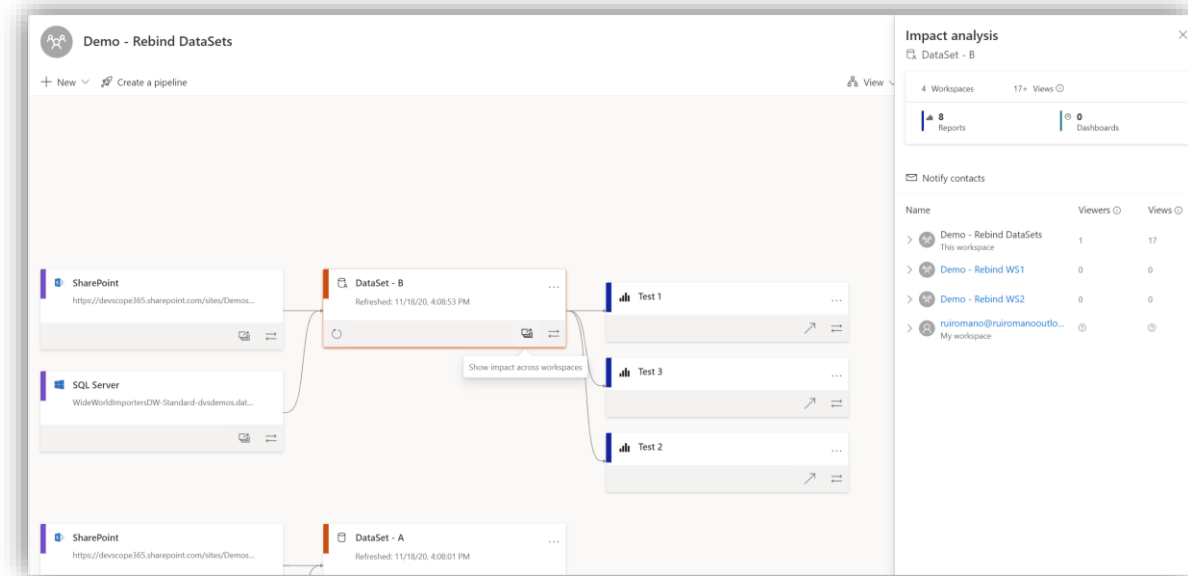
...

Two Approaches:

- Manual - By specifying the reports & datasets to rebind
- Automatic - From one dataset to the other

REST API: [Rebind API](#)

Links: [Automatic Rebind Script](#)



Thanks!

 @RuiRomano

 <https://www.linkedin.com/in/ruiromano/>

 <https://ruiromanoblog.wordpress.com>