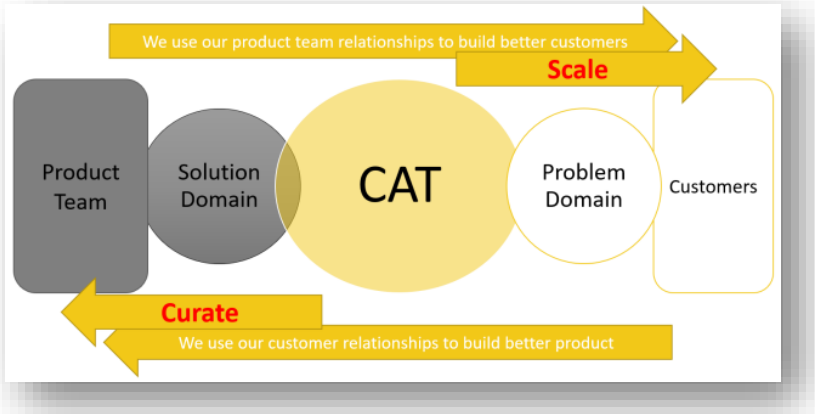
Power BI API Extravaganza

2022-01

Rui Romano

Power BI CAT

Microsoft



Lives in Porto, Portugal

Power BI & Data

+15 years working on Microsoft BI & Data Stack

@RuiRomano

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

https://ruiromanoblog.wordpress.com

Slides here:

https://github.com/RuiRomano/sessionslides

Recorded Session

Power BI API Extravaganza

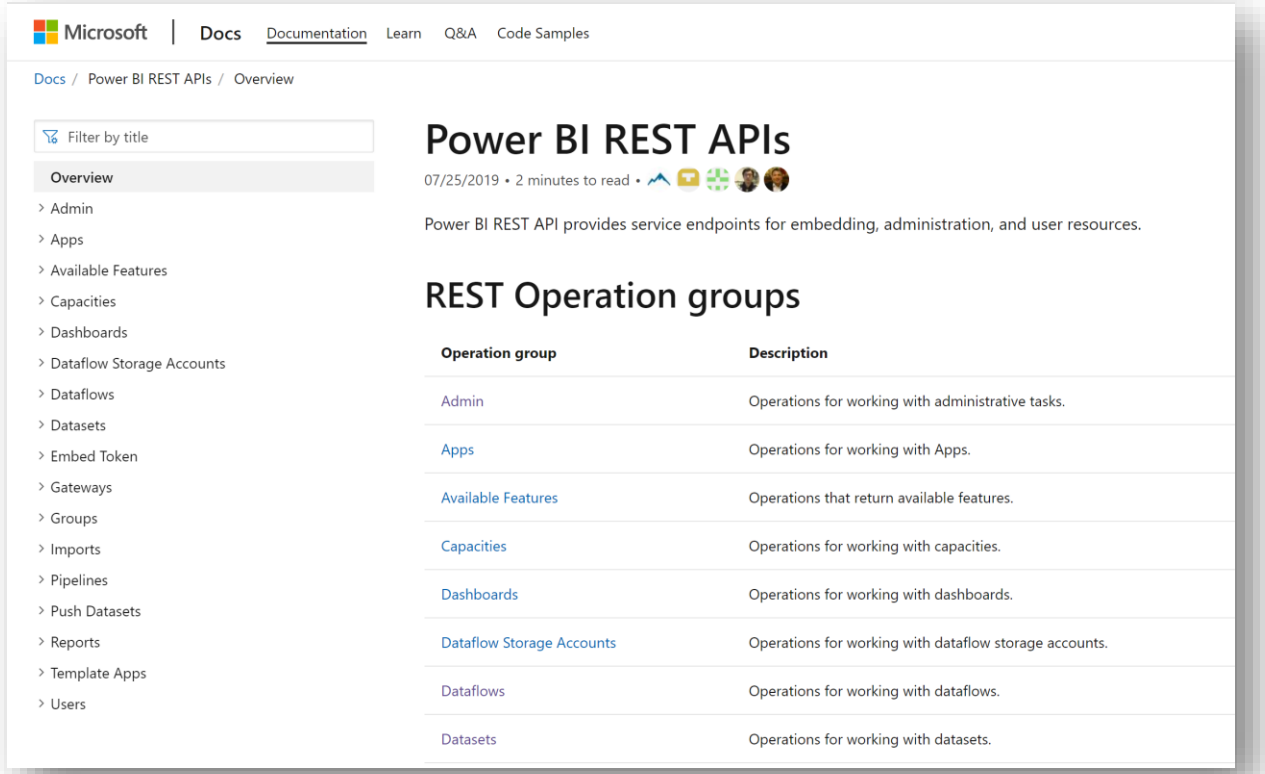
- Not a Power BI Intro!
- 100% focus on the REST API's
- Fast Paced ☺

Focus on the possibilities not details…

Power BI REST API’s ?

