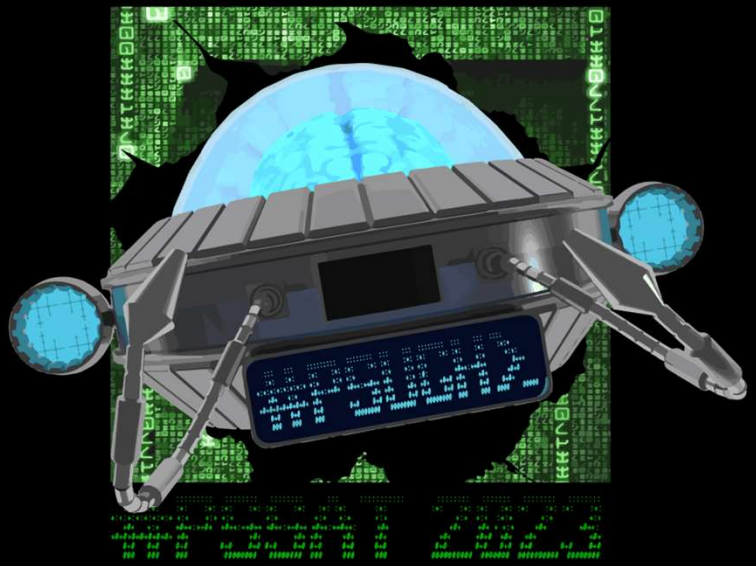


MSGraph PowerShell module

Deep dive into a comfortable implementation of MS Graph API

PS > about_Speaker

- Andreas Bellstedt
  - IT infrastructure guy and PowerShell fellow

@AndiBellstedt

github.com/AndiBellstedt

Netz-Weise
Lernen von den Besten.

au2mator

ScriptRunner®
Your #1 for PowerShell Automation

PS > Agenda

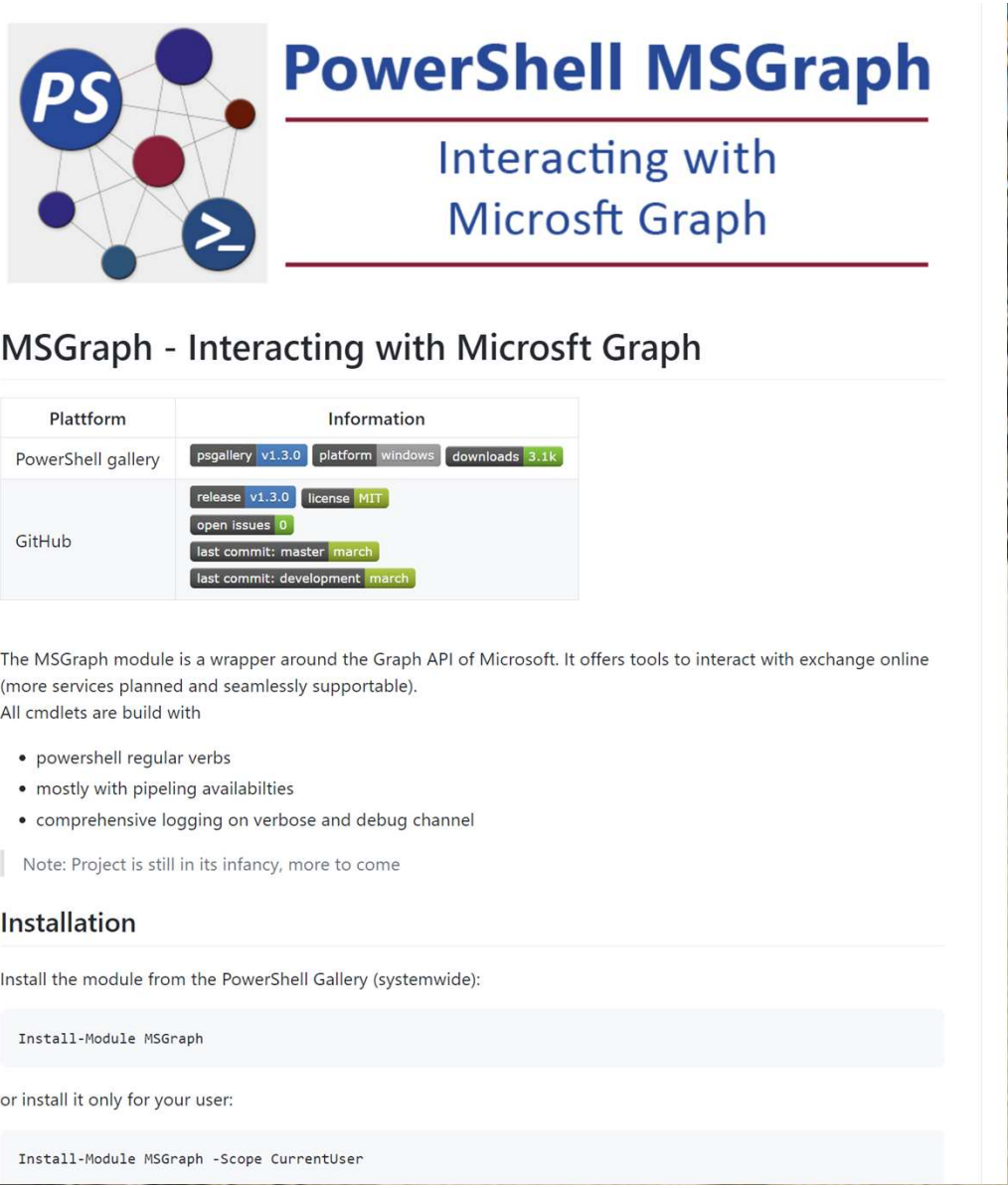
- History – The story behind the module
- Architecture – Bricks and concrete to build it solid
- CI/CD – The way it is released


