Managing Microsoft 365 through the command-line

Thomas Vochten

About me

Thomas Vochten ☕ 🧑‍💻 🏃

technology evangelist Microsoft 365 & Security

🦋 thomasvochten.com

✉ mail@thomasvochten.com

MVP Microsoft® Most Valuable Professional

BIWUG

CRONOS GROEP

Agenda

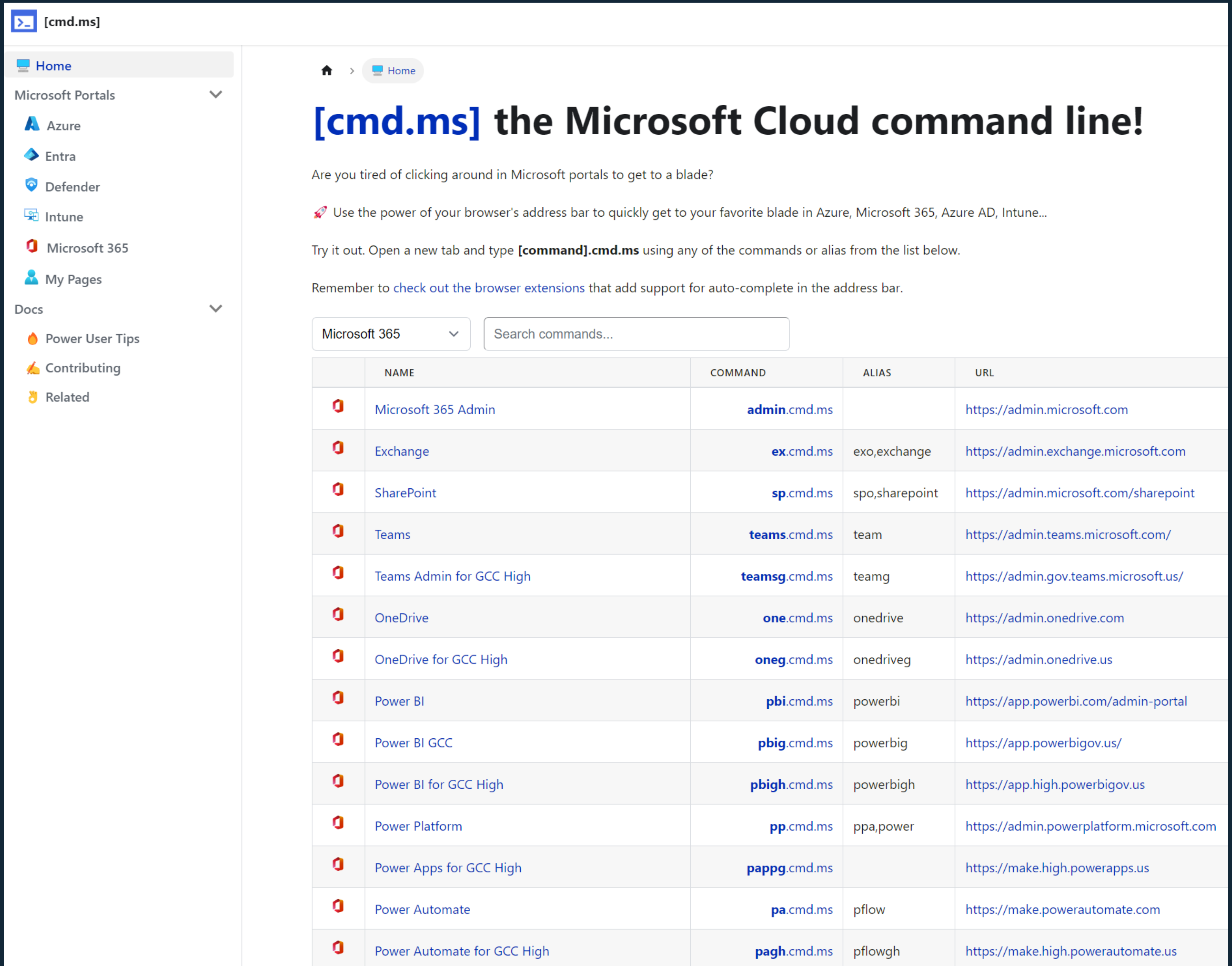
1. Managing Microsoft 365 in a nutshell
2. Getting to know the Microsoft Graph
3. PowerShell to the rescue
4. The rise of the CLI
5. Integration & automation
6. When to use what

Managing Microsoft 365 in a nutshell

Anyone need an admin portal?

- admin.microsoft.com
- security.microsoft.com
- compliance.microsoft.com
- entra.microsoft.com
- intune.microsoft.com
- …

cmd.ms 💎



[cmd.ms] the Microsoft Cloud command line!

Are you tired of clicking around in Microsoft portals to get to a blade?

🚀 Use the power of your browser's address bar to quickly get to your favorite blade in Azure, Microsoft 365, Azure AD, Intune...

Try it out. Open a new tab and type **[command].cmd.ms** using any of the commands or alias from the list below.

Remember to check out the browser extensions that add support for auto-complete in the address bar.

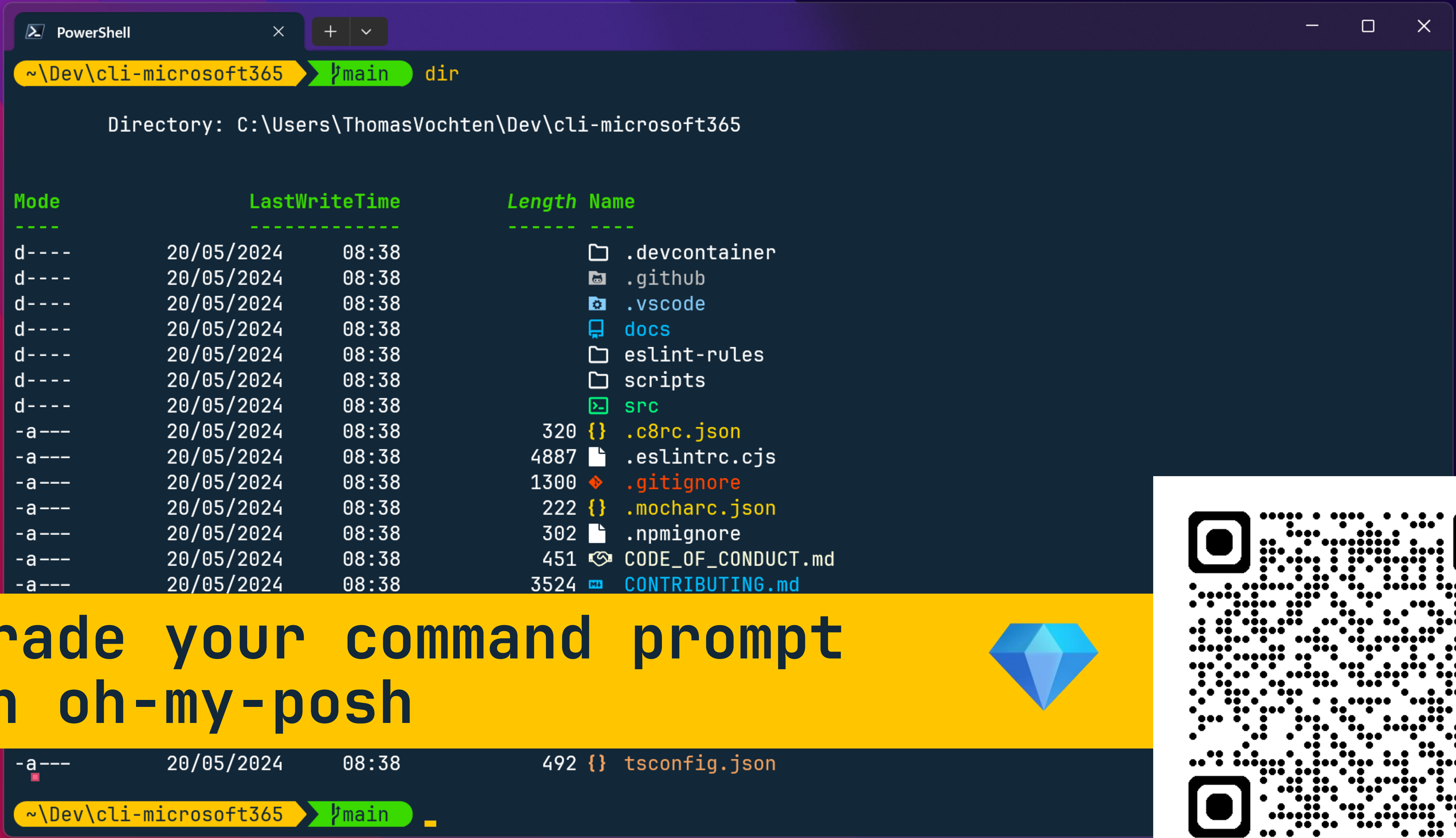
	NAME	COMMAND	ALIAS	URL
	Microsoft 365 Admin	**admin**.cmd.ms		https://admin.microsoft.com
	Exchange	**ex**.cmd.ms	exo,exchange	https://admin.exchange.microsoft.com
	SharePoint	**sp**.cmd.ms	spo,sharepoint	https://admin.microsoft.com/sharepoint
	Teams	**teams**.cmd.ms	team	https://admin.teams.microsoft.com/
	Teams Admin for GCC High	**teamsg**.cmd.ms	teamg	https://admin.gov.teams.microsoft.us/
	OneDrive	**one**.cmd.ms	onedrive	https://admin.onedrive.com
	OneDrive for GCC High	**oneg**.cmd.ms	onedriveg	https://admin.onedrive.us
	Power BI	**pbi**.cmd.ms	powerbi	https://app.powerbi.com/admin-portal
	Power BI GCC	**pbig**.cmd.ms	powerbig	https://app.powerbigov.us/
	Power BI for GCC High	**pbigh**.cmd.ms	powerbigh	https://app.high.powerbigov.us
	Power Platform	**pp**.cmd.ms	ppa,power	https://admin.powerplatform.microsoft.com
	Power Apps for GCC High	**pappg**.cmd.ms		https://make.high.powerapps.us
	Power Automate	**pa**.cmd.ms	pflow	https://make.powerautomate.com
	Power Automate for GCC High	**pagh**.cmd.ms	pflowgh	https://make.high.powerautomate.us

The tools I use

Windows Terminal

Visual Studio Code

Windows Subsystem
for Linux



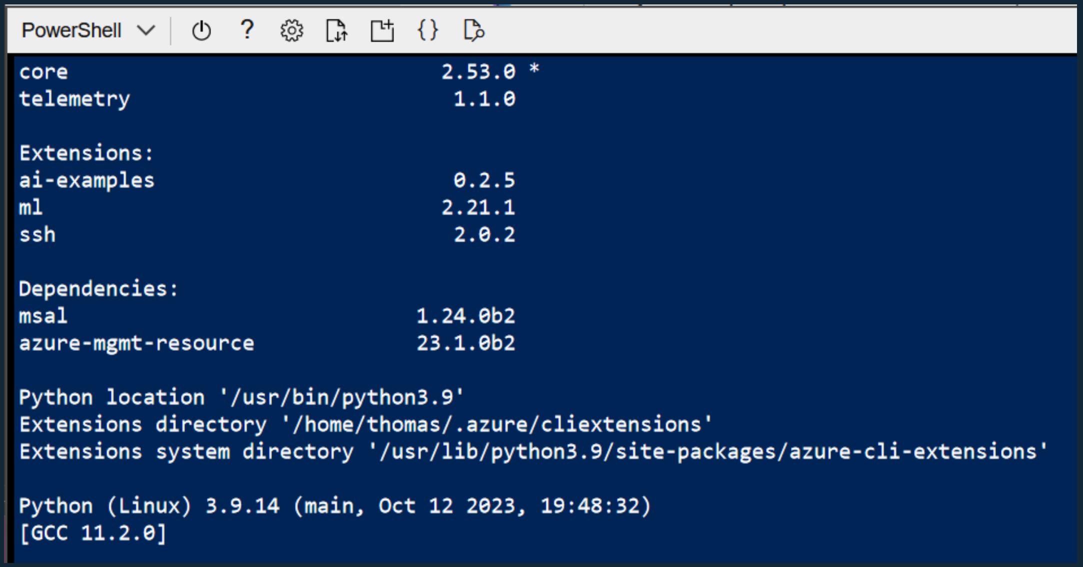
```
~\Dev\cli-microsoft365  main  dir

        Directory: C:\Users\ThomasVochten\Dev\cli-microsoft365


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          20/05/2024     08:38                .devcontainer
d----          20/05/2024     08:38                .github
d----          20/05/2024     08:38                .vscode
d----          20/05/2024     08:38                docs
d----          20/05/2024     08:38                eslint-rules
d----          20/05/2024     08:38                scripts
d----          20/05/2024     08:38                src
-a---          20/05/2024     08:38            320 .c8rc.json
-a---          20/05/2024     08:38           4887 .eslintrc.cjs
-a---          20/05/2024     08:38           1300 .gitignore
-a---          20/05/2024     08:38            222 .mocharc.json
-a---          20/05/2024     08:38            302 .npmignore
-a---          20/05/2024     08:38            451 CODE_OF_CONDUCT.md
-a---          20/05/2024     08:38           3524 CONTRIBUTING.md
-a---          20/05/2024     08:38            492 tsconfig.json

~\Dev\cli-microsoft365  main  _
```