Collection of REST API’s for:

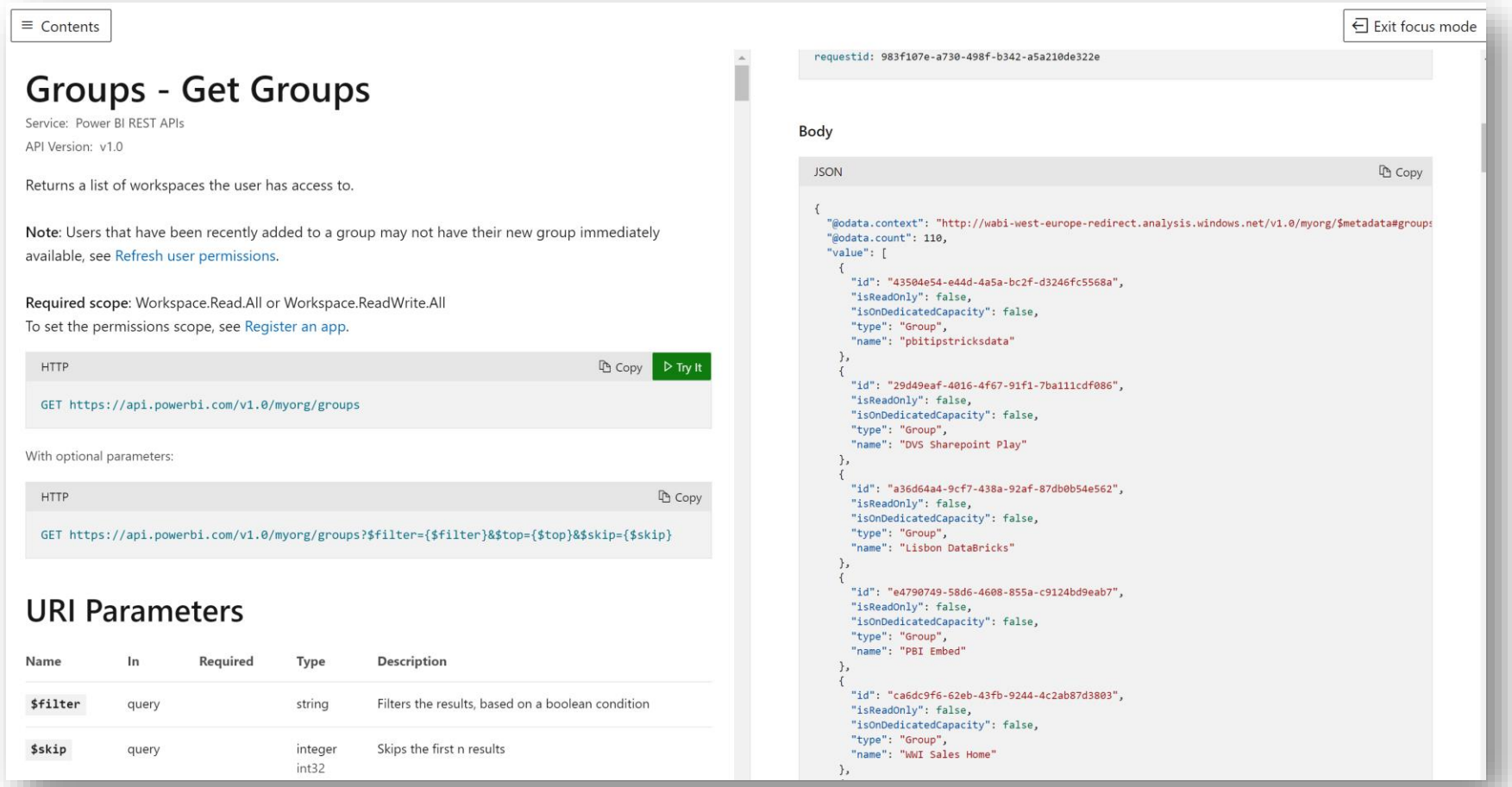
- Manage Power BI Content
- Admin Operations
- Monitoring
- Automation
- Real-Time
- Embedding