The story behind the module
Once upon a time…

History

- There was a lonely department, named accounting
- They were trying to invent some kind of digitalization process
- Exchange was migrated to the cloud, but the accounting system remain onprem
- Some brave IT guys came along, talk to the accounting department and decide going into an adventure to rescue all the lonely billing mails and their attachments

- As they can use PowerShell, the story comes like this…
  - Solving a single problem was not enough
  - A platform for further potential challenges should be created
- A new module was born - MSGraph



PS > Demo

- Enough fairy tale… let's get technical

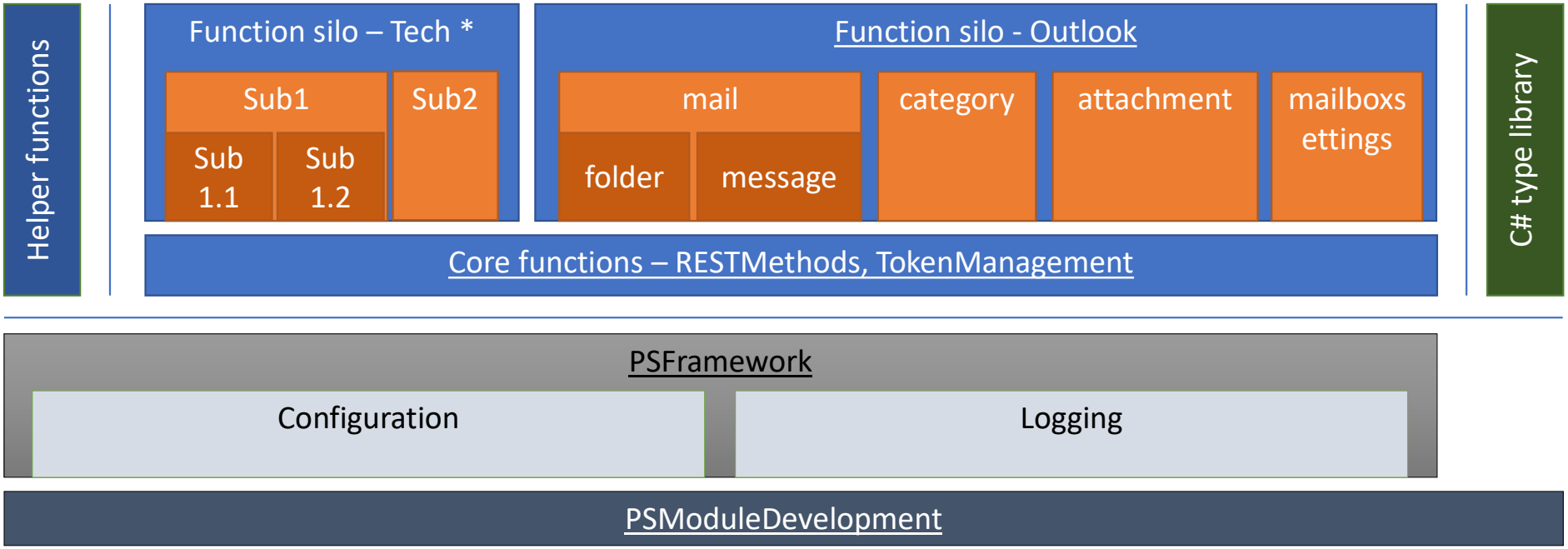
