The Azure AD-ventures with Microsoft Graph and PowerShell

Aleksandar Nikolić | Microsoft MVP

AXXES

Arxus

codit

delaware

devoteam

DEXMACH
A CEGEKA COMPANY

Microsoft

ORDINA

inetum realdolmen
Positive digital flow

proximus

ZURE

Thank you partners!

Integration.team
Connecting anything, anytime, anywhere

NOEST

datarobots

JET BRAINS

U2U

Introduction to Microsoft Graph PowerShell

Microsoft Graph is a unified endpoint for accessing data, intelligence, and insights in Microsoft 365.

Why Microsoft Graph PowerShell Module?

Acts as an API wrapper for Microsoft Graph APIs, exposing the entire API set for use in PowerShell.

It will help administer every Entra ID feature that has an API in Microsoft Graph.

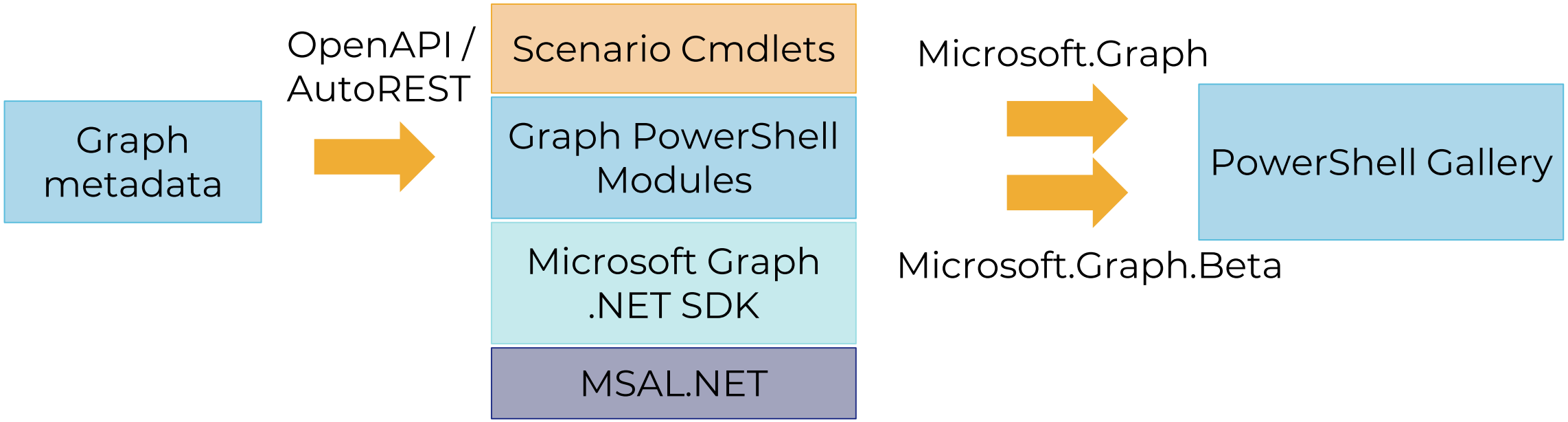
Microsoft Graph PowerShell

The commands in Microsoft Graph PowerShell are **autogenerated** from the Microsoft Graph API schema making it easier to get faster updates and functionality.

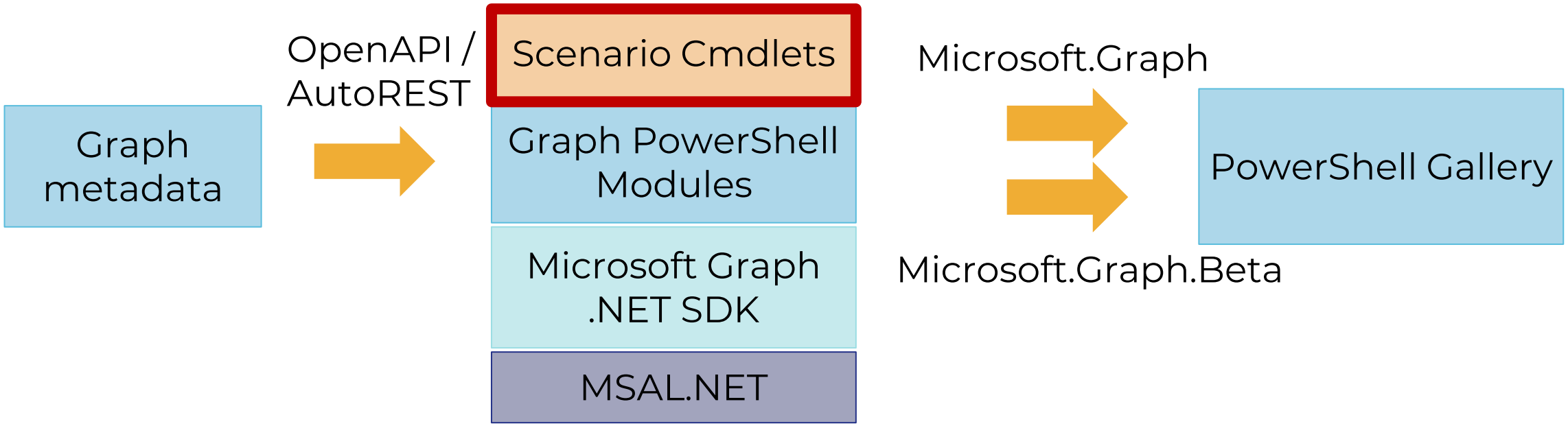
The cmdlet reference content is **also autogenerated** from the API reference.