[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

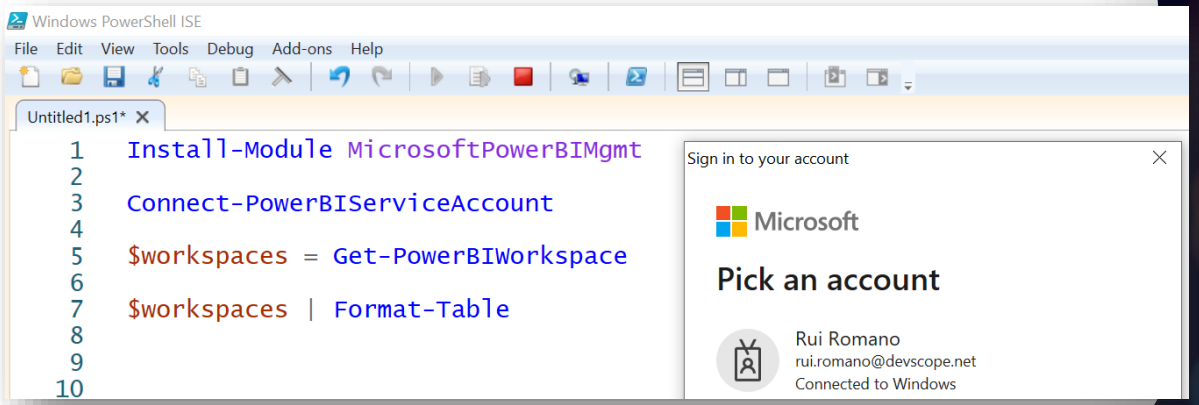


How can I work with the API's?

- 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:

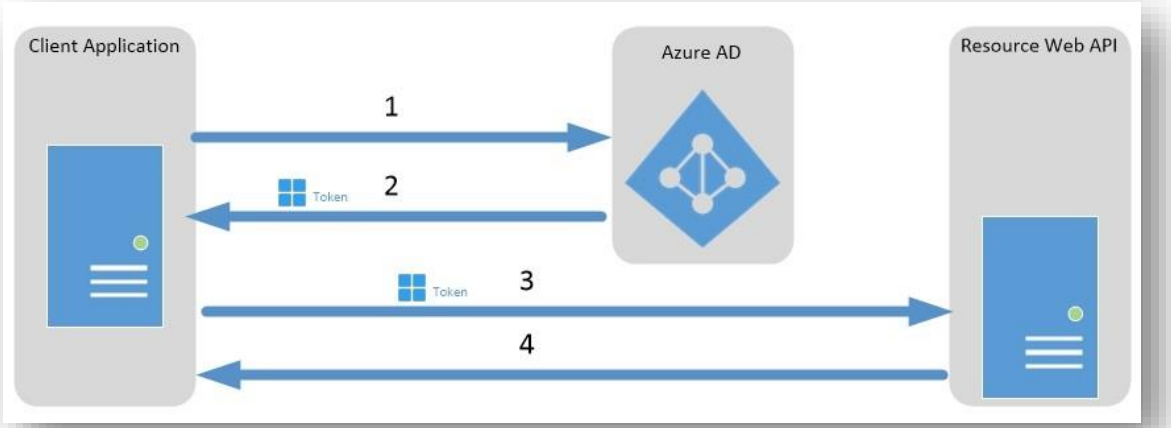
  PowerShell – MicrosoftPowerBIMgmt, PowerBIPS (💗)
  Python – python-power-bi
  .Net – Microsoft.PowerBI.Api