Architecting a module

Bricks and concrete to build it solid

PS > Expectation on the module

- Maintainable & scalable module structure
- Practical & comfortable usage
  - Documentation and help for every function
  - Functions support PS pipeline
- Comprehensible Configuration
  - Customizable
  - Configurable
- Detailed logging
- Reliable naming schema

PS > Architectural landscape



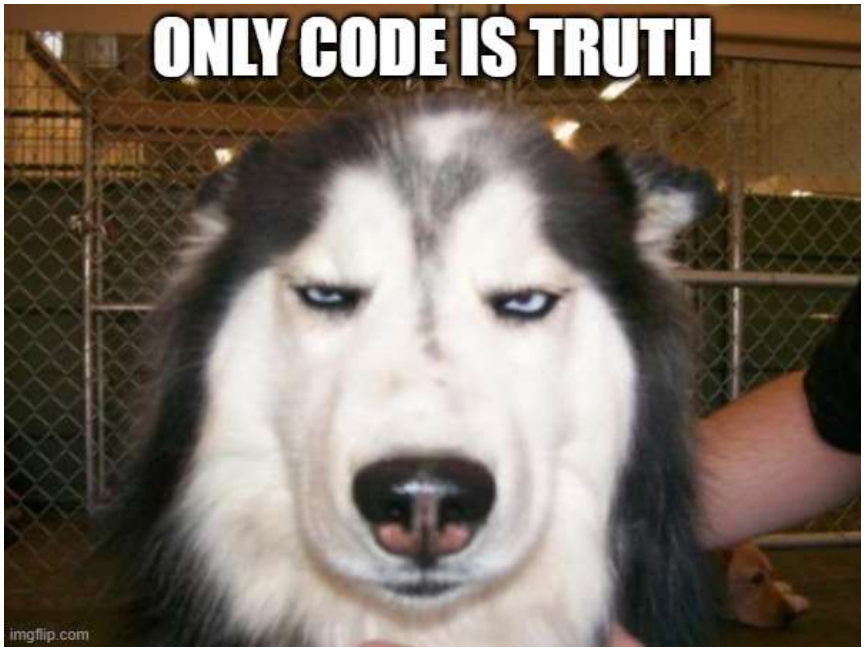
https://learn.microsoft.com/en-us/graph/auth/auth-concepts?view=graph-rest-1.0

Netz-Weise
Lernen von den Besten.

au2mator

ScriptRunner®
Your #1 for PowerShell Automation

PS > Demo



Digging inside the module

Netz-Weise
Lernen von den Besten.