Upgrade your command prompt with oh-my-posh

https://ohmyposh.dev/

Cloud shell 💎

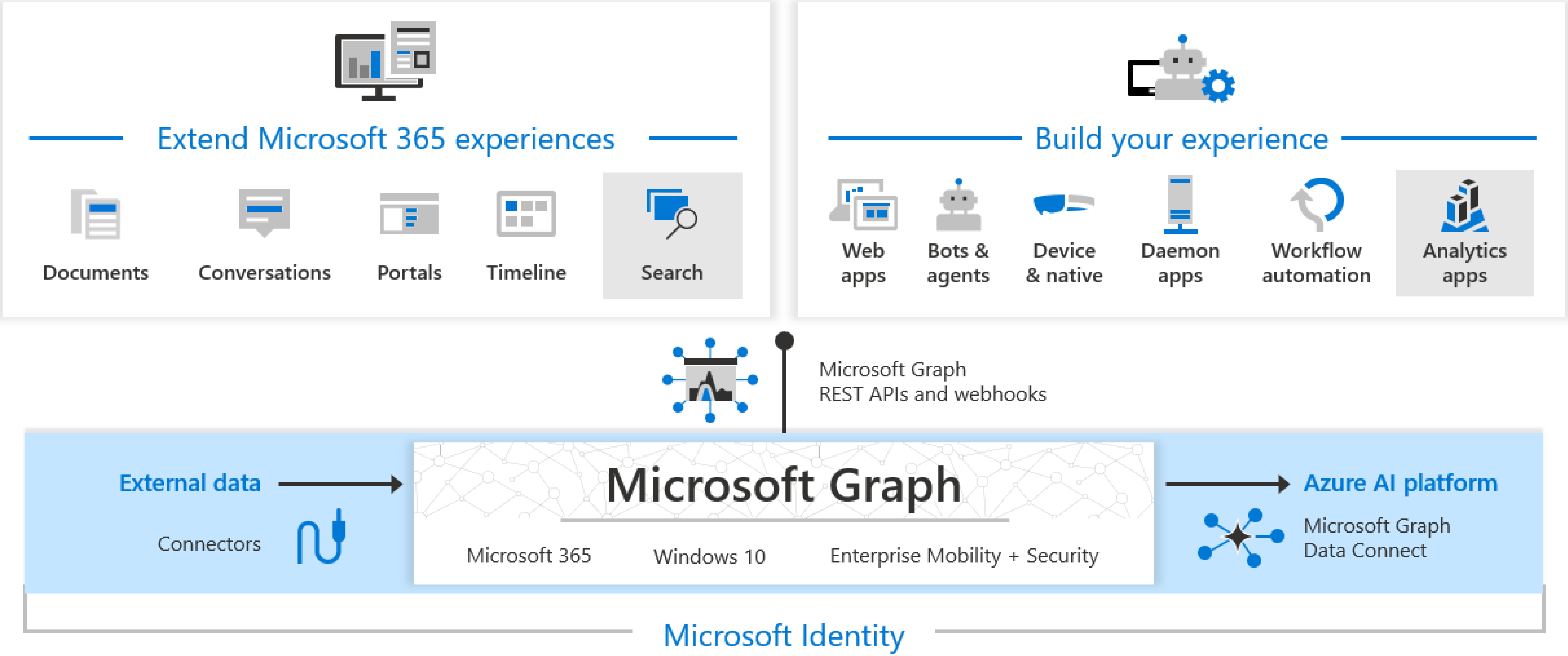
- Authenticated shell when you need it
- Needs to be deployed to an Azure subscription
- Comes preloaded with the most common tools



```
core                              2.53.0 *
telemetry                          1.1.0

Extensions:
ai-examples                        0.2.5
ml                                2.21.1
ssh                                2.0.2

Dependencies:
msal                            1.24.0b2
azure-mgmt-resource             23.1.0b2

Python location '/usr/bin/python3.9'
Extensions directory '/home/thomas/.azure/cliextensions'
Extensions system directory '/usr/lib/python3.9/site-packages/azure-cli-extensions'

Python (Linux) 3.9.14 (main, Oct 12 2023, 19:48:32)
[GCC 11.2.0]
```

Getting to know the Microsoft Graph

Microsoft 365 Platform



Extend Microsoft 365 experiences

Documents

Conversations

Portals

Timeline

Search

Build your experience

Web apps

Bots & agents

Device & native

Daemon apps

Workflow automation

Analytics apps

Microsoft Graph
REST APIs and webhooks

External data

Connectors

Microsoft Graph

Microsoft 365

Windows 10

Enterprise Mobility + Security

Azure AI platform

Microsoft Graph
Data Connect

Microsoft Identity

One API to rule them all

- Is a “RESTful API”
- HTTP / Standards based
- Has a single endpoint: graph.microsoft.com
- Works with structured data (json)

Beneath the covers

GET https://graph.microsoft.com/v1.0/users

```
{
        "businessPhones": [
        "(212) 555-8335"
        ],
        "displayName": "Aaron Painter",
        "givenName": "Aaron",
        "jobTitle": "Strategy Consulting Manager",
        "mail": "aaronp@thvo.net",
        "mobilePhone": null,
        "officeLocation": null,
        "preferredLanguage": null,
        "surname": "Painter",
        "userPrincipalName": "aaronp@thvo.net",
        "id": "676ca8a1-eaab-4e15-8ee2-72c97b53a4df"
}
```

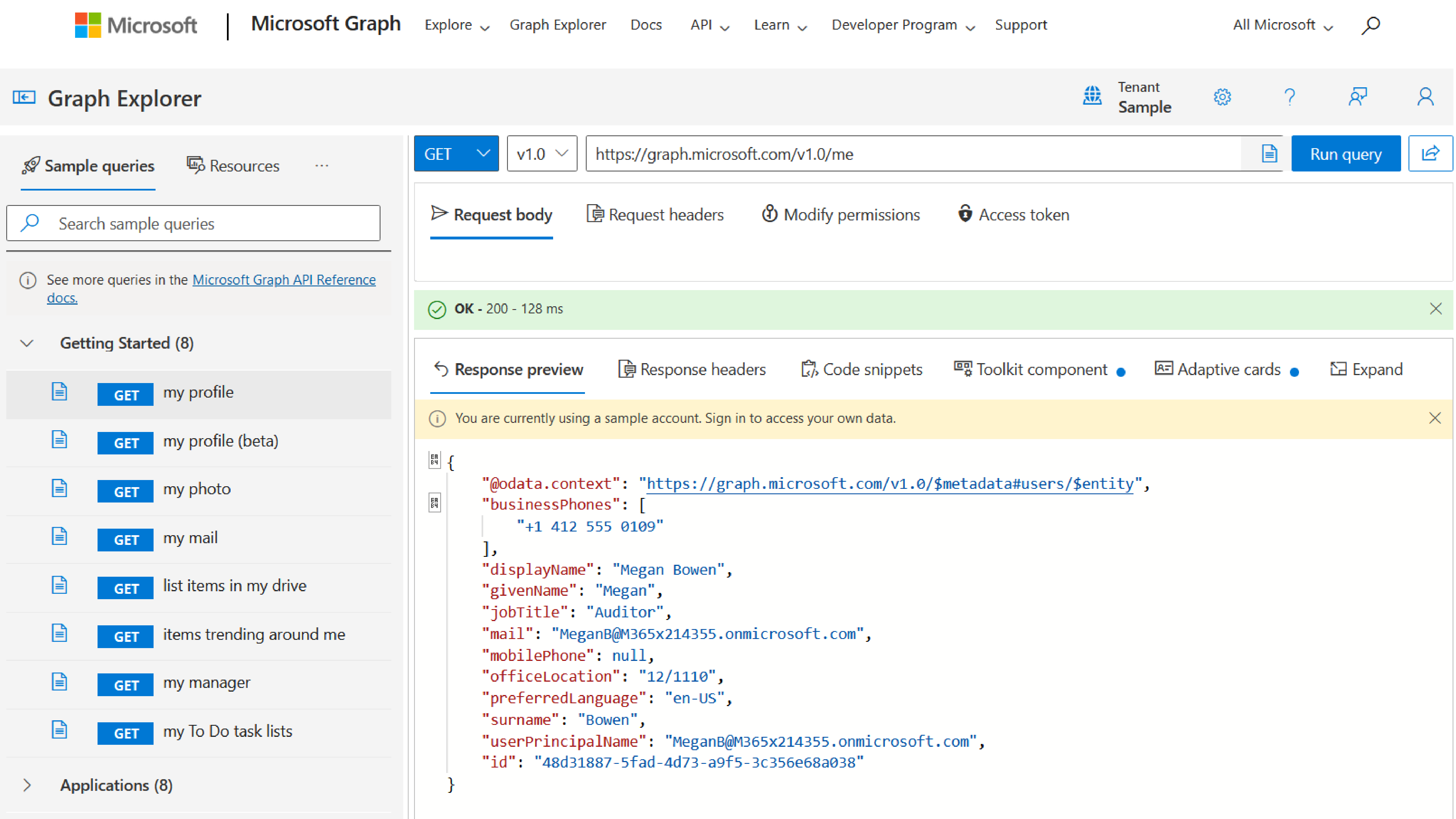
Some common operations

Operation	URL
GET my profile	https://graph.microsoft.com/v1.0/me
GET my files	https://graph.microsoft.com/v1.0/me/drive/root/children
GET my photo	https://graph.microsoft.com/v1.0/me/photo/$value
GET my mail	https://graph.microsoft.com/v1.0/me/messages
GET my high importance email	https://graph.microsoft.com/v1.0/me/messages?$filter=importance%20eq%20'high'
GET my calendar events	https://graph.microsoft.com/v1.0/me/events
GET my manager	https://graph.microsoft.com/v1.0/me/manager
GET last user to modify file foo.txt	https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt/lastModifiedByUser
GET users in my organization	https://graph.microsoft.com/v1.0/users
GET groups in my organization	https://graph.microsoft.com/v1.0/groups
GET people related to me	https://graph.microsoft.com/v1.0/me/people
GET items trending around me	https://graph.microsoft.com/beta/me/insights/trending
GET my notes	https://graph.microsoft.com/v1.0/me/onenote/notebooks

Graph Explorer

- Interactive tool to learn about the Graph
- Works with demo data **or your own tenant**
- Abstracts away a lot of the complexity
- Your first stop in getting to know the Graph

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



GET v1.0 https://graph.microsoft.com/v1.0/me

See more queries in the Microsoft Graph API Reference docs.