Demo MicrosoftPowerBIMgmt



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":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "scope":"create"
}
```

Service Principal – Manual Step by Step

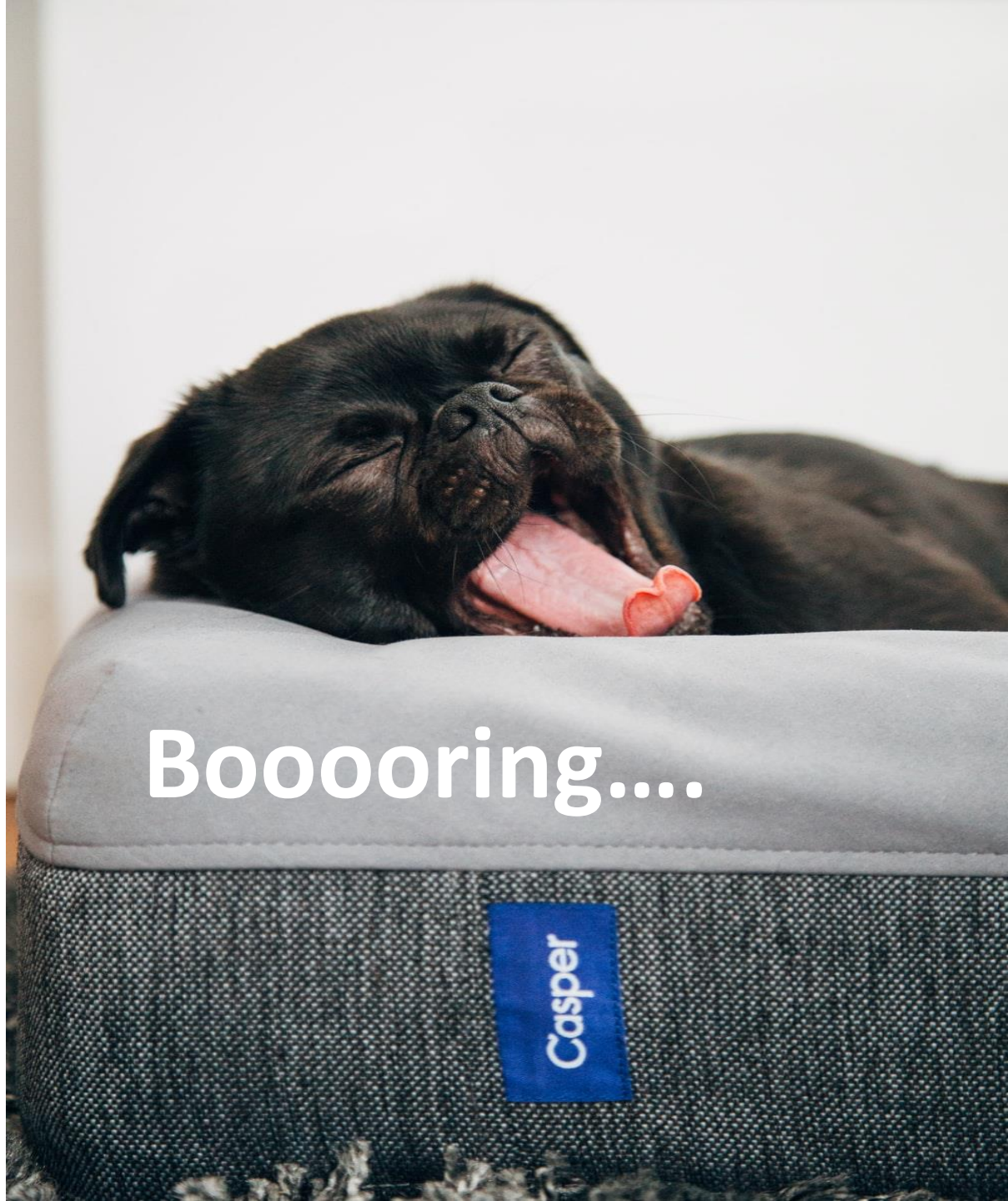
1. Go to Azure AD Active Directory
2. Go to App Registrations and create a new App (leave defaults)
3. Generate a new App Secret
4. Save the App Id, App Secret & Tenant Id
5. Create/Reuse an Azure AD Security Group
6. Add the Service Principal to the Security Group as a member
7. Authorize the Security 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

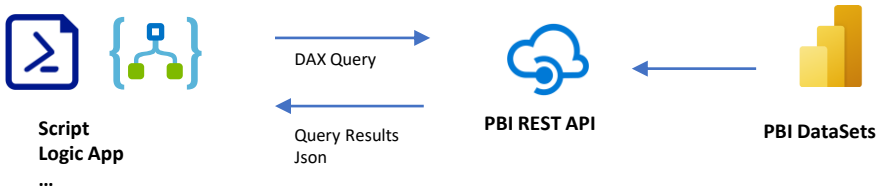
- DAX Queries w/ REST API
- Monitoring
- Productivity / DevOps
- Realtime

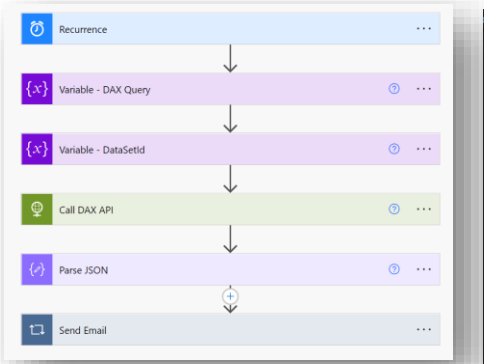


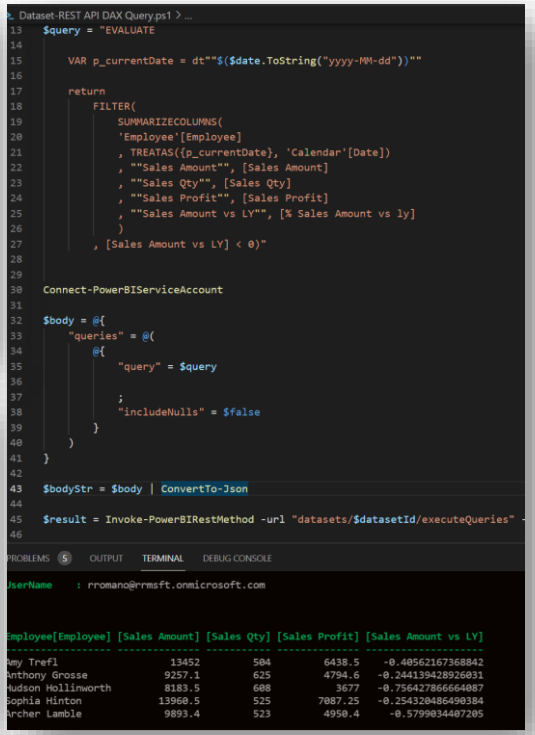
Power BI Execute Queries API