au2mator

ScriptRunner®
Your #1 for PowerShell Automation

PS > Naming schema

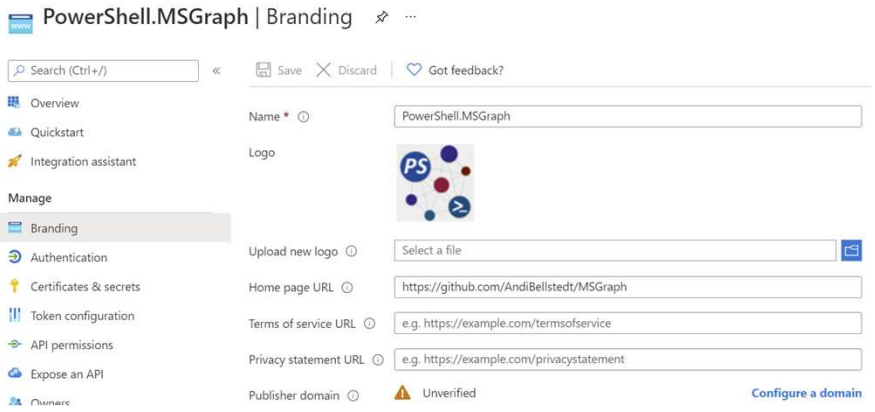
- Avoid collisions on function names with other modules
- Prefix module commands
  - Mga
- Descriptive commands
  - MgaMailMessage, MgaMailFolder, …
- Commonly used verbs
  - Get, Set, Add, Invoke, Remove, …
- Aliases for practical usage
  - Connect <-> New-MgaAccessToken
  - Export <-> Save
  - Update <-> Set

Noun without Prefix	Verb	Name	Type	Resolved command
AccessToken	New	New-MgaAccessToken	Function	
AccessToken	Register	Register-MgaAccessToken	Function	
AccessToken	Update	Update-MgaAccessToken	Function	
AccessToken	Get	Get-MgaAccessTokenRegistered	Function	
AccessToken	Get	Get-MgaRegisteredAccessToken	Alias	Get-MgaAccessTokenRegistered
MailAttachment	Add	Add-MgaMailAttachment	Function	
MailAttachment	Export	Export-MgaMailAttachment	Function	
MailAttachment	Get	Get-MgaMailAttachment	Function	
MailAttachment	Remove	Remove-MgaMailAttachment	Function	
MailAttachment	Save	Save-MgaMailAttachment	Alias	Export-MgaMailAttachment
MailFolder	Get	Get-MgaMailFolder	Function	
MailFolder	Move	Move-MgaMailFolder	Function	
MailFolder	New	New-MgaMailFolder	Function	
MailFolder	Remove	Remove-MgaMailFolder	Function	
MailFolder	Rename	Rename-MgaMailFolder	Function	

Infrastructure requirements

Client

- PowerShell module MSGraph
- Module dependencies
  - PSFramework



Server

- **Entra ID** tenant / Microsoft Account
- Registered Application
- Permissions
  - App registration
  - App granting

What is CI/CD

Continuous integration - Wikipedia

In software engineering, continuous integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day.

CI is intended to be used in combination with automated unit tests written through the practices of test-driven development.

