Graph API Handling via PowerShell

> Requirements: An App Registration with the appropriate permissions and a ClientSecret.

Graph API Authentication

First, the authentication header must be compiled in the script. With this header (here the variable $Header) the authentication at the Graph API can be executed. The top three variables now contain the values, which were compiled in an upper point.

```
$TenantID = "<tenantid>"
$ClientId = "<cliendid>"
$ClientSecret = "<clientsecret>"

$Body = @{
"tenant" = $TenantId
"client_id" = $ClientId
"scope" = "https://graph.microsoft.com/.default"
"client_secret" = $ClientSecret
"grant_type" = "client_credentials"
}

$Params = @{
"Uri" = "https://login.microsoftonline.com/$TenantId/oauth2/v2.0/token"
"Method" = "Post"
"Body" = $Body
"ContentType" = "application/x-www-form-urlencoded"
}
$Global:MicrosoftEntraIDAccessToken = (Invoke-RestMethod @Params).access_token
```

Graph API Resources - Getting Information

This is a simple example query to get information. This only reads out. By the method "GET" this can be recognized on the second line.

```
$Email = "<youremailadress>"
$User = Invoke-RestMethod -Method GET -Uri "https://graph.microsoft.com/v1.0/users/$Email" -
ContentType "Application/Json" -Header @{Authorization = "Bearer
$Global:MicrosoftEntraIDAccessToken" }
```

The following is the output from the $User variable, which has been populated in the top line with information from the Graph API.

```
@odata.context    : https://graph.microsoft.com/v1.0/$metadata#users/$entity
businessPhones    : <yourbusinessphones>
displayName       : <yourdisplayname>
givenName         : <yourforename>
jobTitle          : <yourjobtitle>
mail              : <youremailadress>
mobilePhone       : <yourmobilephonenumber>
officeLocation    : <yourofficelocation>
preferredLanguage : <yourpreferredlanguage>
surname           : <yoursurname>
userPrincipalName : <yourupn>
id                : <youruserid>
```

Graph API Resources - Create information

In the following example, an entity is created via the Graph API in Intune. Here, the necessary information is now also transmitted, using JSON Body.

```
$KGTag = "TST"
$ScopeTagProdName = "SCT-INT-$KGTag-INTUNE-KGObjects-PROD"
$ScopeTagProdBody = @"
{
```

```
    "displayName":"$ScopeTagProdName",
    "description":"ScopeTag for Company $KGTag"
}
"@
$global:ScopeTagProd = Invoke-RestMethod -Method POST -Uri
"https://graph.microsoft.com/beta/deviceManagement/roleScopeTags" -ContentType
"Application/Json" -Header @{Authorization = "Bearer $Global:MicrosoftEntraIDAccessToken" } -
body $ScopeTagProdBody
```

$global:ScopeTagProd is a global variable which has been populated with the return of the graph query above. The content of the variable is as follows:

```
id displayName □□□□□□□   description                 isBuiltIn
-- ----------- □□□□□□□   -----------             □□---------
45 SCT-INT-TST-INTUNE-KGObjects-PROD   ScopeTag for Company TST     False
```

Revision #5
Created 2022-12-02 10:31:43 UTC
Updated 2026-03-03 16:18:58 UTC by Luca Noah Caprez