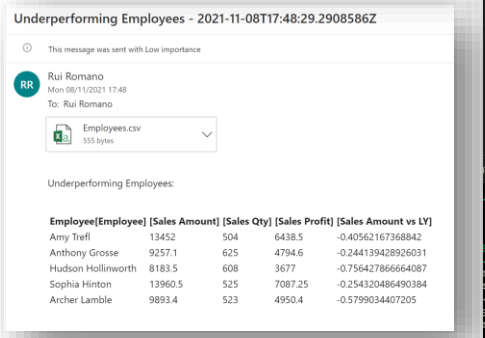
REST API - Dataset Execute Queries

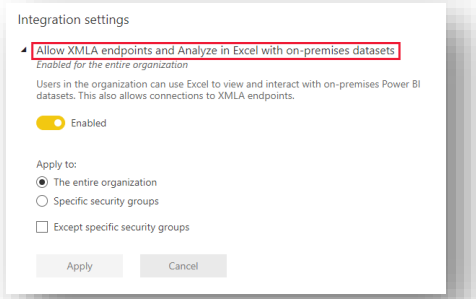
- The coolest API! ☺
- Useful Scenarios:
  Data Integration (*limited)
  Alerting System using Power BI Data
  App/Website integrations
- Works on Premium & **Shared (PRO)**
- Requirements
  “Build” permission on the dataset
  “Allow XMLA Endpoints and Analyze in Excel”
- Main Limitations
  Result sets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











Script: execute dax in PS

Power BI Monitoring

Full session on the topic
link

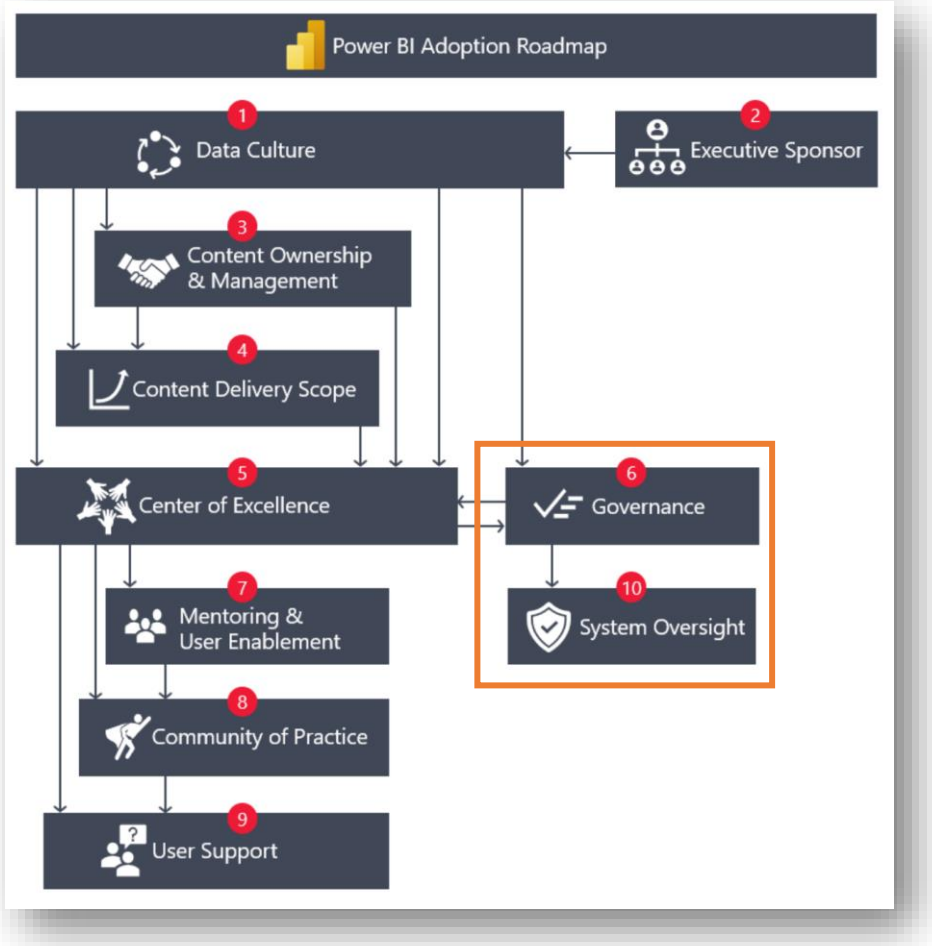
Code & Slides
link

Why Monitoring?

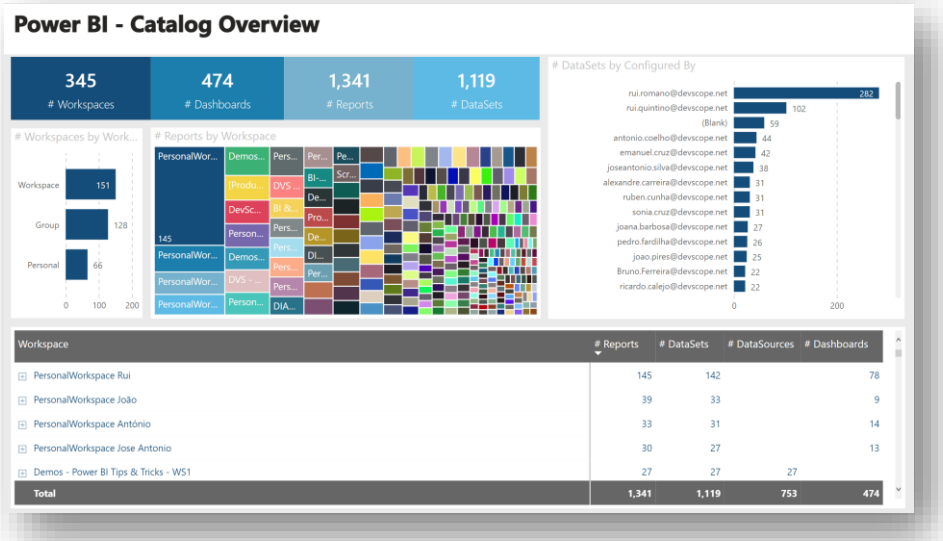
Power BI Monitoring is one of the pillars for a good Power BI Governance, allows you to:

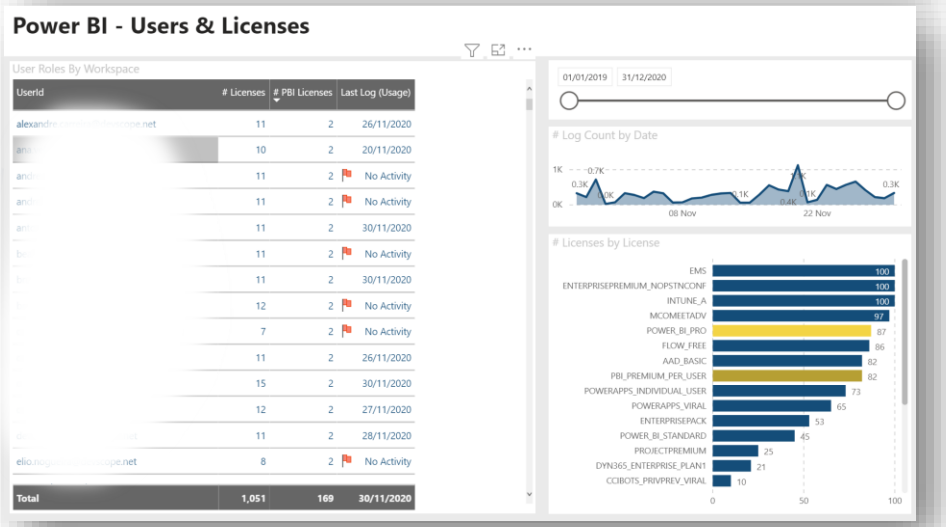
- Focus on what really matters