- Maintain a code repository
- Automate the build
- Make the build self-testing

Continuous deployment - Wikipedia

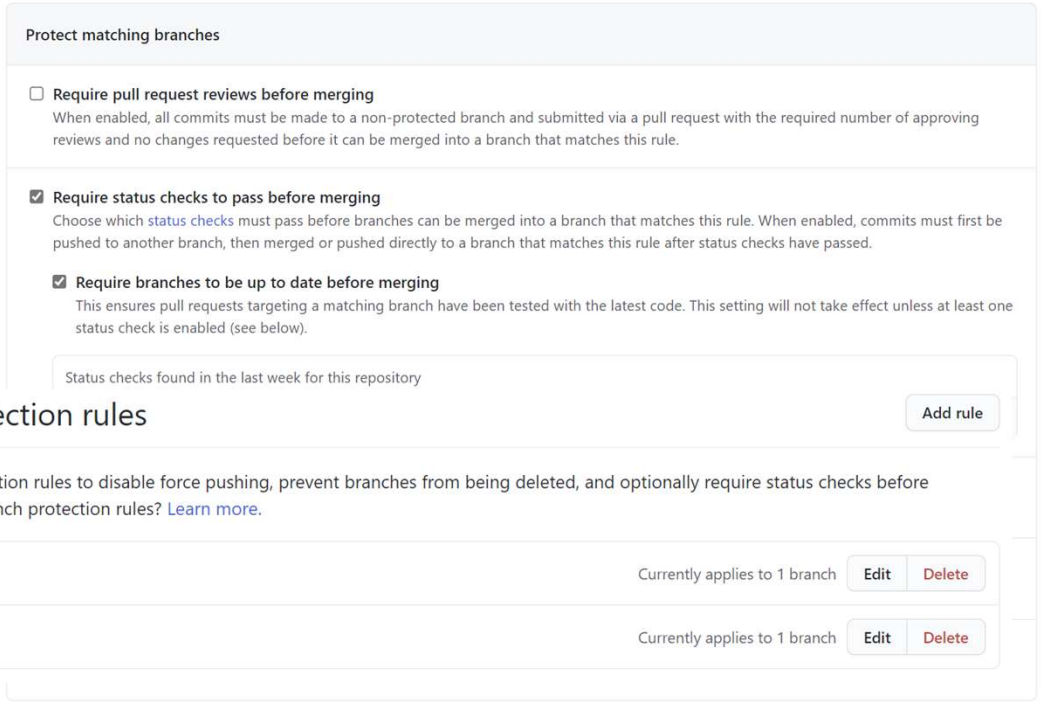
Continuous deployment (CD) is a software engineering approach in which software functionalities are delivered frequently through automated deployments. CD contrasts with continuous delivery, a similar approach in which software functionalities are also frequently delivered and deemed to be potentially capable of being deployed but are actually not deployed.

- Automate the build
- Release after successfull build and test

PS > The way it is released

- Continuous integration & Continuous deployment
- At least 2 branches
  - Master & Development
- Branch protection rules
- The real magic is in:

Azure Pipelines



Netz-Weise
Lernen von den Besten.

au2mator

ScriptRunner®
Your #1 for PowerShell Automation

PS > Azure DevOps - Pipelines

- Defined in yaml files
  - Included inside git repo of module
- Pipeline: Validate
  - Triggered by every commit in Development
  - Triggered by pull request on Master & Development branch
- Pipeline: Build & publish
  - Triggered by commit on master

```yaml
pool:
  vmImage: 'windows-latest'

# Continuous integration only on branch Development
trigger:
  branches:
    include:
    - Development

# Pull request validation only on branch master & development
pr:
  branches:
    include:
    - master
    - Development

steps:
- task: PowerShell@2
  displayName: Ensure prerequisites
  inputs:
    targetType: filePath
    filePath: './build/vsts-prerequisites.ps1'
    arguments: '-ModuleName $(system.teamProject)'

- task: PowerShell@2
  displayName: Validate code compliance
  inputs:
    targetType: filePath
    filePath: './build/vsts-validate.ps1'
    arguments: '-ModuleName $(system.teamProject)'

- task: PublishTestResults@2
  displayName: 'Publish Test Results **/TEST-*.xml'
  inputs:
    testResultsFormat: NUnit
  condition: always()
```

PS > Demo

- GitHub
- Azure DevOps

PS > Conclusion

- Projects from curiosity don‘t go the easy way
- Curiosity brings knowledge
- Module is built with
  - Scaling structure in mind
  - Verbose logging to tell what is going on under the hood
- Module MSGraph is on the PowerShell Gallery
  - (Currently) singularly handling Outlook mail items
  - (Maybe) growing feature set in the future

PS > Question & Answers

Netz-Weise
Lernen von den Besten.

au2mator

ScriptRunner®
Your #1 for PowerShell Automation

– Thank you –

Andreas Bellstedt

@AndiBellstedt

github.com/AndiBellstedt