



# Power Up Your Power BI

## Unlock the Magic of Automation with PowerShell

**Ioana Bouariu**

She/Her/They/Them

Consultant

Amesto Insight



# Ioana Bouariu

Consultant  
Amesto Insight

 [github.com/Jojobit](https://github.com/Jojobit)

JojobitTweets

 [Jojobit.bsky.social](https://Jojobit.bsky.social)

 [/in/ioanabouariu](https://in.ioanabouariu)



She/Her/They/Them

Data geek with a passion for AI, BI and everything data. Co-pilot in a crew of 5. Coding & automation enthusiast.

Power BI accidental admin and jack-of-all-trades at Insight, Amesto Solutions, a consulting company in Oslo, Norway.

# What is **PowerShell**

- PowerShell is a task automation and configuration management program from Microsoft, consisting of a command-line shell and the associated scripting language.
- It is particularly useful for automating repetitive tasks or managing complex systems.

# What is **Power BI**

- Power BI is a popular data visualization and business intelligence tool that can be used to create interactive reports, dashboards, and visualizations.

# Why PowerShell + Power BI



# What is **Power BI cmdlet**



- Small, specialized commands that are already defined in the Power BI module/package for PowerShell, which we have to install in PowerShell.
- Typically structured as verb-noun pairs, such as "Get-Item" or "New-Item".

# How Key cmdlets



`Get-PowerBIWorkspace` #Get a list of all the workspaces  
`New-PowerBIWorkspace -name` #Create a new workspace  
`Get-PowerBIReport` #Get a list of all the reports  
`New-PowerBIReport -path -name -workspace` #Upload a report  
`Get-PowerBIDataset` #Get a list of all the datasets  
`Add-PowerBIWorkspaceUser` #Adds users to a workspace

And [more](#).

# What is **Power BI REST API**



- The Power BI API is a REST-based API that allows you to access and manipulate data in Power BI.
- It is designed to be used with any programming language that can make HTTP requests, including PowerShell.



# How Power BI REST API



```
Invoke-PowerBIRestMethod -Method GET -Url  
("datasets/"+dataset.id+"/parameters")
```

#Get a list of the parameters of a dataset in workspace

```
Invoke-PowerBIRestMethod -Method POST -Url  
("groups/"+workspace.id+"/datasets/"+dataset.id+"/refresh  
es")
```

#Trigger a refresh of a specific dataset in workspace

And [more](#).

# Deep stuff

## Advanced topics: if statement

```
if (Get-PowerBIWorkspace -name "Demo") {  
    Write-Host "Workspace already exists"}  
  
else {  
    New-PowerBIWorkspace -Name "Demo"  
}
```



# Deep stuff

## Advanced topics: foreach loop

```
foreach ($company in $CompanyList) {  
    New-PowerBIWorkspace -Name $company  
}
```

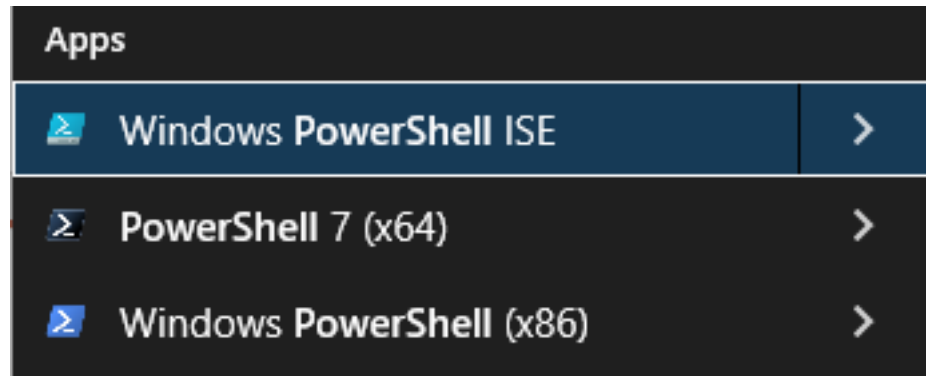
# Deep stuff

## **Advanced topics: pipe**

Get-PowerBIWorkspace | where-object name -like  
"Demo\*"

# Where The environment

- PowerShell ISE



- Visual Studio Code – with GitHub Copilot

# Let's head over to **DEMO**

```
        parameter for "$di.name" in "$w.name"  
        method post -Url ("datasets/" + $di.id + "/Default.UpdateParameters")  
  
    now  
    {  
        c -workspaceid $w.Id  
        .RestMethod -Method get -Url ("datasets/" + $di.id + "/parameters") | Convert  
        parameter for "$di.name" in "$w.name" is "$check"
```

```
    reports, at once, in all the workspaces  
    foreach ($workspaces) {  
        dataset -workspaceid $w.id  
        PowerBIDataset -workspaceid $w.Id  
        PowerBIRestMethod -Method Post -Url ("groups/" + $w.id + "/datasets/" + $d.id + "/refreshes")
```



Install-Module Fabtools

# How Key Fabtools cmdlets



Invoke-FabricDatasetRefresh #Start a dataset refresh

Get-FabricDatasetRefreshes #Get the refresh history

Get-FabricWorkspaceUsers #Get the users of a workspace

New-FabricWorkspaceUsageMetrics #Create usage metrics  
report

Remove-FabricWorkspace #Delete a workspace





# How Key Fabtools cmdlets

Get-AllFabricCapacities #Find all Fabric capacities

Set-FabricWorkspaceToCapacity #Move a workspace to another  
Fabric capacity

Get-FabricCapacityState #Get the current state of a Fabric  
capacity

Suspend-FabricCapacity #Pause a Fabric capacity

Resume-FabricCapacity #Activate a Fabric capacity

And [more](#).

# Finally

## Last tips and conclusion

### Tips:

- When in doubt, **automate**
- **Keep it simple** and easy to follow
- Use the PBI module and **cmdlets**
- **Test your code** before setting it in production
- **Use source control** to manage your code and version history
- Use the resources around you.


**Collaborate** with others.

# New-PowerBIWorkspace

# #BuildSomethingAwesome

# Thank you

## Reach out

 [github.com/Jojobit  
/in/ioanabouariu/](https://github.com/Jojobit/in/ioanabouariu/)

 [@JojobitTweets](https://twitter.com/JojobitTweets)

 [@jojobit.bsky.social](https://bsky.app/profile/jojobit.bsky.social)



# Q&A zone

What are your questions?

[github.com/Jojobit](https://github.com/Jojobit)  
[/in/ioanabouariu/](https://www.linkedin.com/in/ioanabouariu/)  
[@JojobitTweets](https://twitter.com/JojobitTweets)  
[@jojobit.bsky.social](https://bsky.app/profile/jojobit.bsky.social)



# Session evaluation

Your feedback is important to us



**Evaluate this session at:**

[www.PASSDataCommunitySummit.com/evaluation](http://www.PASSDataCommunitySummit.com/evaluation)

