**Delegate sample:**

AAD authentication flow: [https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Factive-directory%2Fdevelop%2Fv2-oauth2-auth-code-flow&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236095755%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=mkWO%2FUwHFSLwDQc60TpdlX3xQa0rRob1FL7N30OSMKY%3D&reserved=0)

**PowerShell code sample to implement Authorization Code Auth flow**

#Define Client Variables Here

#############################

$TenantId="ed3c1c81-f3be-495c-8028-f11b7ad6415a"

$clientId = "87202bda-1a2a-4b6c-917b-df77c98c640d"

$clientSecret = "xxx"

$scope = "<https://graph.microsoft.com/.default>"

$redirectUri = "<https://localhost>" #this can be set to any URL

#$resource = "<https://graph.microsoft.com>"

#UrlEncode variables for special characters

###########################################

Add-Type -AssemblyName System.Web

$clientSecretEncoded = [System.Web.HttpUtility]::UrlEncode($clientSecret)

$redirectUriEncoded =  [System.Web.HttpUtility]::UrlEncode($redirectUri)

$scopeEncoded = [System.Web.HttpUtility]::UrlEncode($scope)

#Obtain Authorization Code

##########################

Add-Type -AssemblyName System.Windows.Forms

$form = New-Object -TypeName System.Windows.Forms.Form -Property @{Width=440;Height=640}

$web  = New-Object -TypeName System.Windows.Forms.WebBrowser -Property @{Width=420;Height=600;Url=$url}

$url = "<https://login.microsoftonline.com/$TenantId/oauth2/v2.0/authorize?response_type=code&redirect_uri=$redirectUriEncoded&client_id=$clientId&scope=$scopeEncoded&prompt=admin_consent>"

$DocComp  = {

        $Global:uri = $web.Url.AbsoluteUri

        if ($Global:uri -match "error=[^&]\*|code=[^&]\*") {$form.Close() }

    }

$web.ScriptErrorsSuppressed = $true

$web.Add\_DocumentCompleted($DocComp)

$form.Controls.Add($web)

$form.Add\_Shown({$form.Activate()})

$form.ShowDialog() | Out-Null

$queryOutput = [System.Web.HttpUtility]::ParseQueryString($web.Url.Query)

$output = @{}

foreach($key in $queryOutput.Keys){

    $output["$key"] = $queryOutput[$key]

}

$regex = '(?<=code=)(.\*)(?=&)'

$authCode  = ($uri | Select-string -pattern $regex).Matches[0].Value

#Get Access Token with obtained Auth Code

#########################################

$body = "grant\_type=authorization\_code&redirect\_uri=$redirectUri&client\_id=$clientId&client\_secret=$clientSecretEncoded&code=$authCode&resource=$resource"

$authUri = "<https://login.microsoftonline.com/common/oauth2/token>"

$tokenResponse = Invoke-RestMethod -Uri $authUri -Method Post -Body $body -ErrorAction STOP

#Call Graph API to download a file

#########################################

#$DownloadUri=https://graph.microsoft.com/v1.0/me/drive/root:/ee - Copy.xlsx:/content

$DownloadUri = "<https://graph.microsoft.com/v1.0/sites/ec884a3f-7f7e-460a-900b-39c61f8195be/drive/items/01DFEMAO44C7FIP2APCBEZFLF67OL6JAIZ/content>"

$destinationFilePath = "C:\Users\menxia\Desktop\Files\Graph Test\Test.pdf"

$header =@{

    'Authorization' = "Bearer $($tokenResponse.access\_token)"

}

$results = Invoke-RestMethod -Uri $DownloadUri -Headers $header -Method Get -OutFile $destinationFilePath

 ===========================

**Graph PowerShell SDK**: [https://docs.microsoft.com/en-us/graph/powershell/get-started](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fgraph%2Fpowershell%2Fget-started&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236105710%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=%2FGgsy49doZb1FtG2uYeU%2BCT7CCgX18X9IyAup4F52EY%3D&reserved=0)

1. Run command to connect Microsoft Graph in PowerShell:



This uses device code flow to authenticate: [https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-device-code](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Factive-directory%2Fdevelop%2Fv2-oauth2-device-code&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236115669%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=pEIiEzEuziES55HA47BHAa0s%2FHz7KKAS%2BXug2Hu0TAw%3D&reserved=0)

Graphical user interface, application

Description automatically generated    Graphical user interface, text, application

Description automatically generated   Graphical user interface, text, application, email

Description automatically generated   Graphical user interface, text, application

Description automatically generated

Verify the context:

Graphical user interface, text

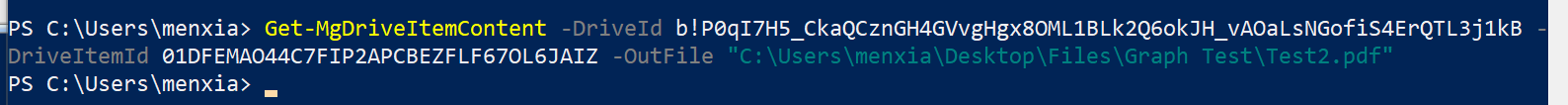
Description automatically generated

Actually it is using this auto-registered app called Microsoft Graph PowerShell for doing the authentication:

Graphical user interface, application

Description automatically generated

1. Call graph API to download a file:



1. Reference: [Microsoft.Graph.Files Module | Microsoft Docs](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fpowershell%2Fmodule%2Fmicrosoft.graph.files%2F%3Fview%3Dgraph-powershell-beta&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236115669%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=eDW5%2F0phdSetPkkqSfmEfyn9TQK5FBlHIFxpO2Mfl3s%3D&reserved=0)