- Detect Patterns & react to them
- Get Insights & Anticipate issues (ex: wrong sharing, non authorized export)
- Enforce guidelines (ex: naming convention)

[Power BI Adoption Roadmap]



Power BI Monitoring Reports





How? – Azure (Recommended)

Data Sources

Extract

Store

Analyze

PBI REST API
- Metadata / Async
- Activity (**30 days**)

Graph API
- Users
- Licenses

Azure Function

Key Vault

App Insights

Azure Storage
\Metadata
\Activity
\Graph

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	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 / Ops

- Refresh
- Rebinding
- Premium Capacity Switch
- Devops

Async Refresh API's

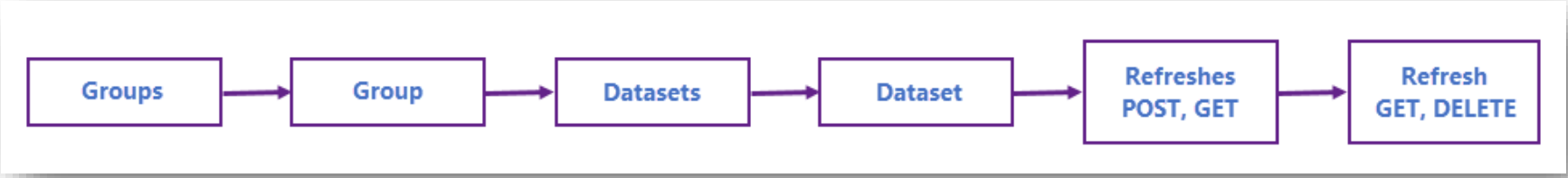
"Fire & Forget" refresh

Call it from scripts / logic apps / power automate

Great Enhancements

- Auto Retries
- Batched commits

