

Using the Microsoft Graph for the reluctant IT Pro

@thomasvochten

About me



Thomas Vochten

Technology Evangelist     
#Microsoft365 #Azure #CommunityRocks

@thomasvochten

<https://thomasvochten.com>

mail@thomasvochten.com



Microsoft®
Most Valuable
Professional

BIWUG




Agenda

1. Introduction to the Microsoft Graph
2. Authentication & Authorization
3. PowerShell to the rescue!



Introduction to the Microsoft Graph

@thomasvochten

A scene featuring two puppets of Albert Einstein sitting at a table. The puppet on the left is leaning forward with his eyes closed, while the puppet on the right looks on with a wide-eyed, slightly mischievous expression. The background is dark and out of focus, suggesting an indoor setting with red curtains or walls.

Why are we even here?

The promise of the Graph

Microsoft Graph API is the gateway for



Azure AD



Excel



Intune



Outlook



OneDrive



OneNote



SharePoint



Planner

- Rich content
- Deep insights
- Real-time updates
- Extensible

Some use cases

- 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”

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

So many questions... 🙄

- That still sounds like developer stuff to me!
- Do I have to befriend `Invoke-RestMethod`?
- Why can't we just use the regular PowerShell modules?
- How and where do I run these commands?

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>

@thomasvochten

Demo

Graph Explorer

@thomasvochten

Alternative: Postman

Download the Microsoft Graph postman collection

<https://docs.microsoft.com/en-us/graph/use-postman>

(free for personal use)



POSTMAN



<https://www.postman.com/>

Filter

History Collections

Trash

Microsoft Graph v1.0
121 requests

- On Behalf of a User
 - Applications (beta)
 - Batch
 - Events
 - Files
 - Groups
 - Insights
 - Mail
 - Notebooks
 - Extensions
 - People
 - Planner
 - Security
 - SharePoint
 - Subscriptions
 - Tasks
 - Teams
 - Users

- GET Get My Profile
- GET Get My About Me
- GET Get My Skills
- GET Get My Manager

POST Get User Access Token GET Get My Profile

Get My Profile

Examples (0)

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

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 275 ms Size: 1.2 KB Save Download

```
1 {
2   "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",
3   "businessPhones": [
4     "8006427676"
5   ],
6   "displayName": "MOD Administrator",
7   "givenName": "MOD",
8   "jobTitle": null,
9   "mail": "admin@M365x...OnMicrosoft.com",
10  "mobilePhone": "425-882-1032",
11  "officeLocation": null,
12  "preferredLanguage": "en-US",
13  "surname": "Administrator",
14  "userPrincipalName": "admin@M365x...onmicrosoft.com",
15  "id": "c2fbdded-6070-4463-8c56-b784c2610b65"
16 }
```




Authentication & Authorization basics

@thomasvochten

App types and permissions

Get access on behalf of users



Single page app



Web app



Mobile or
desktop app



App with
middle tier web API

Get access as a service



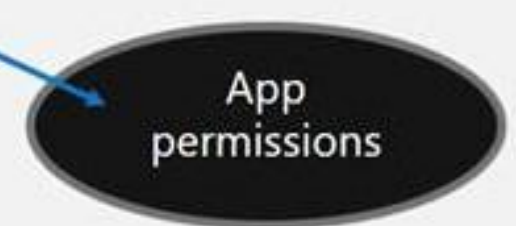
Service or
daemon app

Permission type: delegated



Effective permission

Permission type: application



Users can consent for their data or admin can consent for all users

Only admin can consent



Create an app identity



Configure authentication



Grant the necessary permissions to the app

Typical AuthN & AuthZ workflow



User prompt / device code



AppId & Secret



Certificate

Authentication & Identity options

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

Demo

Authentication & Authorization

@thomasvochten



PowerShell to the rescue!

@thomasvochten

PowerShell SDK

- Abstracts away most of the complexity
- Use the Graph Explorer at will
- `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`

PowerShell SDK

- `Select-MgProfile -Name "beta"`
- Using permission scopes

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

Repeat if you need additional permissions.

Navigating the SDK

`Get-Command -Module Microsoft.Graph* *team*`

Prefix is always “Mg”

Verbs in HTTP vs PowerShell

- GET Get-Mg...
- POST New-Mg...
- PUT New-Mg...
- PATCH Update-Mg...
- DELETE Remove-Mg...