**Application only sample:**

AAD authentication flow: [https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Factive-directory%2Fdevelop%2Fv2-oauth2-client-creds-grant-flow&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236125623%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=NGLtwOHa0vUXrsFj%2FZfmtOmMxzUMJxLZq8Wg4pEF51Q%3D&reserved=0)

**PowerShell code sample to implement client credential grant flow**

#Define Client Variables Here

#############################

$TenantId='ed3c1c81-f3be-495c-8028-f11b7ad6415a'

$ClientId='87202bda-1a2a-4b6c-917b-df77c98c640d'

$ClientSecret='xxx'

$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](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Flogin.microsoftonline.com%2F%24TenantId%2Foauth2%2Fv2.0%2Ftoken&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236135582%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=MJbcHQ9%2F5gnYfJTbHFvkOZ0oI1L9mfGSCzqiNsq%2BJ3o%3D&reserved=0)   #tenant name is also fine here, like xia053.onmicrosoft.com

    'Method' = 'Post'

    'Body' = $Body

    'ContentType' = 'application/x-www-form-urlencoded'

}

#Get Access Token

##########################

$AuthResponse = Invoke-RestMethod @Params

#Call Graph API to add a new folder

####################################

$FolderName ='New Folder Name'

$SiteId = 'ec884a3f-7f7e-460a-900b-39c61f8195be'

$Uri=https://graph.microsoft.com/v1.0/sites/$SiteId/drive/root/children

$post = @"

{

        "name":  "$FolderName",

        "folder":  { }

}

"@

$header =@{

    'Authorization' = "Bearer $($AuthResponse.access\_token)"

    'Content-Type' = 'application/json'

}

$results = Invoke-WebRequest -Uri $Uri -Headers $header -Method Post -Body $post

========================================

**Graph PowerShell SDK**: [https://docs.microsoft.com/en-us/graph/powershell/app-only?tabs=azure-portal](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fgraph%2Fpowershell%2Fapp-only%3Ftabs%3Dazure-portal&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236135582%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=f9DyF8Pn%2F4JvxbxO5OmAcARDbRtUfH9dpwRh9f0PkbY%3D&reserved=0)

Note: we do not support client secret due to security reason

[https://github.com/microsoftgraph/msgraph-sdk-powershell/issues/686](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fmicrosoftgraph%2Fmsgraph-sdk-powershell%2Fissues%2F686&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236145539%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=COww6W4Ex1zvcS%2BtPkkjV9qqiwNU3UDuBESeH9RTCRI%3D&reserved=0)

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

1. Create a self-signed certfiicate for test: [https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-self-signed-certificate](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Factive-directory%2Fdevelop%2Fhowto-create-self-signed-certificate&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236145539%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=h85dm8Vz2yyict3ySa2VOS7BSosLxWtHw%2BMj%2BjyAmsQ%3D&reserved=0)

$cert = New-SelfSignedCertificate -Subject "CN=VicCert" -CertStoreLocation "Cert:\CurrentUser\My" -KeyExportPolicy Exportable -KeySpec Signature -KeyLength 2048 -KeyAlgorithm RSA -HashAlgorithm SHA256



Export-Certificate -Cert $cert -FilePath "C:\Users\menxia\Desktop\VicCert.cer"

Graphical user interface, text

Description automatically generated

$mypwd = ConvertTo-SecureString -String "Access1" -Force -AsPlainText

Export-PfxCertificate -Cert $cert -FilePath "C:\Users\menxia\Desktop\VicCert.pfx" -Password $mypwd

Graphical user interface, text

Description automatically generated

1. Register an application in Azure AD and upload the certificate (.cer) into the registered app.
2. Authenticate to Graph in PowerShell:

Connect-MgGraph -ClientID 87202bda-1a2a-4b6c-917b-df77c98c640d -TenantId ed3c1c81-f3be-495c-8028-f11b7ad6415a -CertificateName "CN=VicCert"

Text

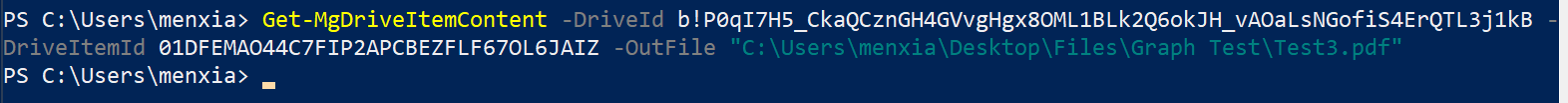
Description automatically generated

Connect-MgGraph -ClientID 87202bda-1a2a-4b6c-917b-df77c98c640d -TenantId ed3c1c81-f3be-495c-8028-f11b7ad6415a -CertificateThumbprint 8F1973A927B0DCFA4E4A71B251A3B3CE98C48CEB

Text

Description automatically generated

1. Call Graph API to download a file:



1. Reference: [https://docs.microsoft.com/en-us/graph/powershell/navigating](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fgraph%2Fpowershell%2Fnavigating&data=04%7C01%7Cchunlonl%40microsoft.com%7C87388d9f66f044c1d96e08d9afe13f50%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637734204236155495%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=Eo73oGHa50p3Y3pB1Kfnwc4PTOu4sQtZyndG%2FNMprS8%3D&reserved=0)

Graphical user interface, text, application, email

Description automatically generated