Getting Started (8)

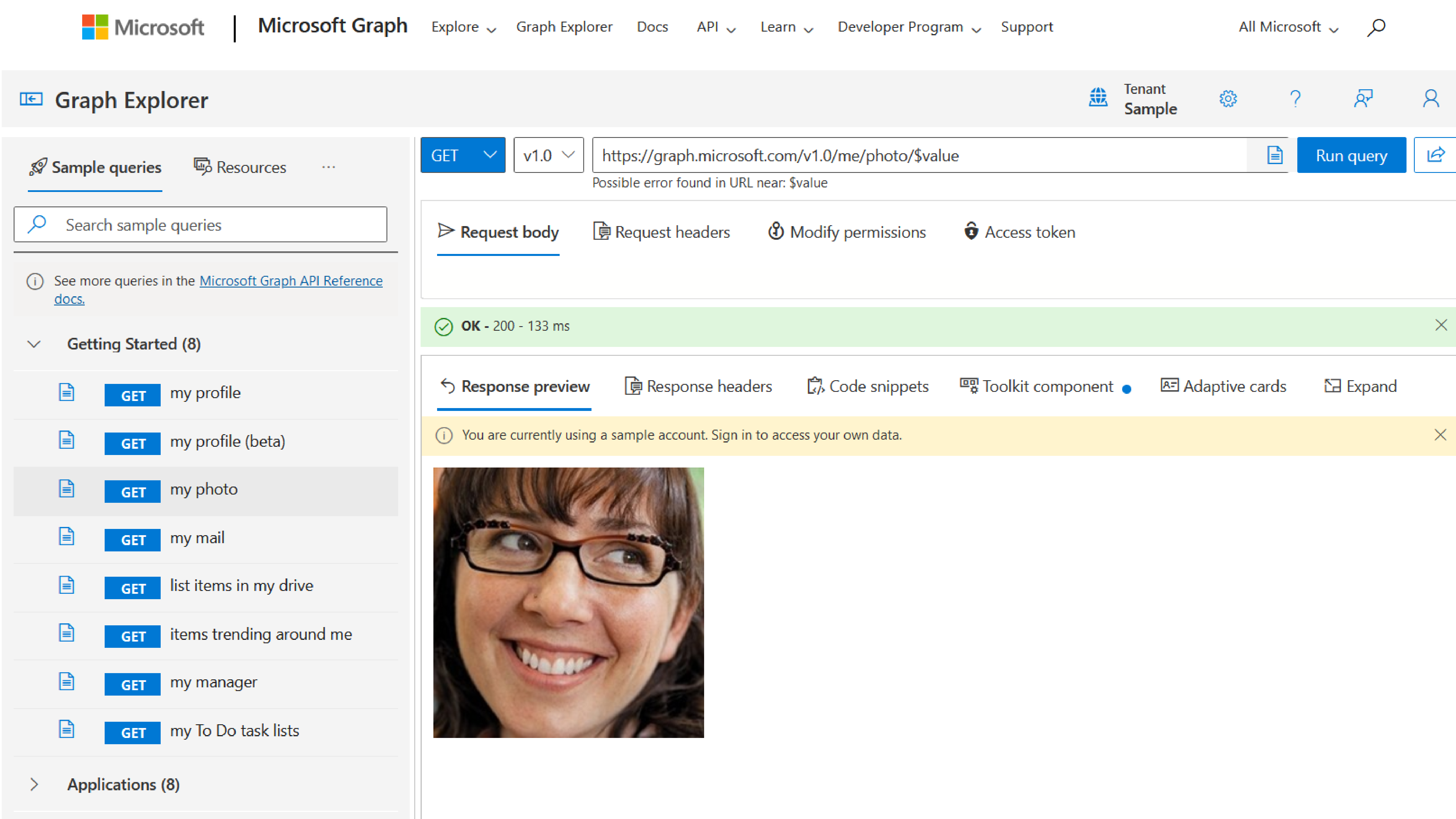
- GET my profile
- GET my profile (beta)
- GET my photo
- GET my mail
- GET list items in my drive
- GET items trending around me
- GET my manager
- GET my To Do task lists

Applications (8)

OK - 200 - 128 ms

You are currently using a sample account. Sign in to access your own data.

```
{
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",
    "businessPhones": [
        "+1 412 555 0109"
    ],
    "displayName": "Megan Bowen",
    "givenName": "Megan",
    "jobTitle": "Auditor",
    "mail": "MeganB@M365x214355.onmicrosoft.com",
    "mobilePhone": null,
    "officeLocation": "12/1110",
    "preferredLanguage": "en-US",
    "surname": "Bowen",
    "userPrincipalName": "MeganB@M365x214355.onmicrosoft.com",
    "id": "48d31887-5fad-4d73-a9f5-3c356e68a038"
}
```



GET v1.0 https://graph.microsoft.com/v1.0/me/photo/$value

Possible error found in URL near: $value

See more queries in the Microsoft Graph API Reference docs.

Getting Started (8)

GET my profile

GET my profile (beta)

GET my photo

GET my mail

GET list items in my drive

GET items trending around me

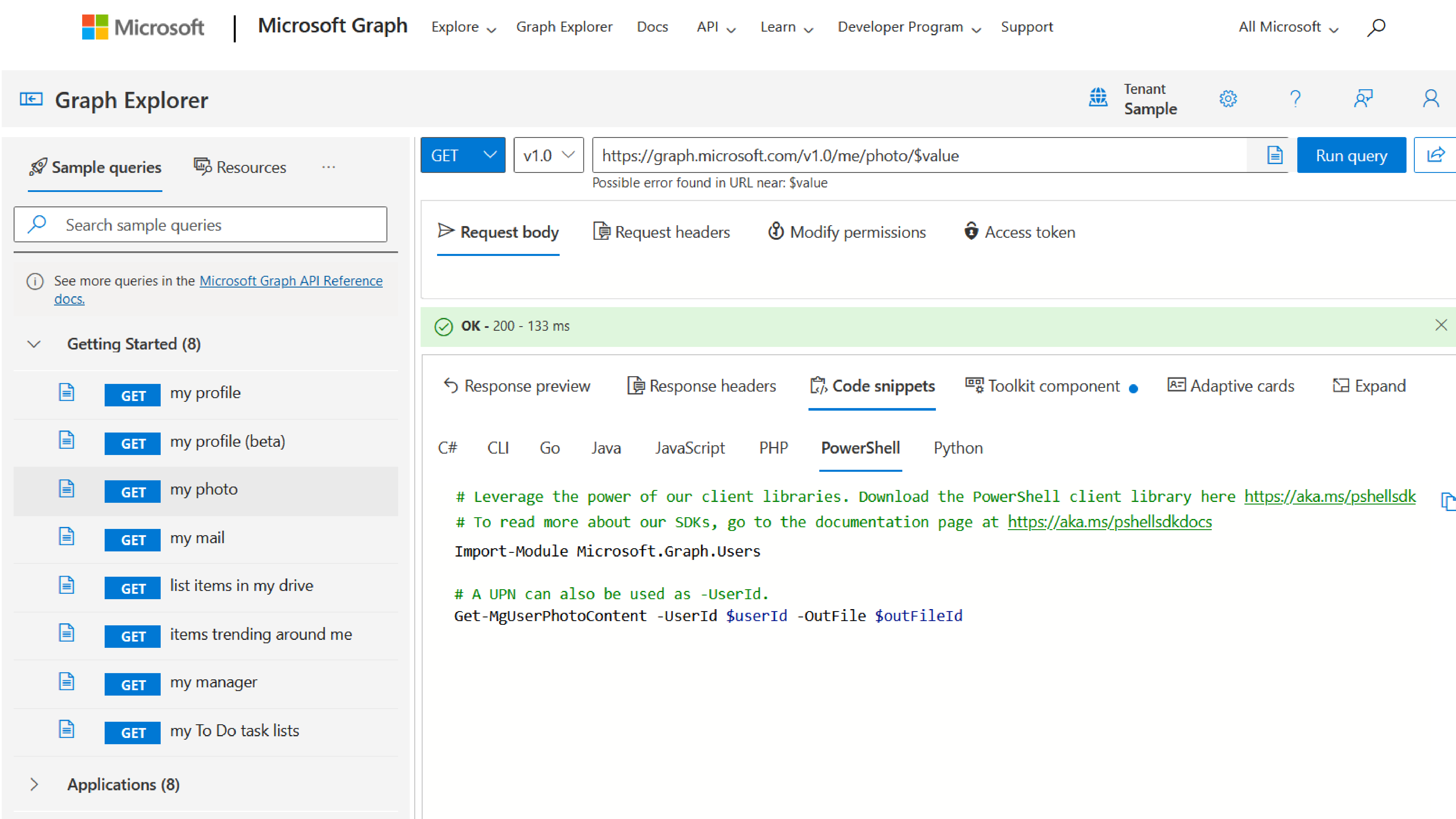
GET my manager

GET my To Do task lists

Applications (8)

OK - 200 - 133 ms

You are currently using a sample account. Sign in to access your own data.



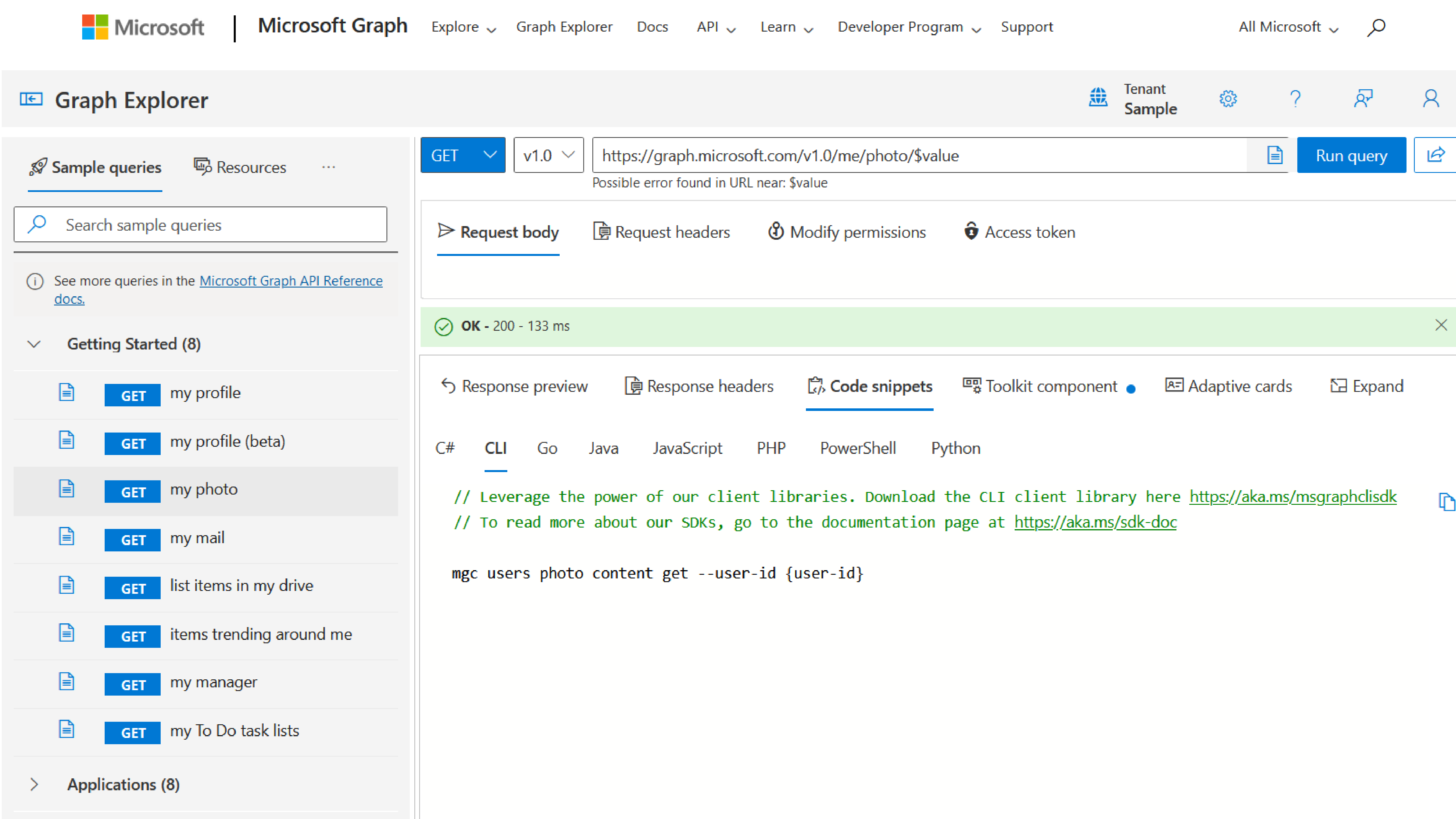
See more queries in the Microsoft Graph API Reference docs.

https://graph.microsoft.com/v1.0/me/photo/$value

Possible error found in URL near: $value

OK - 200 - 133 ms

```
# 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

# A UPN can also be used as -UserId.
Get-MgUserPhotoContent -UserId $userId -OutFile $outFileId
```



See more queries in the Microsoft Graph API Reference docs.