Basic examples

Get-MgUser

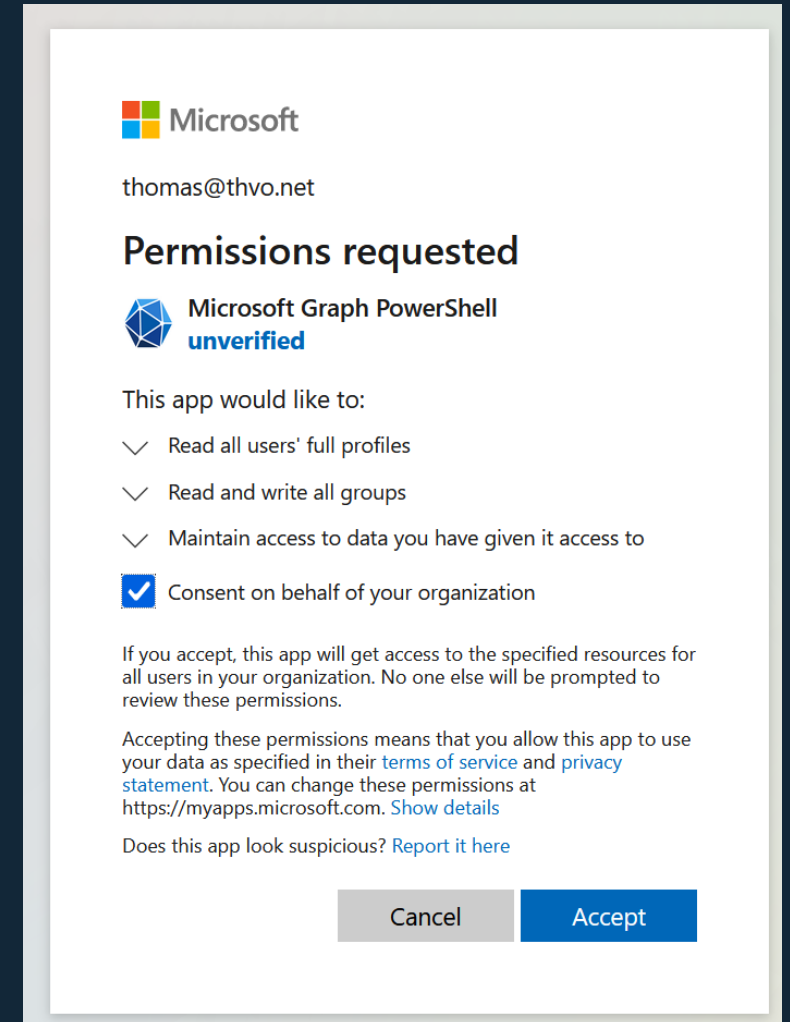
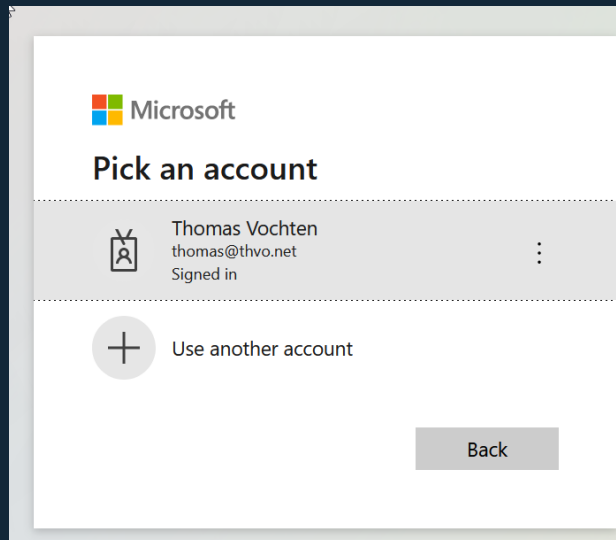
New-MgTeam

Update-MgTeamChannel

Remove-MgGroup

Connecting to the Graph

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



A better alternative

- You need a X.509 certificate to use as a credential
- You need to register your “app” in Azure AD and configure the permissions scopes

Connect-MgGraph

```
-ClientId YOUR_APP_ID  
-TenantId YOUR_TENANT_ID  
-CertificateName YOUR_CERT_SUBJECT
```

Get-MgContext to verify

Demo

Finally some PowerShell!

@thomasvochten

No cmdlet to be found?

Invoke-MGGraphRequest

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

v1 versus v2 (beta)

- Different cmdlets for different endpoints
(Select-MgProfile is no more)
 - Get-MgUser
 - Get-MgBetaUser
- Much smaller size
Better performance
- Install-module Microsoft.Graph.Beta
- New AuthN methods: managed identities, environment variables, ...
- Migration tool will be available

Staying on top of all changes

API Changelog



<https://developer.microsoft.com/en-us/graph/changelog>

@thomasvochten

Trouble in paradise

No support for piping

Managed Identities

OData filters

Takeaways

- The Microsoft Graph is not just for developers!
- Make sure you understand authentication and authorization
- Learn to use the Graph Explorer
- Read the docs, they're pretty good

Thank you

@thomasvochten

<https://thomasvochten.com>

mail@thomasvochten.com