

Learn to Automate Identity Management like a Pro with Graph PowerShell!

 Jan Vidar Elven | Senior Architect @ Evidi | MVP Security |
@JanVidarElven

Where are we heading?

- The past:
 - MSOL PowerShell 😐 (Get-MsolUser)
- The present up to now..
 - AzureAD / AzureAD PowerShell 😊 (Get-AzureADUser)
 - EOL soon..
- The now-> Future!
 - PowerShell Graph SDK FTW! 🥳 (Get-MgUser)

Q: ? 🤔

- What is the biggest drawback of the MSOL and Azure AD PowerShell modules??

- A: 🧐 They are dependent on Windows Powershell 😊

Graph PowerShell SDK 101

- Install, Update & Configure
- 1.0 vs Beta (This has changed from SDK v1 to v2!)
- Graph Modules
- How to Connect
- Permission Scopes
 - Delegated
 - Application

Install Graph PS SDK

```
# Install Current User
```

```
Install-Module -Name Microsoft.Graph -Scope CurrentUser
```

```
# Install for All Users (Admin privilege)
```

```
Install-Module -Name Microsoft.Graph -Scope AllUsers
```

```
# Install Beta
```

```
Install-Module -Name Microsoft.Graph.Beta
```

Verify Modules

```
Get-Module -Name Microsoft.Graph - ListAvailable -All
```

```
Get-InstalledModule -Name Microsoft.Graph.*
```

Connect to Graph

Interactive authentication:

```
Connect-MgGraph
```

Device authentication:

```
Connect-MgGraph -UseDeviceAuthentication
```

Access token:

```
Connect-MgGraph -AccessToken $AccessToken
```

Application:

```
Connect-MgGraph -ClientId "YOUR_APP_ID"  
-TenantId "YOUR_TENANT_ID" -CertificateThumbprint "YOUR_CERT_THUMBPRINT"
```

Managed Identity:

```
Connect-MgGraph -Identity
```

<https://learn.microsoft.com/en-us/powershell/microsoftgraph/authentication-commands?view=graph-powershell-1.0>

Scopes | Roles | Permissions

```
# Connect with Scopes
```

```
Connect-MgGraph -Scopes User.Read.All
```

```
# Find necessary permissions
```

```
Find-MgGraphCommand -command Get-MgUser |  
    Select -First 1 -ExpandProperty Permissions
```


Use Docs and Graph Explorer

• [https://](https://learn.microsoft.com/en-us/graph/api/user-get?view=graph-rest-1.0&tabs=powershell)

resource for

The screenshot shows the Microsoft Graph REST API Explorer interface. The URL bar displays `https://learn.microsoft.com/en-us/graph/api/user-get?view=graph-rest-1.0&tabs=powershell`. Below the URL, a message states "requires more time to complete related background operations." The "Version" dropdown is set to "Microsoft Graph REST API v1.0". The "Examples" tab is selected, and the "PowerShell" language is chosen. The response status is "OK - 200 - 127ms". The "Code snippets" tab is active, showing the following PowerShell code:

```
# Leverage the power of our client libraries. Download the powershell client library here https://aka.ms/pshellsdk
# To read more about our SDKs, go to the documentation page at https://aka.ms/pshellsdkdocs
Import-Module Microsoft.Graph.Users

Get-MgUser -UserId $userId -Property "displayName,givenName,surName,userPrincipalName,mobilePhone"
```

At the bottom, there are "Update" and "Delete" buttons.

Azure AD Compatibility Adapter

- Uses Graph PowerShell SDK

```
Install-Module Microsoft.Graph.Compatibility.AzureAD -AllowPrerelease
```

```
Import-Module Microsoft.Graph.Compatibility.AzureAD -Force
```

- Authenticates the same way as Graph PS (Connect-MgGraph)

```
*CompatAD* → Get-CompatADUser ..
```

Demo Graph PowerShell ++



Serverless Automation for Graph PowerShell

- Azure Functions
 - Useful for both automation, scheduling and API request scenarios
- Azure Automation
 - Useful for scheduling and automation



Azure Functions PowerShell

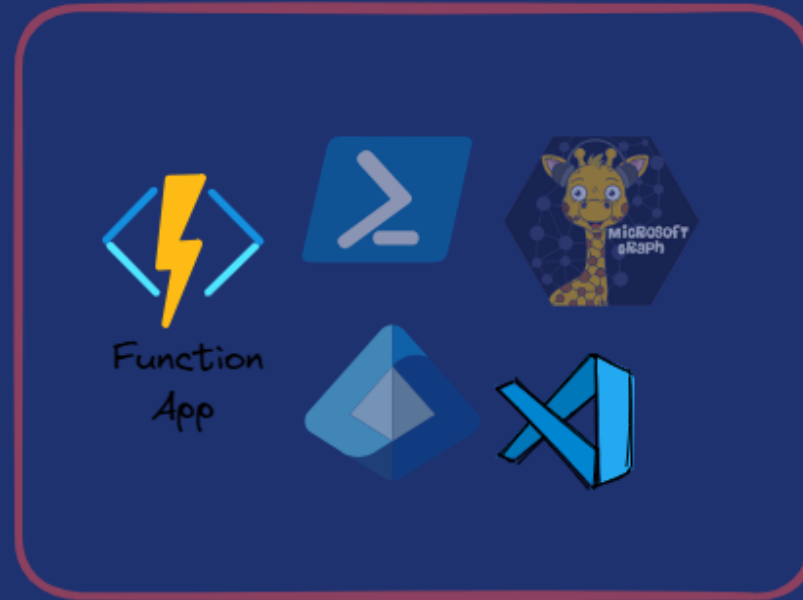
- Develop in Function App 😊
- Develop locally in Visual Studio Code 🤖
 - Azure Function Core Tools [Develop Azure Functions locally using Core Tools | Microsoft Learn](#)
 - `func init MyProjFolder --worker-runtime powershell`
 - `func new --template "Http Trigger" --name MyHttpTrigger`

The Requirements.psd1 file

- Managed Dependencies
- Load Graph PowerShell modules automatically ✨

```
1 # This file enables modules to be automatically managed by the Functions service.
2 # See https://aka.ms/functionsmanageddependency for additional information.
3 #
4 @{
5     # For latest supported version, go to 'https://www.powershellgallery.com/packages/Az'. Uncomment the ne
6     # 'Az' = 'MAJOR_VERSION.*'
7     'Microsoft.Graph.Authentication' = '2.*'
8 }
```

Demo Azure Functions PowerShell & MS Graph!



Bonus! Azure DevOps with Workload Identity Federation!

- Run Azure Pipelines as federated identity as service connection to access Microsoft Graph PowerShell SDK!
- <https://gotoguy.blog/2023/09/15/connect-to-microsoft-graph-in-azure-devops-pipelines-using-workload-identity-federation/>

Presentation & Demo Source : \>

<https://github.com/JanVidarElven/ExpertsLiveEurope2023>



@JanVidarElven

Thank you to our Sponsors