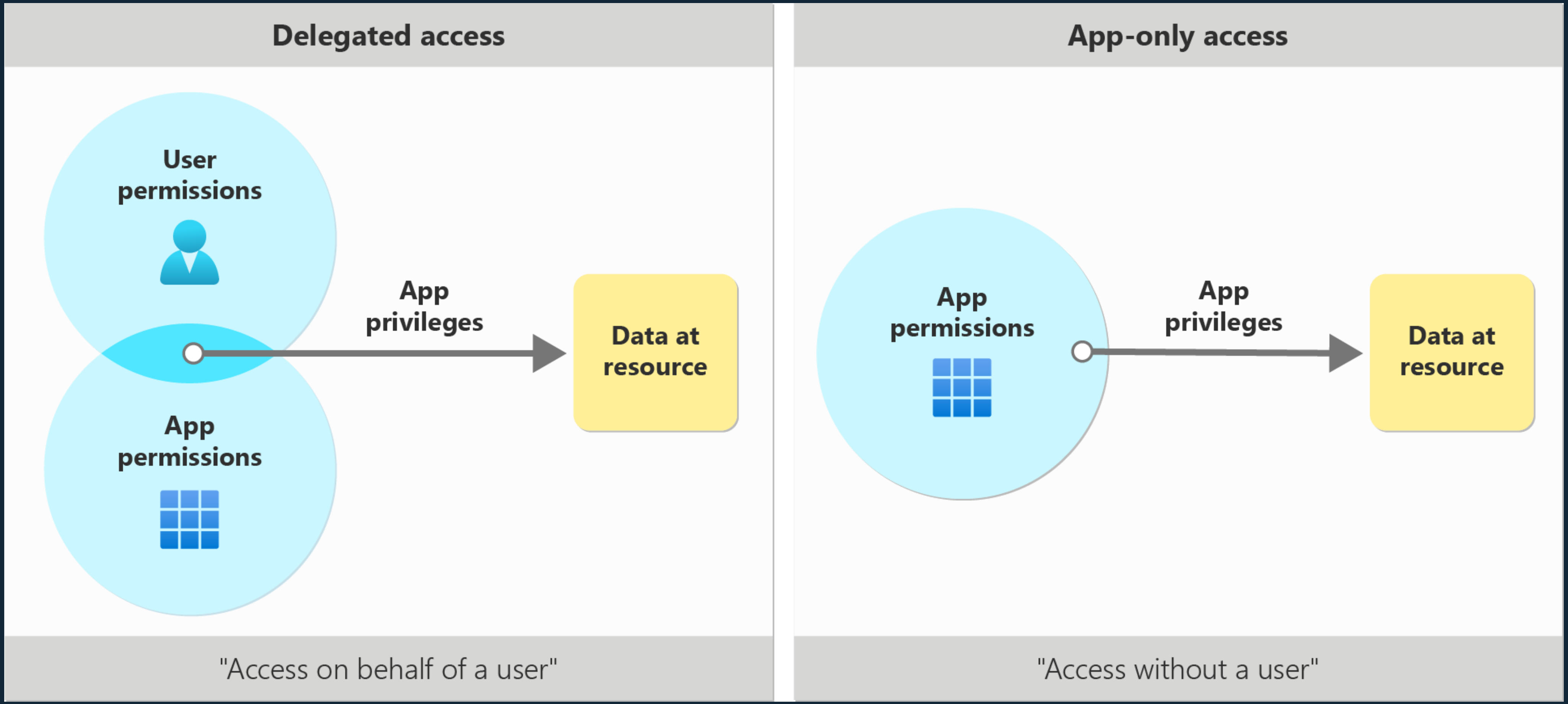
https://graph.microsoft.com/v1.0/me/photo/$value

Possible error found in URL near: $value

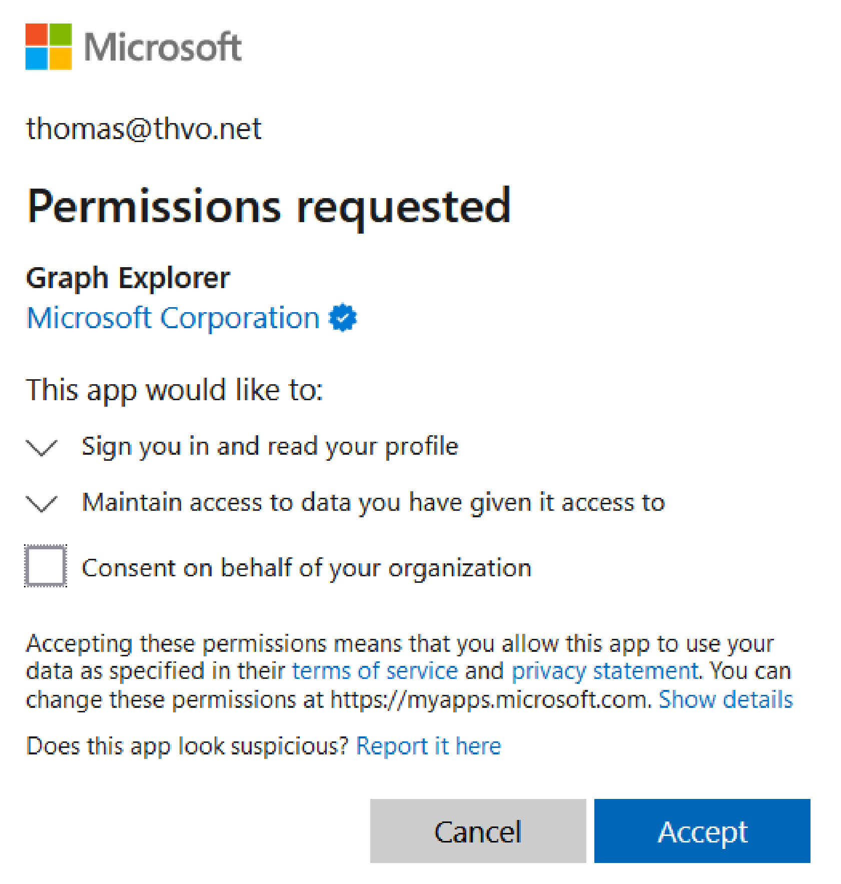
OK - 200 - 133 ms

```
// Leverage the power of our client libraries. Download the CLI client library here https://aka.ms/msgraphclisdk
// To read more about our SDKs, go to the documentation page at https://aka.ms/sdk-doc

mgc users photo content get --user-id {user-id}
```

Graph permissions model



Giving consent (delegated)



thomas@thvo.net

Permissions requested

Graph Explorer
Microsoft Corporation

This app would like to:

- Sign you in and read your profile
- Maintain access to data you have given it access to

☐ Consent on behalf of your organization

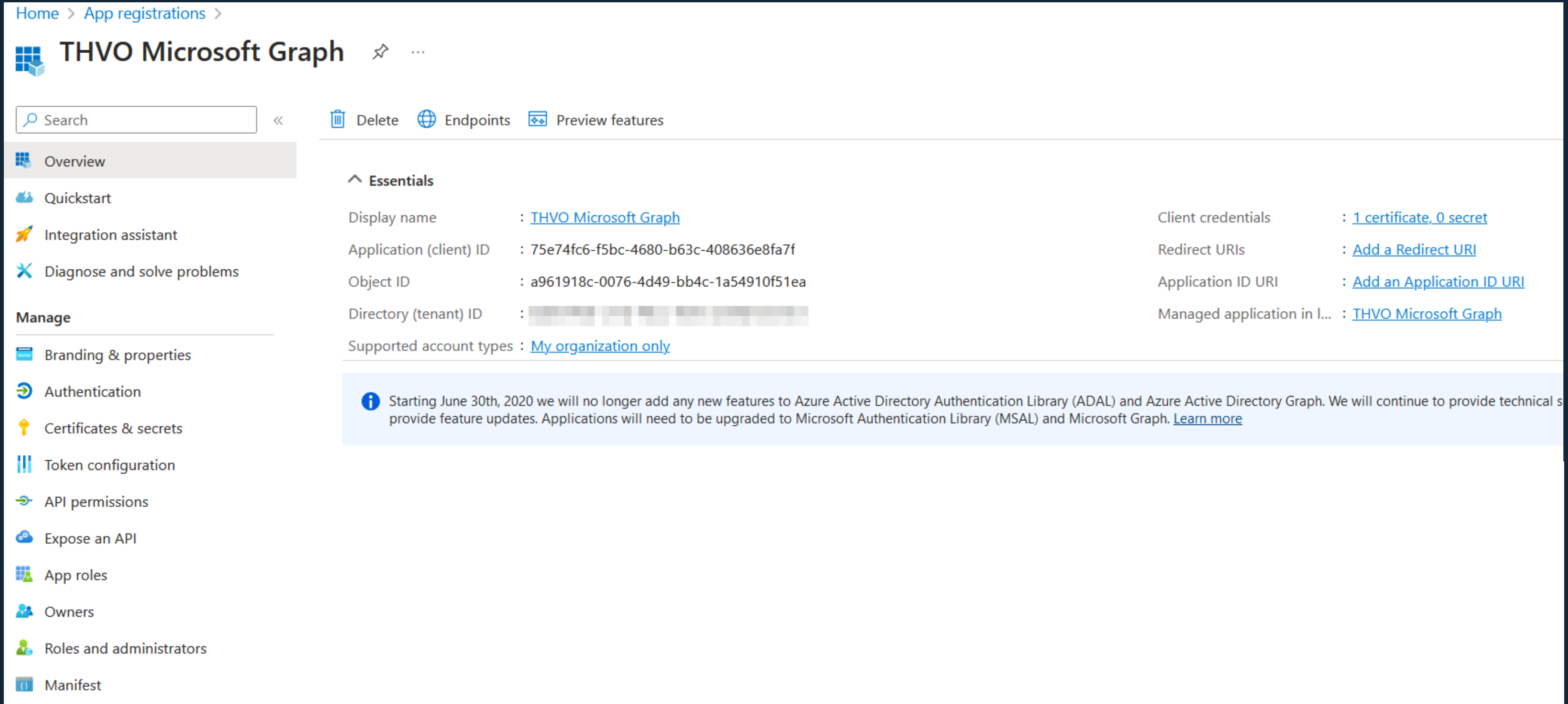
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Typical AuthN & AuthZ workflow

- Register an application in Entra ID
- Configure authentication
- Grant the necessary permissions to the app

Registering an app in Entra ID



THVO Microsoft Graph

Essentials

Display name	: THVO Microsoft Graph
Application (client) ID	: 75e74fc6-f5bc-4680-b63c-408636e8fa7f
Object ID	: a961918c-0076-4d49-bb4c-1a54910f51ea
Directory (tenant) ID	: [redacted]
Supported account types	: My organization only
Client credentials	: 1 certificate, 0 secret
Redirect URIs	: Add a Redirect URI
Application ID URI	: Add an Application ID URI
Managed application in l...	: THVO Microsoft Graph

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical s
provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

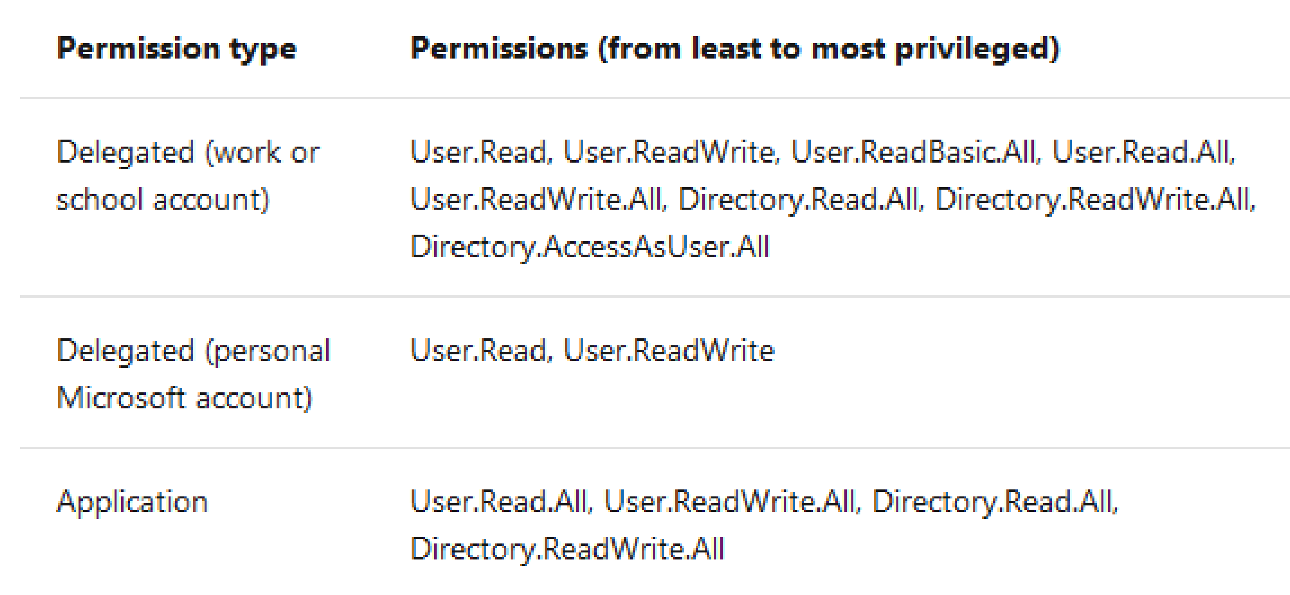
Authentication options

- User prompt / device code (interactive)
- AppId & Secret
- Certificate
- Managed Identity