```powershell
# Get running refreshes, only 1 operation is allowed

$refreshes = Invoke-PowerBIRestMethod -url "groups/$workspaceId/datasets/$datasetId/refreshes?`$top=10" -method Get | ConvertFrom-Json | select -ExpandProperty value

if (!($refreshes |? { $_.refreshType -eq "ReliableProcessing" -and $_.status -iin @("Unknown", "inProgress", "notStarted") }))
{
    Write-Host "Posting a new Refresh Command"

    Invoke-PowerBIRestMethod -url "groups/$workspaceId/datasets/$datasetId/refreshes" -method Post -Body $executeJsonBody
}

Write-Host "Waiting for refresh to end"

$refreshes = Invoke-PowerBIRestMethod -url "groups/$workspaceId/datasets/$datasetId/refreshes?`$top=10" -method Get | ConvertFrom-Json | select -ExpandProperty value

$refreshId = $refreshes[0].requestId

do
{
    $refreshDetails = Invoke-PowerBIRestMethod -url "groups/$workspaceId/datasets/$datasetId/refreshes/$refreshId" -method Get | ConvertFrom-Json

    Write-Host "Status: $($refreshDetails.status)"
    Write-Host "sleeping..."

    Start-Sleep -Seconds 2

}
while($refreshDetails.status -iin @("Unknown", "inProgress", "notStarted"))