Microsoft Graph PowerShell is the replacement for the Azure AD PowerShell and MSOnline modules and is recommended for interacting with Entra ID.

Microsoft Graph PowerShell



Microsoft Graph PowerShell



Microsoft Graph PowerShell features & benefits

Access to all Microsoft Graph APIs

Supports PowerShell 7

Supports modern authentication

Uses least privilege

Advanced queries

Open source

How to authenticate to Microsoft Graph

Least Permission Model

Permission handling differs significantly between the Azure AD PowerShell module and the Microsoft Graph PowerShell SDK.

Unlike permissions inherited from signed-in account, the permissions used by the SDK are granted to the service principal used to run SDK cmdlets.

For interactive sessions, the service principal is the Microsoft Graph

You must request permissions to perform actions, even when connecting with a highly-permissioned account

How to Figure Out What Microsoft Graph Permissions You Need

Guess and hope it will work ☺

- The danger here is that you end up with a heavily-permissioned service principal

Use the Graph Explorer (https://aka.ms/ge)

- The permissions required to run the query in the *Modify permissions* tab

Read the Microsoft Graph documentation

- Microsoft Graph REST API references (v1.0 and beta)
- Microsoft Graph permissions reference

Use **`Find-MgGraphPermission`** and **`Find-MgGraphCommand`** to identify Graph permissions

Source: https://practical365.com/microsoft-graph-api-permission

Authentication

The PowerShell SDK supports two types of authentication: delegated access and app-only access.

Delegated access: Log in as a user, grant consent to the SDK to act on our behalf, and call the Microsoft Graph.

App-only access: Grants permissions directly to an application, and requires an administrator to consent to the required permission scopes.

Delegated Access

Delegated access uses a public client to get an access token and consume Microsoft Graph resources on behalf of the signed-in user.

Microsoft Graph PowerShell module supports the following delegated access scenarios:

- **Interactive Browser**

```
Connect-MgGraph -Scopes "User.ReadBasic.All", "Calendars.Read.Shared“
```

- **Device Code**

```
Connect-MgGraph -Scopes "User.ReadBasic.All", "Calendars.Read.Shared" -
UseDeviceCode
```

App-only Access

App-only access uses a confidential client to get an access token and consume Microsoft Graph resources without a user context (uses an app's context).

Microsoft Graph PowerShell module supports the following app-only access scenarios:

- Client credential via a certificate
- Client credential via client secret (v2)
  - Using PSCredential object
  - Using environment variables
- Managed Identity (v2)

Managed identities and MSGraph

DEMO

Migrate to Microsoft Graph PowerShell

Azure AD Graph Deprecation and Retirement

In 2019, Microsoft announced deprecation of the Azure AD Graph service.

In 2022, Microsoft communicated that Azure AD Graph will be retired and stop functioning after June 30, 2023.

On June 15, 2023, Microsoft announced that they’ve change their mind

Enter the retirement cycle for Azure AD Graph APIs

The first step will involve blocking newly created applications from using Azure AD Graph APIs.

Legacy PowerShell Modules Deprecation and Retirement

MSOnline, AzureAD, and AzureADPreview modules will be deprecated on March 30, 2024

Only security fixes will be offered after deprecation

When these modules are deprecated, they will continue to work for a minimum of six (6) months before being retired.

Users should update PowerShell scripts to use Microsoft Graph PowerShell SDK module

Upgrading to Microsoft Graph PowerShell

Scripts written in Azure AD PowerShell won't automatically work with Microsoft Graph PowerShell.

The new cmdlet names use the Mg prefix.

However, migration is more than just becoming familiar with the new cmdlet names.

There are renamed modules, cmdlets, parameters, and other important changes.

The biggest challenge: returned objects are very different.

Resources

Cmdlet Map

https://learn.microsoft.com/en-us/powershell/microsoftgraph/azuread-msoline-cmdlet-map

API Reference

https://learn.microsoft.com/en-us/graph/api/overview

Graph Explorer

https://developer.microsoft.com/en-us/graph/graph-explorer

Graph PowerShell Conversion Analyzer

https://graphpowershell.merill.net/

Resources

Azure AD PowerShell to Microsoft Graph PowerShell migration FAQ

https://learn.microsoft.com/en-us/powershell/azure/active-directory/migration-faq

Microsoft.Graph.Compatibility.AzureAD module (preview)

https://www.powershellgallery.com/packages/Microsoft.Graph.Compatibility.AzureAD

PSAzureMigrationAdvisor

https://github.com/FriedrichWeinmann/PSAzureMigrationAdvisor

Migrate Azure AD script to Microsoft Graph PowerShell

DEMO

Fixing Microsoft Graph PowerShell

DEMO

Let’s have some fun fixing Microsoft Graph PowerShell!

Thank you.

https://github.com/alexandair/cloudbrew23