Preferred

Granting permissions

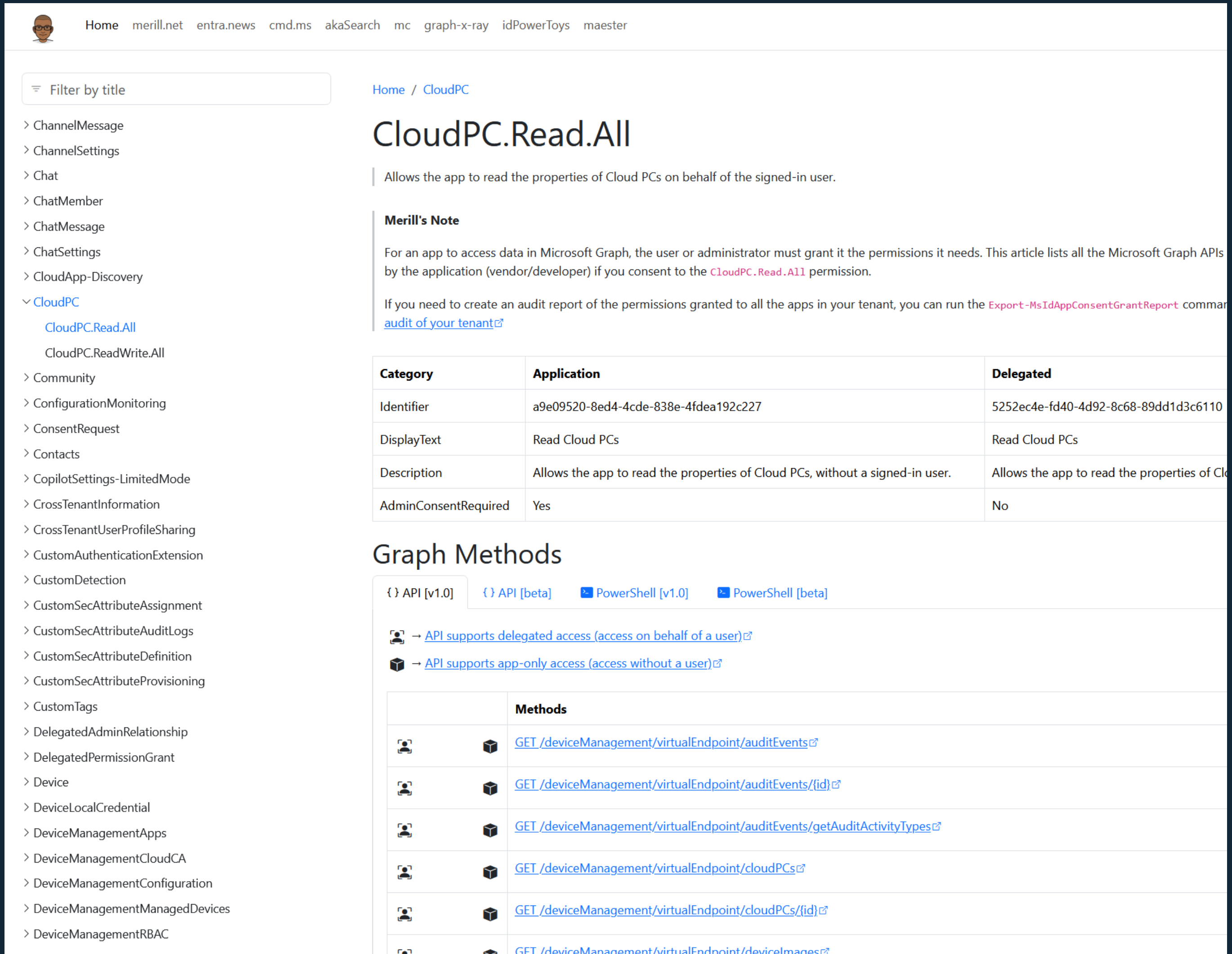
- Every operation requires specific permissions
- Don’t grant too many permissions
- Some permissions require admin consent!



Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	User.Read, User.ReadWrite, User.ReadBasic.All, User.Read.All, User.ReadWrite.All, Directory.Read.All, Directory.ReadWrite.All, Directory.AccessAsUser.All
Delegated (personal Microsoft account)	User.Read, User.ReadWrite
Application	User.Read.All, User.ReadWrite.All, Directory.Read.All, Directory.ReadWrite.All

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

Graph Permissions Explorer 💎



- ChannelMessage
- ChannelSettings
- Chat
- ChatMember
- ChatMessage
- ChatSettings
- CloudApp-Discovery
- CloudPC
  - CloudPC.Read.All
  - CloudPC.ReadWrite.All
- Community
- ConfigurationMonitoring
- ConsentRequest
- Contacts
- CopilotSettings-LimitedMode
- CrossTenantInformation
- CrossTenantUserProfileSharing
- CustomAuthenticationExtension
- CustomDetection
- CustomSecAttributeAssignment
- CustomSecAttributeAuditLogs
- CustomSecAttributeDefinition
- CustomSecAttributeProvisioning
- CustomTags
- DelegatedAdminRelationship
- DelegatedPermissionGrant
- Device
- DeviceLocalCredential
- DeviceManagementApps
- DeviceManagementCloudCA
- DeviceManagementConfiguration
- DeviceManagementManagedDevices
- DeviceManagementRBAC

Home / CloudPC

CloudPC.Read.All

> Allows the app to read the properties of Cloud PCs on behalf of the signed-in user.

> **Merill's Note**
>
> For an app to access data in Microsoft Graph, the user or administrator must grant it the permissions it needs. This article lists all the Microsoft Graph APIs by the application (vendor/developer) if you consent to the `CloudPC.Read.All` permission.
>
> If you need to create an audit report of the permissions granted to all the apps in your tenant, you can run the `Export-MsIdAppConsentGrantReport` commar audit of your tenant

Category	Application	Delegated
Identifier	a9e09520-8ed4-4cde-838e-4fdea192c227	5252ec4e-fd40-4d92-8c68-89dd1d3c6110
DisplayText	Read Cloud PCs	Read Cloud PCs
Description	Allows the app to read the properties of Cloud PCs, without a signed-in user.	Allows the app to read the properties of Cl
AdminConsentRequired	Yes	No

Graph Methods

👤 → API supports delegated access (access on behalf of a user)

📦 → API supports app-only access (access without a user)

		Methods
👤	📦	GET /deviceManagement/virtualEndpoint/auditEvents
👤	📦	GET /deviceManagement/virtualEndpoint/auditEvents/{id}
👤	📦	GET /deviceManagement/virtualEndpoint/auditEvents/getAuditActivityTypes
👤	📦	GET /deviceManagement/virtualEndpoint/cloudPCs
👤	📦	GET /deviceManagement/virtualEndpoint/cloudPCs/{id}
👤	📦	GET /deviceManagement/virtualEndpoint/deviceImages

https://graphpermissions.merill.net/permission/

PowerShell to the rescue

So many PowerShell modules

- Teams
- SharePoint Online
- Exchange
- AZ
- PnP
- M365 DSC
- …

Microsoft Graph SDK for PowerShell

One module to rule them all

Cross-platform

Modern Authentication

Fine-grained privilege control

The possibilities are endless...

- Onboarding users
- Working with Excel data
- Finding meeting times
- Converting documents
- Managing employee profiles
- Keeping email data in sync
- Correlating security alerts
- …

“When a user leaves, query their OneDrive with the Graph to let users know about shared documents they will lose access to”

PowerShell SDK

- `Install-Module Microsoft.Graph`
- Verify: `Get-InstalledModule Microsoft.Graph`

Prereqs:

- PowerShell 5.1 or later
- .NET Framework 4.7.2 or later
- `Install-Module PowerShellGet -Force`

Navigating the SDK

```
Get-Command -Module Microsoft.Graph* *team*
Find-MgGraphCommand -Uri '/users/{id}'
```

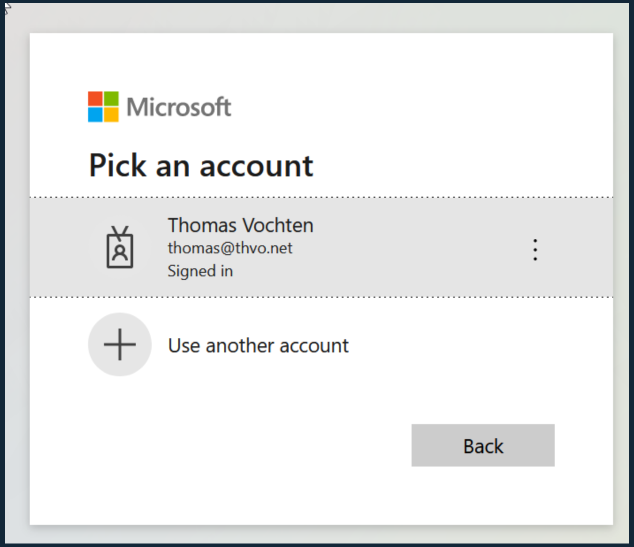
Prefix is always “Mg”

Verbs in HTTP vs PowerShell

• GET	Get-Mg…
• POST	New-Mg…
• PUT	New-Mg…
• PATCH	Update-Mg…
• DELETE	Remove-Mg…

Connecting to the Graph

```
Connect-MgGraph -Scopes
"User.Read.All","Group.ReadWrite.All“

(Get-MgContext).Scopes
```

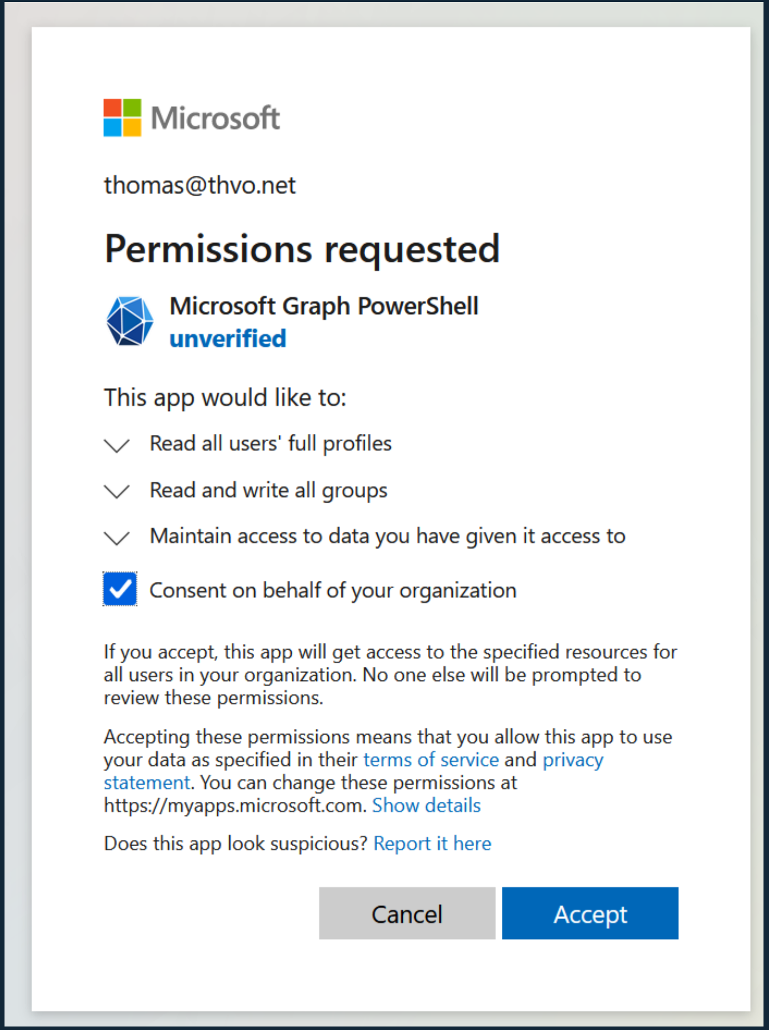


Microsoft

Pick an account

Thomas Vochten
thomas@thvo.net
Signed in

Use another account



Microsoft

thomas@thvo.net

Permissions requested

Microsoft Graph PowerShell
unverified

This app would like to:

- Read all users' full profiles
- Read and write all groups
- Maintain access to data you have given it access to

☑ Consent on behalf of your organization

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

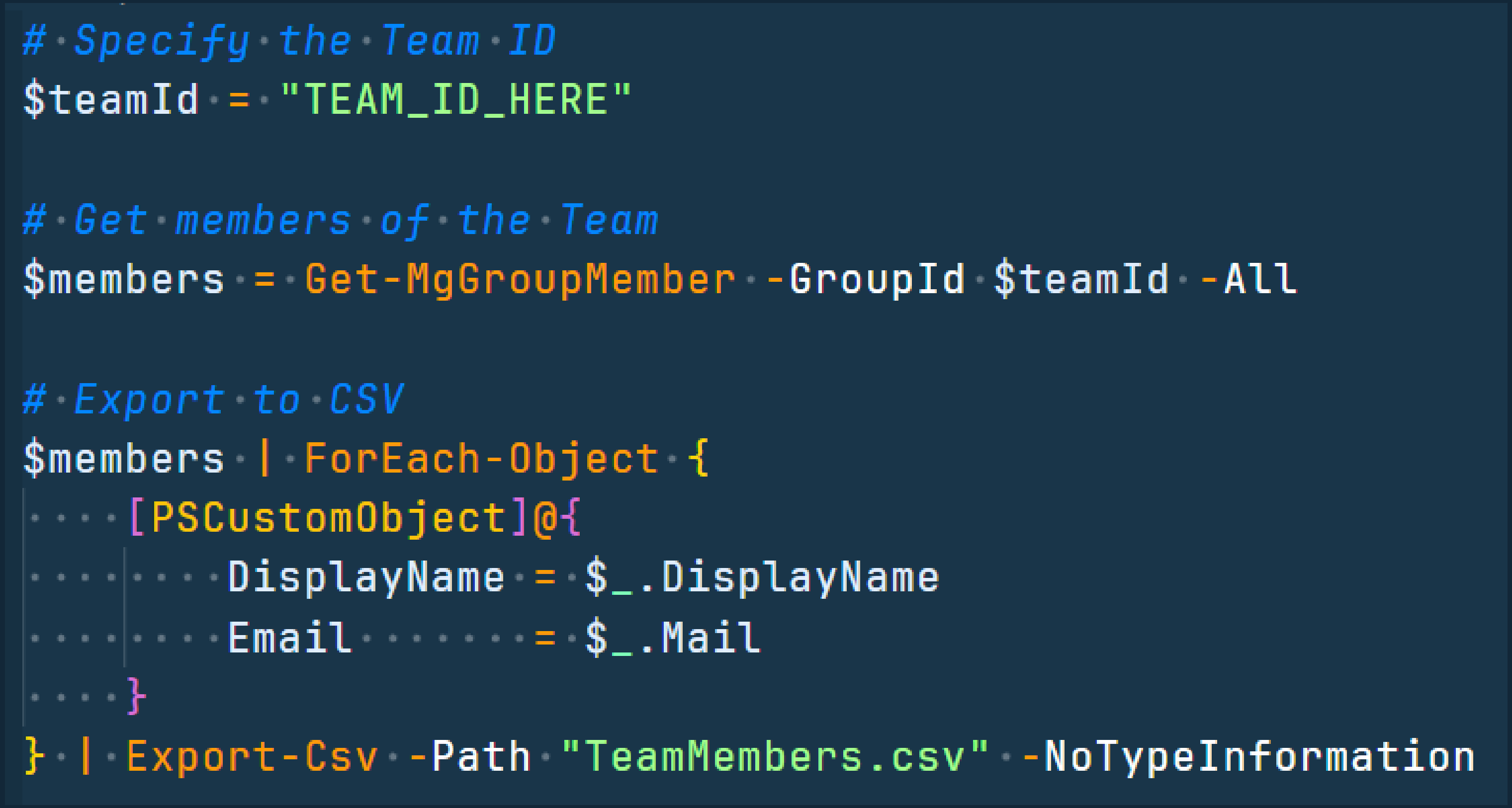
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Does this app look suspicious? Report it here

Basic examples

```
Get-MgUser
New-MgTeam
Update-MgTeamChannel
Remove-MgGroup
```

Example: exporting team members



```powershell
# Specify the Team ID
$teamId = "TEAM_ID_HERE"

# Get members of the Team
$members = Get-MgGroupMember -GroupId $teamId -All

# Export to CSV
$members | ForEach-Object {
    [PSCustomObject]@{
        DisplayName = $_.DisplayName
        Email       = $_.Mail
    }
} | Export-Csv -Path "TeamMembers.csv" -NoTypeInformation
```

Using a certificate or managed identity 🔐

```
Connect-MgGraph
   -ClientID YOUR_APP_ID
   -TenantId YOUR_TENANT_ID
   -CertificateName YOUR_CERT_SUBJECT

Connect-MgGraph -Identity


Get-MgContext
```

No cmdlet that suits your needs?

```
Invoke-MGGraphRequest
```

```
Invoke-MgGraphRequest -Method GET
https://graph.microsoft.com/v1.0/me
```

The rise of the CLI

The Microsoft 365 CLI

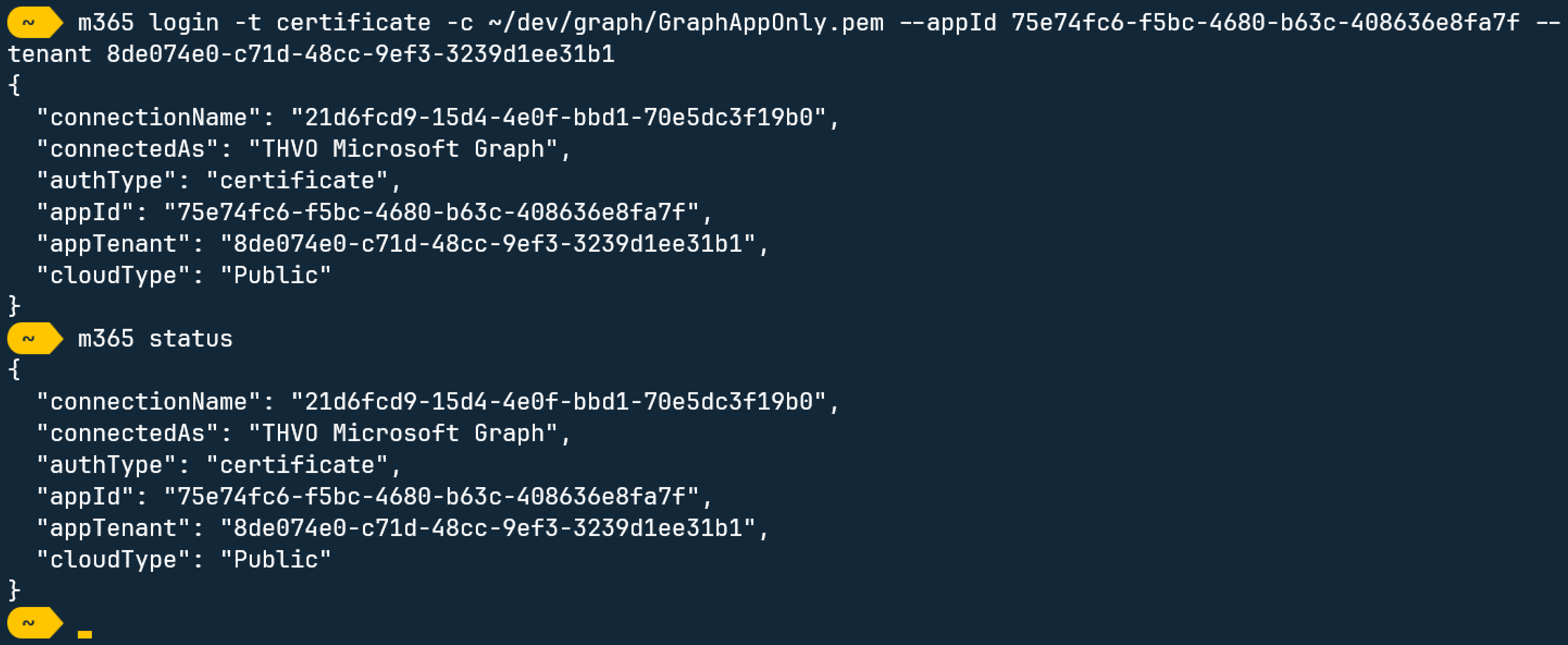
- Community-driven, by the PnP team
- What can I use it for?
- Prerequisites: node.js (comes as a npm package)

```
npm install -g @pnp/cli-microsoft365
```

- Cross-platform
- Can be run inside a Docker container
- AuthN & AuthZ also through app registration

https://pnp.github.io/cli-microsoft365

Logging in with the m365 CLI

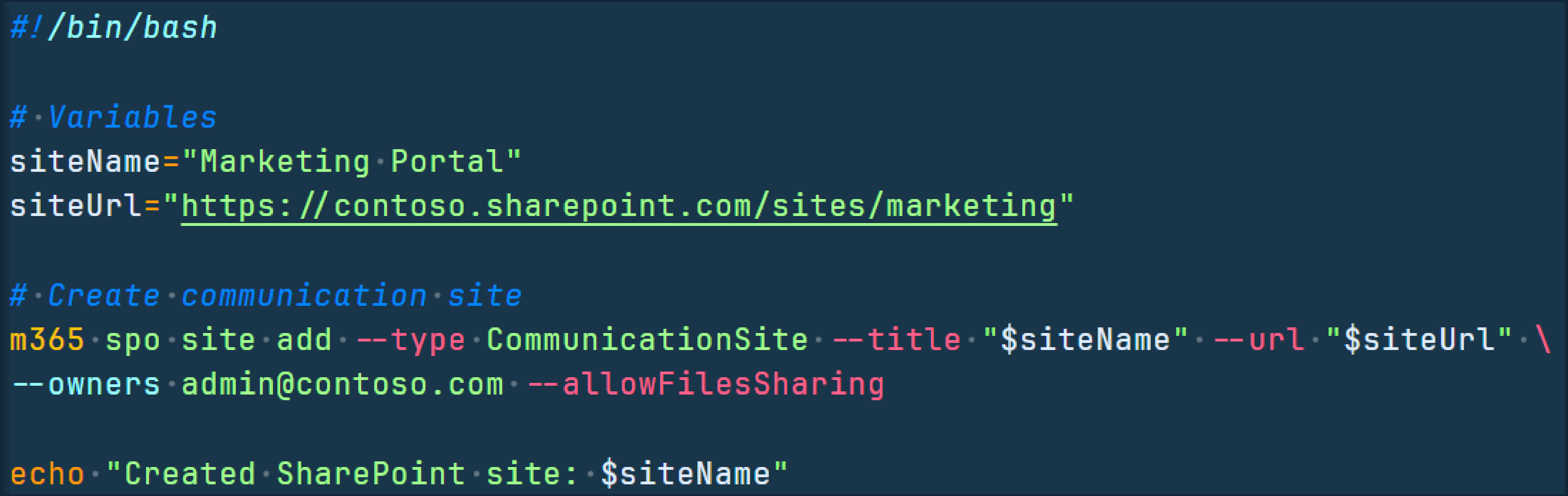


```
~  m365 login -t certificate -c ~/dev/graph/GraphAppOnly.pem --appId 75e74fc6-f5bc-4680-b63c-408636e8fa7f --
tenant 8de074e0-c71d-48cc-9ef3-3239d1ee31b1
{
  "connectionName": "21d6fcd9-15d4-4e0f-bbd1-70e5dc3f19b0",
  "connectedAs": "THVO Microsoft Graph",
  "authType": "certificate",
  "appId": "75e74fc6-f5bc-4680-b63c-408636e8fa7f",
  "appTenant": "8de074e0-c71d-48cc-9ef3-3239d1ee31b1",
  "cloudType": "Public"
}
~  m365 status
{
  "connectionName": "21d6fcd9-15d4-4e0f-bbd1-70e5dc3f19b0",
  "connectedAs": "THVO Microsoft Graph",
  "authType": "certificate",
  "appId": "75e74fc6-f5bc-4680-b63c-408636e8fa7f",
  "appTenant": "8de074e0-c71d-48cc-9ef3-3239d1ee31b1",
  "cloudType": "Public"
}
~  _
```

