

Upload LF Repo Docs to SharePoint

Vital Links

[Graph Explorer](#)

[Microsoft Docs - Upload Small Files](#)

[Requesting an Access Token](#)

Description

This Workflow and Script allows you to upload small files to SharePoint Online using Microsoft Graph API.

Tested File Types: pdf, tiff, xlsx, docx

Try more and let me know if they work! The script should be able to handle anything.

The uploaded file size is limited to 4MB or less.

The Graph API endpoint I used makes a single request, which causes this size limitation. Graph API is capable of creating "upload sessions" for larger files, but that remains an area for improvement for now.

Setting Up

What You'll Need

- LaserFiche Repo Entry - The Script's Default Entry targets the file for upload.
- Access Token - Used to authorize the Graph API call
 - Create an App Registry in Azure AD
 - Set Parameters in Workflow to request Access Token using new App Registry
- Graph API Parameters
 - SharePoint Drive-ID
 - SharePoint Item-ID of either
 - Target Containing Folder (for new files)
 - Target File (for updating an existing file)

Selecting the File For Upload

The implementation uploads a file designated by the Script's Default Entry. The Upload HTTP Request is made inside the script, so this bit is pretty simple to set up.

More Info: The Graph API Call requires the file contents to be sent as a "Binary Stream." This data type isn't supported outside of the SDK script, which is why the HTTP Request needs to be made within the script. To export the document to the binary stream, the script must handle electronic document (like a pdf) from those which aren't (like tiff) slightly differently. This is the only distinction made between files, though, and so should be able to handle any file type.

The Access Token

The Access Token is required by the Graph API when making HTTP Requests. To get an Access Token, the workflow makes a request to <https://login.microsoftonline.com> to authenticate against your Azure AD App Registry. So you'll need to

1. Make an App Registration
2. Set the App Registration info in Workflow

Registering an App in Azure AD

You can find [MS documentation here](#), but I've summarized the relevant instructions below if you want to [just go for it](#).

1. Give your App Registration a name
2. Probably only want to support "this organization only"
3. Don't need a Redirect URI

More info: The redirect URI is for apps that send users to MS for login and then your app awaits MS to send an access token to the redirect URI. We don't need that, as Workflow will simply send a request to MS auth service and then it'll send an access token right back.
4. Once registered, grab the data you're going to need:
 - a. Application (client) ID
 - b. Directory (tenant) ID
5. Set the Apps Permissions (as required by the [Graph API Upload Call](#)). These are:
 - a. Application Level: Files.ReadWrite.All, Sites.ReadWrite.All
6. Create a new Client Secret
 - a. Make a note of this somewhere, it cannot be exposed again on Azure AD Portal.

Setting the Request Parameters

The parameters for acquiring an Access Token can be set in the "Access Token Params" activity in workflow. The parameters are as follows:

tenant		From your App Registration
client_id		From your App Registration
scope	https%3A%2F%2Fgraph.microsoft.com%2F.default	Default App permissions
client_secret		From your App Registration
grant_type	client_credentials	For getting an Access Token without a user

For more info about these parameters check [MS Docs - Get an access token](#).

Finally, you'll need to create a webservice object as (I hope) this isn't saved in the workflow I've provided. In the "HTTP Web Request" Activity, select "Manage Web Services..." from the dropdown menu under **Web Service**. Create a new webservice (I called mine "Microsoft Auth") and set the URL to <https://login.microsoftonline.com>.

For more info, see [LF Docs - Web Services](#).

Setting the Graph API Parameters

These are the Graph API calls made to upload your document.

New file	/drives/{drive-id}/items/{parent-id}/{filename}/content
Update file	/v1.0/drives/{drive-id}/items/{item-id}/content

These are handled within the SDK Script, you just need to supply the parameters.

The Parameters You'll Need

Uploading a New File

Drive ID	ID of the SharePoint drive you will be using
Parent ID	ID of the Drive-Item which will be the parent folder of your new upload
FileName	Provide a name for your file here, include the extension

Uploading contents to Update an Existing File

Drive ID	ID of the SharePoint drive you will be using
Item ID	ID of the Drive-Item which you will be updating

How to Get IDs

You can dig out the IDs you need by playing with the [Graph Explorer](#).

I did not find this extraordinarily intuitive, but I'll provide an example of how I find IDs below. If you find/have a better method for finding the required IDs, please tell me know about it. Otherwise, best of luck.

Digging in Graph Explorer Example:

-- These are all GET v1.0 requests --

Find a SharePoint Site:

Search for Site	sites?search={search}
---------------------------------	-----------------------

the ?\$select filter isn't required on any below queries, it just makes the responses easier to read

Use Site-ID to get Site Drive:

Site drive	sites/{site-id}/drive?\$select=name,id,webUrl
----------------------------	---

Note Drive-ID for params and use it to find Item-IDs:

Drive Root Folder	/drives/{drive-id}/root/children?\$select=name,id,webUrl
-----------------------------------	--

You can upload to this folder or use this ID to explore sub folders like

Drive Root Folder	/drives/{drive-id}/items/{item-id}/children?\$select=name,id,webUrl
-----------------------------------	---

That last query will also get you the IDs of files, a requirement for uploading updates.

Setting the Upload Parameters

The SDK Script pulls values from Workflow Tokens which are defined in the workflow Input Parameters.

I also added an assign tokens activity "Upload Request Param Interception" (disabled by default) where you can modify the tokens in the workflow if you wish. For example, you could drop a "Retrieve Business Process Variables" activity in your workflow to use info from a forms process.