Write-Host "Refresh complete: $((([datetime]$refreshDetails.endTime) - ([datetime]$refreshDetails.startTime)).TotalSeconds)s"
```



Links: sample script, docs

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 ☺

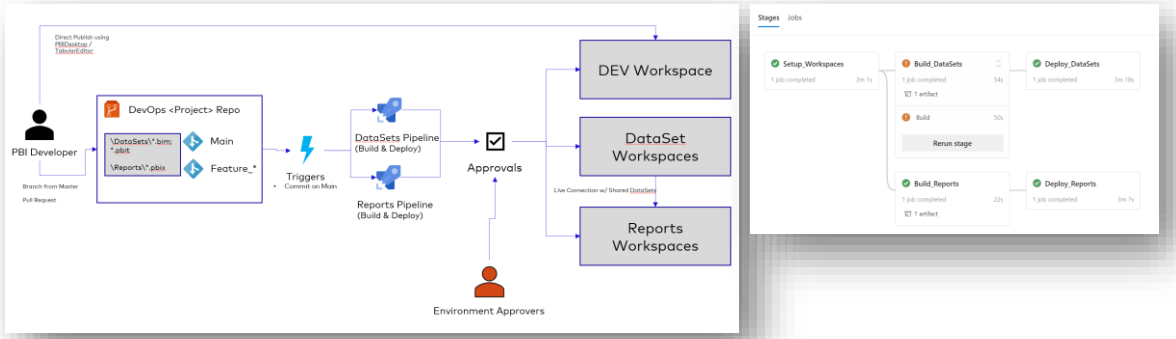
```powershell
Connect-PowerBIServiceAccount

if ($workspaces)
{
    Publish-PBIWorkspaces -configPath $configPath
}

if ($datasets)
{
    Publish-PBIDataSets -configPath $configPath -path "$path\DataSets"
}

if ($reports)
{
    # README - The live connected PBIX reports need to be binded to an existent Datas
    Publish-PBIReports -configPath $configPath -path "$path\Reports"
}

if ($paginatedReports)
{
    Publish-PBIReports -configPath $configPath -path "$path\PaginatedReports"
}
```

If you have Premium 💎

Deployment Pipelines

Must Watch: Daniel Otykier Session

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



Links: pbidevops

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]

```powershell
#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
$premiumWorkspaces = Invoke-PowerBIRestMethod -url "admin/groups?`$top=5000&`$filter=isOnDedicated

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

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

    $workspaceIds = @($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'"
}
```

Download: here

REST API - Rebind Reports

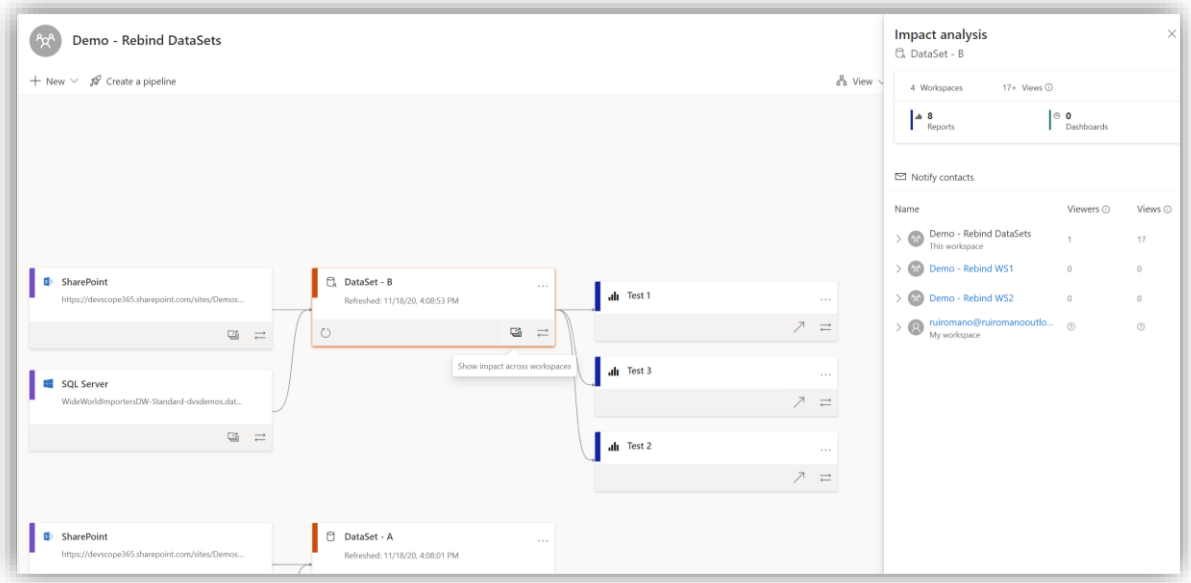
Why?

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]

Recap

- Avoid repetitive work
- Learn more about all the API’s avaiable
- Build your own collection of reusable scripts

Thanks!

 @RuiRomano

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

 https://ruiromanoblog.wordpress.com