Getting help with the m365 CLI

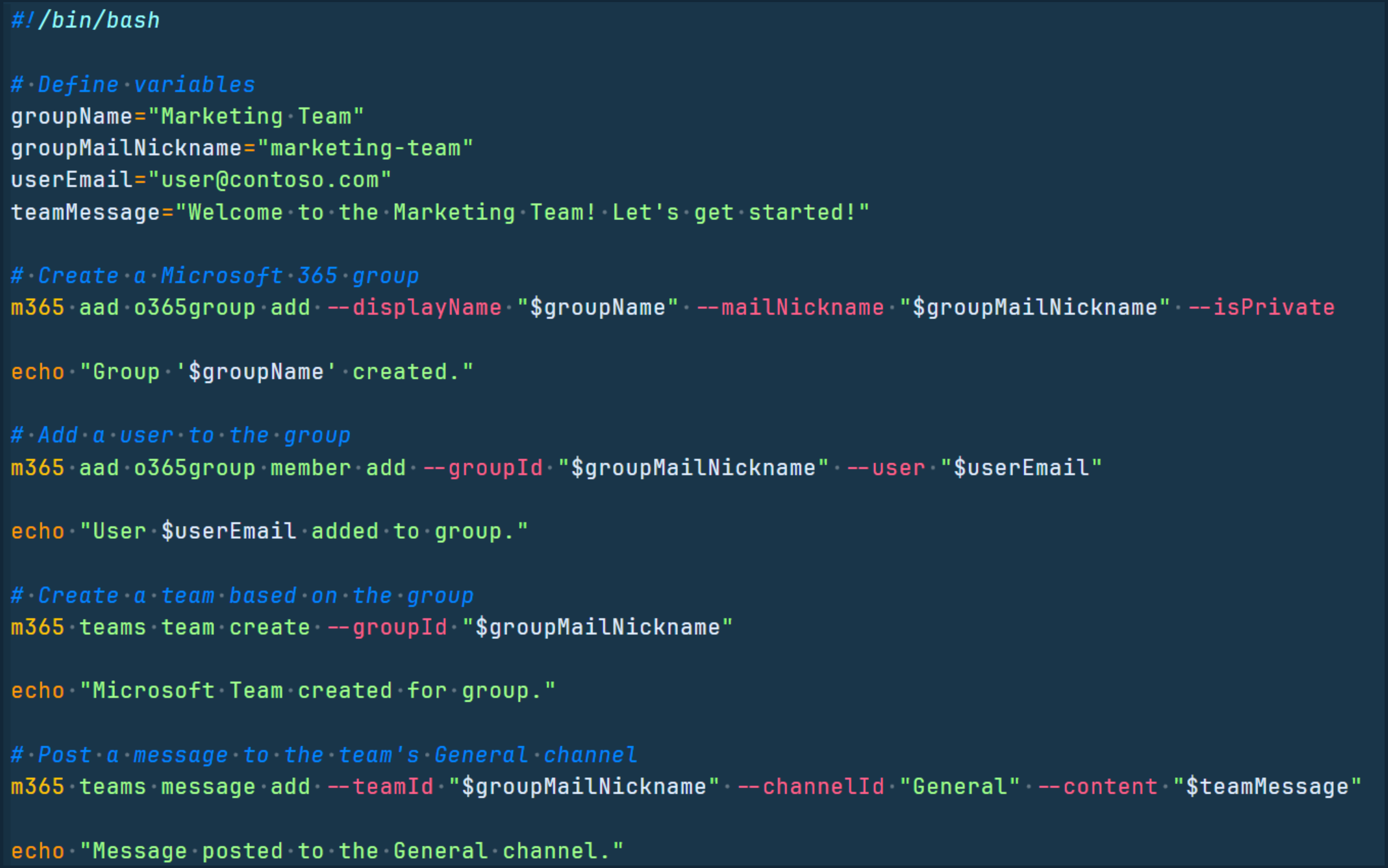
```
~ m365 help planner

CLI for Microsoft 365 v10.1.0
Manage Microsoft 365 and SharePoint Framework projects on any platform

Commands groups:

  planner bucket *  5 commands
  planner plan *    5 commands
  planner roster *  8 commands
  planner task *    11 commands
  planner tenant *  2 commands
```

Example: SharePoint site creation



```bash
#!/bin/bash

# Variables
siteName="Marketing Portal"
siteUrl="https://contoso.sharepoint.com/sites/marketing"

# Create communication site
m365 spo site add --type CommunicationSite --title "$siteName" --url "$siteUrl" \
--owners admin@contoso.com --allowFilesSharing

echo "Created SharePoint site: $siteName"
```

Example: group creation



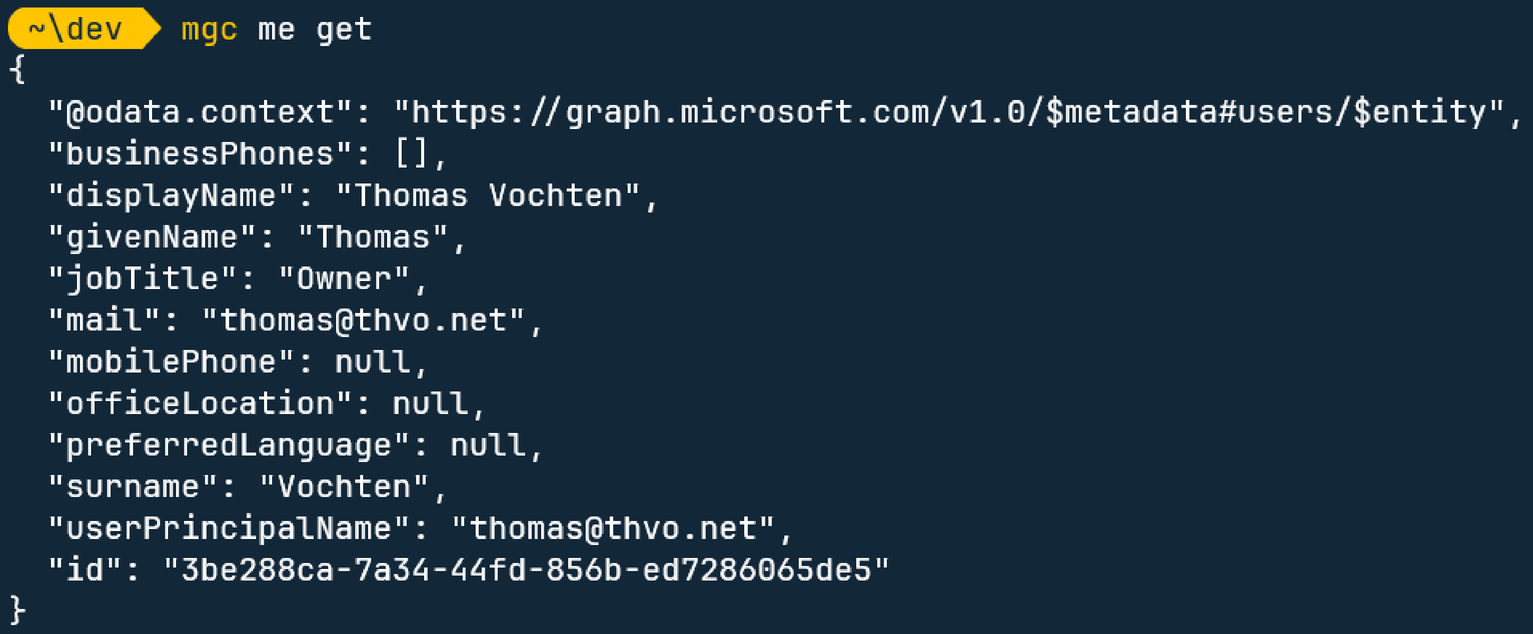
```bash
#!/bin/bash

# Define variables
groupName="Marketing Team"
groupMailNickname="marketing-team"
userEmail="user@contoso.com"
teamMessage="Welcome to the Marketing Team! Let's get started!"

# Create a Microsoft 365 group
m365 aad o365group add --displayName "$groupName" --mailNickname "$groupMailNickname" --isPrivate

echo "Group '$groupName' created."

# Add a user to the group
m365 aad o365group member add --groupId "$groupMailNickname" --user "$userEmail"

echo "User $userEmail added to group."

# Create a team based on the group
m365 teams team create --groupId "$groupMailNickname"

echo "Microsoft Team created for group."

# Post a message to the team's General channel
m365 teams message add --teamId "$groupMailNickname" --channelId "General" --content "$teamMessage"

echo "Message posted to the General channel."
```

The MS Graph CLI

- Official Microsoft CLI
- Manual installation for now
- Supports native Windows (no nodejs required)
- What can I use it for? Why is it there?
- Prerequisites
- AuthN & AuthZ as any Graph application

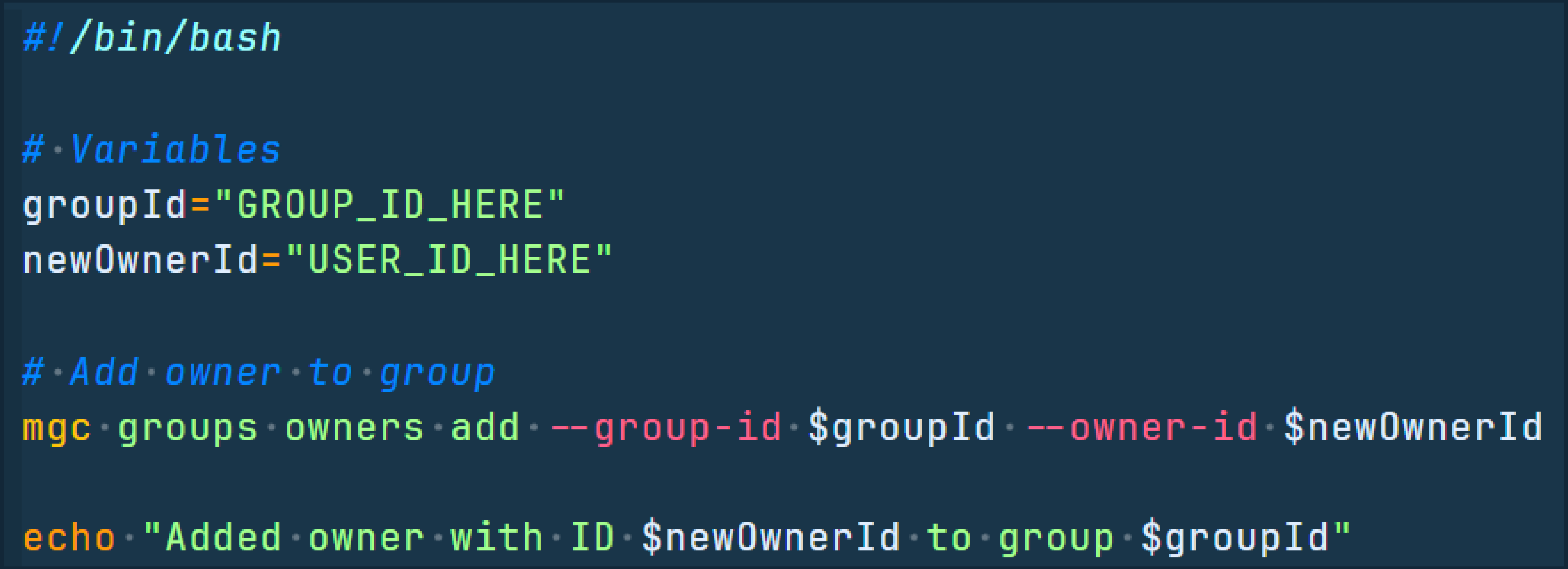
https://learn.microsoft.com/en-us/graph/cli/overview

The MS Graph CLI



```
~\dev mgc me get
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",
  "businessPhones": [],
  "displayName": "Thomas Vochten",
  "givenName": "Thomas",
  "jobTitle": "Owner",
  "mail": "thomas@thvo.net",
  "mobilePhone": null,
  "officeLocation": null,
  "preferredLanguage": null,
  "surname": "Vochten",
  "userPrincipalName": "thomas@thvo.net",
  "id": "3be288ca-7a34-44fd-856b-ed7286065de5"
}
```

