Download & read files via Graph API

> Requirements: App Registration with appropriate permissions, as described here: [Read SharePoint files … | LNC DOCS (lucanoahcaprez.ch)](#)

With this tutorial, files can be read from SharePoint Online repositories automatically and without user interaction. This can be especially valuable when large accesses to files need to be made, but the budget for Azure Blob Storage, for example, is not available. It also allows very easy customization of the corresponding files in SharePoint Online, which can then be used directly in the automation.

Preparations

First of all, the app registration must be created with the appropriate permissions. This is described here: [Read SharePoint files … | LNC DOCS (lucanoahcaprez.ch)](#) Then the app must be assigned the appropriate permissions on SharePoint Online once.

This can be achieved with the following PowerShell code. The two upper variables must be filled in accordingly with the SharePoint Site URL and the App Registration ID.

```
$siteUrl = "https://<yourtenantname>.sharepoint.com/sites/<yoursitename>"
$ClientId = "<yourappid>"

# Connect to SPO Site (PNPOnline Modul required)
Connect-PnPOnline -Url $siteUrl -Interactive #

# Grant write permission to the App Registration to the corresponding SPO site & files
Grant-PnPAzureADAppSitePermission -Permissions "Write" -Site $siteUrl -AppId $ClientId -DisplayName "$ClientId - Write Access"
```

Getting the files via Graph API

The following code can be used in an automation for example. Here all files are written into an array without the execution needing any interaction.

It is also important here that the variables are correctly filled in at the beginning.

```powershell
$SiteID = "<yoursharepointsiteid>"
$TenantId = "<yourtenantid>"
$ClientId = "<yourappregistrationid>"
$ClientSecret = "<yourclientsecret>"
$LibraryName = "Documents"
$OutPathRoot = "<youroutputfolderforfilecontent>"

if (-not $(Test-Path -LiteralPath $OutPathRoot)) {
    New-Item $OutPathRoot -Force -ItemType Directory | Out-Null
}

# Get Bearer Token for authentication against Graph API
$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"
}
$AuthResponse = Invoke-RestMethod @Params
$Header = @{
    "Authorization" = "Bearer $($AuthResponse.access_token)"
}

# Get Drive ID of the Site
$Params = @{
    "Method"    = "Get"
    "Uri"       = "https://graph.microsoft.com/v1.0/sites/$SiteID/drive"
    "Headers"   = $Header
```

```
    "ContentType" = "application/json"
}
$DriveID = $(Invoke-RestMethod @Params | where { $_.name -eq $LibraryName }).id


# Get all files from the SharePoint folder
$Params = @{
    "Method"     = "Get"
    "Uri"        = "https://graph.microsoft.com/v1.0/sites/$SiteID/drives/$DriveID/root/children"
    "Headers"    = $Header
    "ContentType" = "application/json"
}
$SPFiles = $(Invoke-RestMethod @Params).Value
```

Select the option how you want to work with the file(s)

Afterwards, the files can be either all downloaded, downloaded individually or only the content can be viewed. Accordingly, the appropriate code piece must be selected.

Download all files

```
# Download all the files
foreach ($File in $SPFiles) {
    if ($File.'@microsoft.graph.downloadUrl') { # Schliesst Ordner aus
        $Params = @{
            "Method"     = "Get"
            "Uri"        = $File.'@microsoft.graph.downloadUrl'
            "ContentType" = "application/json"
            "OutFile"    = "$OutPathRoot\$($File.name)"
        }
        Invoke-RestMethod @Params
    }
}
```

Download one file

```
# Download only one file
$File = $SPFiles[1]
```

```
$Params = @{
    "Method"    = "Get"
    "Uri"       = $File.'@microsoft.graph.downloadUrl'
    "ContentType" = "application/json"
    "OutFile"   = "$OutPathRoot\$($File.name)"
}
Invoke-RestMethod @Params
```

Get the content of one file

```
# Get the content of a file
$File = $SPFiles[1]
$Params = @{
    "Method"    = "Get"
    "Uri"       = $File.'@microsoft.graph.downloadUrl'
    "ContentType" = "application/json"
}
$Content = Invoke-RestMethod @Params
$Content
```

Revision #6
Created 31 March 2023 13:27:41
Updated 6 August 2025 15:26:57 by Luca Noah Caprez