Example: adding an owner to a group



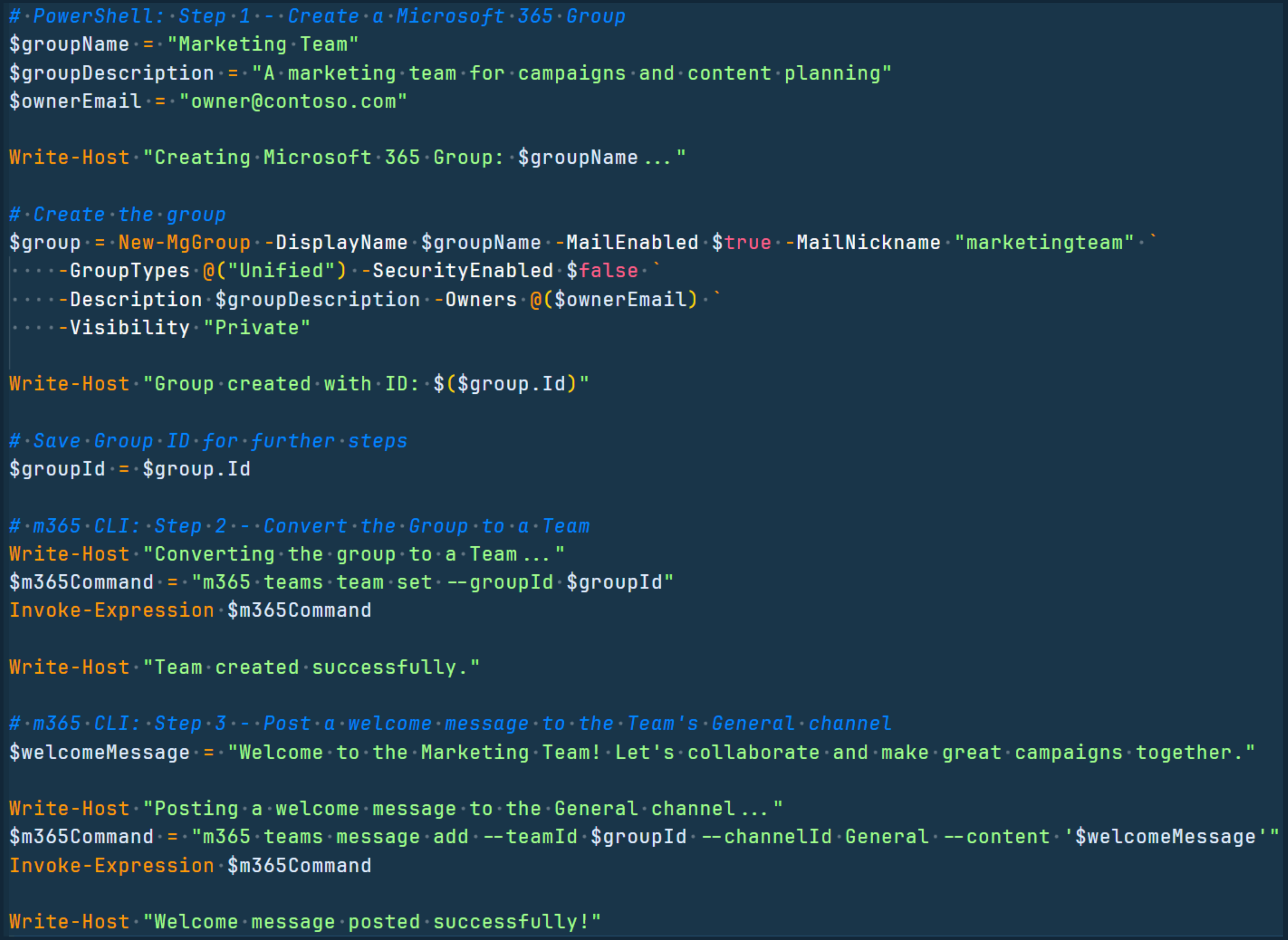
```bash
#!/bin/bash

# Variables
groupId="GROUP_ID_HERE"
newOwnerId="USER_ID_HERE"

# Add owner to group
mgc groups owners add --group-id $groupId --owner-id $newOwnerId

echo "Added owner with ID $newOwnerId to group $groupId"
```

Integration & automation

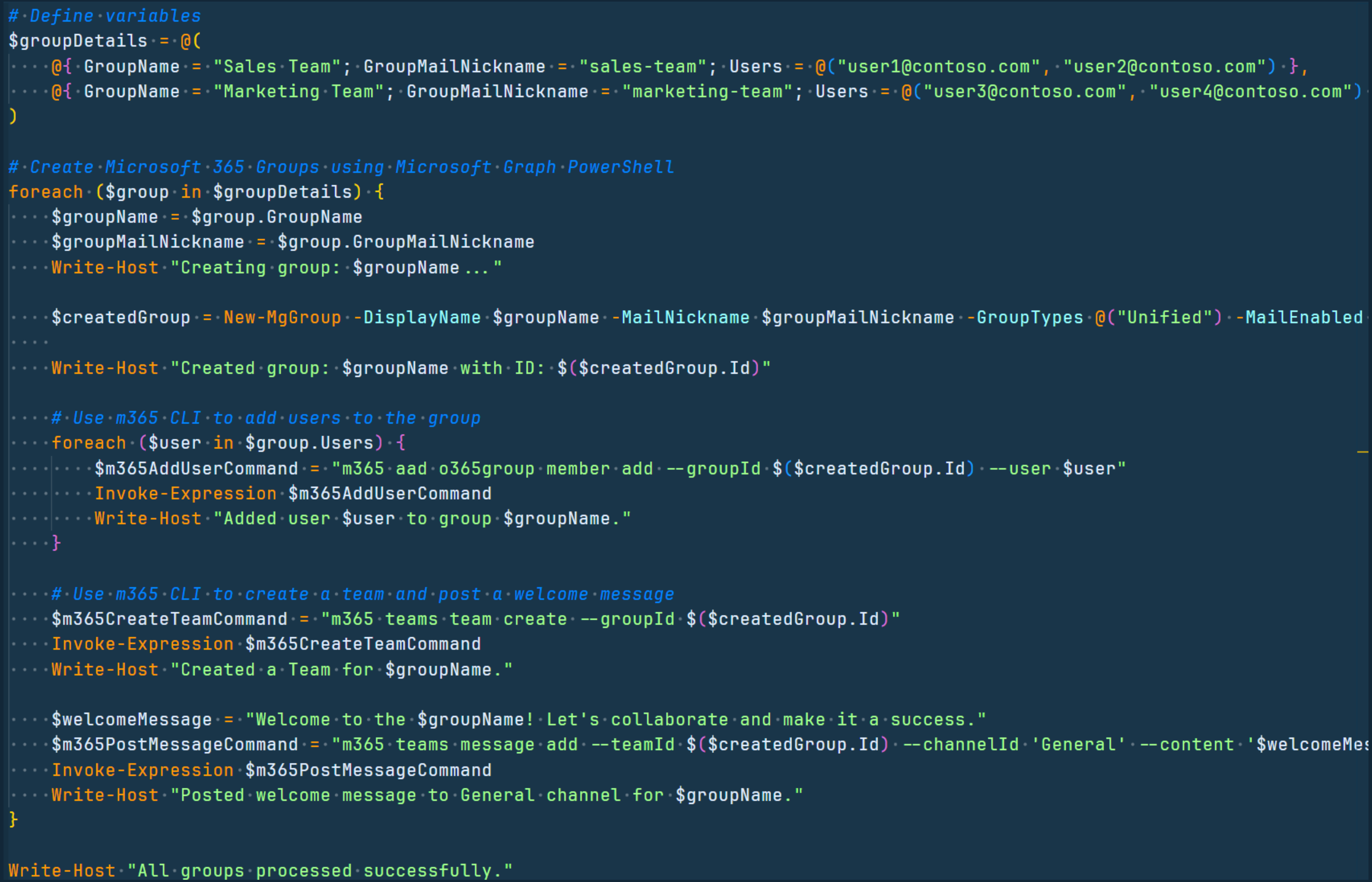
Better together!

Use PowerShell as the orchestrator

Use a CLI when needed



```powershell
# PowerShell: Step 1 - Create a Microsoft 365 Group
$groupName = "Marketing Team"
$groupDescription = "A marketing team for campaigns and content planning"
$ownerEmail = "owner@contoso.com"

Write-Host "Creating Microsoft 365 Group: $groupName..."

# Create the group
$group = New-MgGroup -DisplayName $groupName -MailEnabled $true -MailNickname "marketingteam" `
    -GroupTypes @("Unified") -SecurityEnabled $false `
    -Description $groupDescription -Owners @($ownerEmail) `
    -Visibility "Private"

Write-Host "Group created with ID: $($group.Id)"

# Save Group ID for further steps
$groupId = $group.Id

# m365 CLI: Step 2 - Convert the Group to a Team
Write-Host "Converting the group to a Team..."
$m365Command = "m365 teams team set --groupId $groupId"
Invoke-Expression $m365Command

Write-Host "Team created successfully."

# m365 CLI: Step 3 - Post a welcome message to the Team's General channel
$welcomeMessage = "Welcome to the Marketing Team! Let's collaborate and make great campaigns together."

Write-Host "Posting a welcome message to the General channel..."
$m365Command = "m365 teams message add --teamId $groupId --channelId General --content '$welcomeMessage'"
Invoke-Expression $m365Command

Write-Host "Welcome message posted successfully!"
```



```
# Define variables
$groupDetails = @(
    @{ GroupName = "Sales Team"; GroupMailNickname = "sales-team"; Users = @("user1@contoso.com", "user2@contoso.com") },
    @{ GroupName = "Marketing Team"; GroupMailNickname = "marketing-team"; Users = @("user3@contoso.com", "user4@contoso.com")
)

# Create Microsoft 365 Groups using Microsoft Graph PowerShell
foreach ($group in $groupDetails) {
    $groupName = $group.GroupName
    $groupMailNickname = $group.GroupMailNickname
    Write-Host "Creating group: $groupName ..."

    $createdGroup = New-MgGroup -DisplayName $groupName -MailNickname $groupMailNickname -GroupTypes @("Unified") -MailEnabled
    
    Write-Host "Created group: $groupName with ID: $($createdGroup.Id)"

    # Use m365 CLI to add users to the group
    foreach ($user in $group.Users) {
        $m365AddUserCommand = "m365 aad o365group member add --groupId $($createdGroup.Id) --user $user"
        Invoke-Expression $m365AddUserCommand
        Write-Host "Added user $user to group $groupName."
    }

    # Use m365 CLI to create a team and post a welcome message
    $m365CreateTeamCommand = "m365 teams team create --groupId $($createdGroup.Id)"
    Invoke-Expression $m365CreateTeamCommand
    Write-Host "Created a Team for $groupName."

    $welcomeMessage = "Welcome to the $groupName! Let's collaborate and make it a success."
    $m365PostMessageCommand = "m365 teams message add --teamId $($createdGroup.Id) --channelId 'General' --content '$welcomeMes
    Invoke-Expression $m365PostMessageCommand
    Write-Host "Posted welcome message to General channel for $groupName."
}

Write-Host "All groups processed successfully."
```

Automate what exactly?

- License management
- Reporting
- Auditing
- Monitoring
- Change management & tracking
- …

Choices

- Scheduled Tasks
- Azure Webjobs
- Azure Automation
- Azure Functions

- Azure Logic Apps
- Power Automate
- Azure DevOps / GitHub Actions
- Azure batch

Automating PowerShell

Azure Automation

Azure Functions

When to use what

When to use what

It depends 😳

- On the capabilities provided
- On the type of automation needed
- On what you’re used to

Comparison

Feature/Capability	Graph SDK (PowerShell)	m365 CLI	Graph CLI	Native Modules
Platform Support	Cross-platform	Cross-platform	Cross-platform	Windows-only
Ease of Use	Familiar PowerShell syntax	Simple commands	JSON-based API focus	Easy for specific tasks
Best for Admins	Yes	Yes	Limited	Yes
Best for Developers	High	High	High	Low
Automation Integration	Strong	Strong	Moderate	Moderate
Access to Graph API	Simplified	Limited	Full	No

Useful resources

- 🌐 Microsoft Graph docs / Microsoft Graph Explorer
- 🌐 Microsoft Graph PowerShell docs
- 🌐 PowerShell Module Browser
- 🌐 CLI for Microsoft 365
- 🌐 PnP PowerShell
- 🌐 Practical 365
- 🌐 Planet PowerShell

- 📗 Automating Microsoft 365 with PowerShell
- 📗 Office 365 for IT Pros

Takeaways

- Get to know the Microsoft Graph, even if you’re not a developer.
- Graph PowerShell is probably the easiest way to get started
- The true power of these tools comes from using them together
- Discover automation possibilities, but use the simplest tool that could possibly work

Thank you

🦋 thomasvochten.com

✉ mail@thomasvochten.com

Get the slides