Table of Contents

[0. Introduction 2](#_Toc39156729)

[0.1 SharePoint REST API V1 2](#_Toc39156730)

[0.2 Prerequisites 2](#_Toc39156731)

[1. Create a SharePoint App Registration 3](#_Toc39156732)

[1.1 Register a SharePoint App 3](#_Toc39156733)

[1.2 Provide the SharePoint App with Permissions 6](#_Toc39156736)

[2. Writing Requests to the SharePoint Rest API in Python 8](#_Toc39156737)

[2.1 Get an AccessToken for SharePoint in Python 8](#_Toc39156738)

[3. Working with Folders and Files with SharePoint REST 14](#_Toc39156739)

[3.1 Working with Files and Folder by using REST 14](#_Toc39156740)

[3.1.1 Retrieve all the Files in a Folder. 14](#_Toc39156741)

[3.1.2 Retrieve a Specific File. 15](#_Toc39156742)

[3.1.3 Create a File and Add it to a Folder. 15](#_Toc39156743)

[3.1.4 Create a New Folder. 16](#_Toc39156744)

[3.2 Working with large files by using REST 17](#_Toc39156745)

[3.2.1 Create a File by using this REST Endpoint 17](#_Toc39156746)

[4. References 19](#_Toc39156747)

[Get AccessToken 19](#_Toc39156748)

[Retrieve all Files in a folder 19](#_Toc39156749)

[Retrieve a specific File 20](#_Toc39156750)

[Delete a File 20](#_Toc39156751)

[Add a new Folder 20](#_Toc39156752)

[Add a new File inside a Folder 21](#_Toc39156753)

[Add a large File (more than 1.5mb) inside a Folder 21](#_Toc39156754)

# 0. Introduction

# 0.1 SharePoint REST API V1

In this documentation you will be able to start using the SharePoint Rest API V1 in your server. We will guide you on how to start creating new SharePoint applications. As well as how to generate an access token so you can start to Create, Read, Update, and Delete objects in SharePoint. At the end of this documentation you will find the Python code that were used for the examples.

## 0.2 Prerequisites

* SharePoint Administration Permission
* Being Able to access portal.azure.com
* Python 3.6 or greater
* Text editor

*Visual Studio Code was the prefer Text Editor for this documentation.*

# 1. Create a SharePoint App Registration

SharePoint includes a Representational State Transfer (REST) service that is comparable to the existing SharePoint client object models. Developers can interact remotely with SharePoint data by using any technology that supports REST web requests. This means that developers can perform **Create**, **Read**, **Update**, and **Delete** (CRUD) operations from their SharePoint Add-ins, solutions, and client applications, using REST web technologies.

In this documentation we are going to take you *step-by-step* on how you can start using SharePoint Rest API in Python.

## 1.1 Register a SharePoint App

In this section we are going to use [SharePoint Rest API v1](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-to-know-the-sharepoint-rest-service?tabs=csom). Will be covering the access to SharePoint data via [SharePoint REST API v1](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-to-know-the-sharepoint-rest-service?tabs=csom).

First, we are going to start by creating SharePoint app and then use the credentials of the app to get an **Access Token**. Once we get the **Access Token**, we are going to use the token in a HTTP request to get the required data from SharePoint. We will cover the HTPP request on the next section.

To start, you will need to register a new App as a *SharePoint Administrator*.

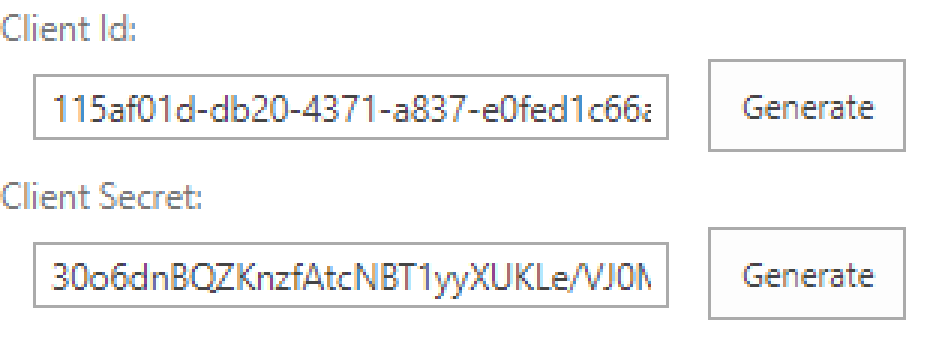
To create and access the app’s information site use the following link.

https://**{yourtenantname}.**sharepoint.com**/\_layouts/15/appregnew.aspx**

The page will display the information below,



Here we will generate a **Client Id** and **Client Secret** as follows,



Copy the generated **Client Id** and **Client Secret** into your favorite editor since we will need these later once we start working with Python.

Once you generate a **Client Id** and **Client Secret** a Fill the remaining fields.

***Title***:

A brief description of the token.

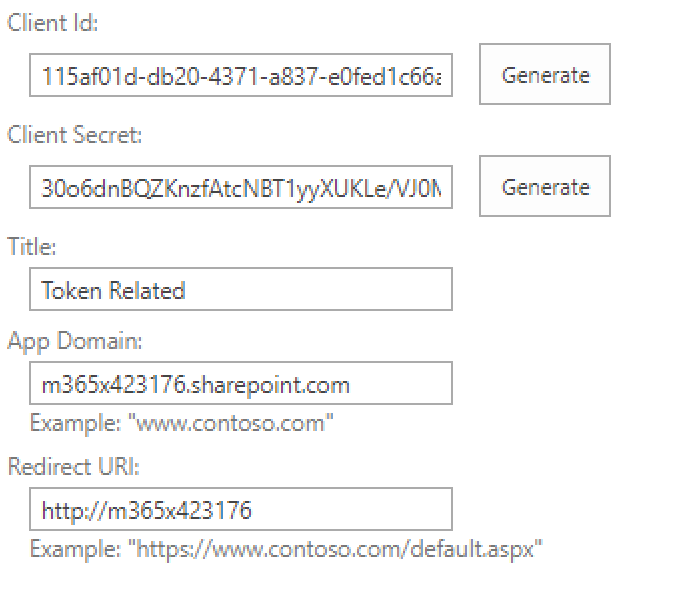
***App Domain:***

Your SharePoint Domain Name.

***Redirect URI:***

URI to redirect when the Authentication endpoint its call.

Fill the details in that page as per the following table and click **“Create”.**



## 

***… And you are all set. You successfully created your first SharePoint App Registration.***

## 1.2 Provide the SharePoint App with Permissions

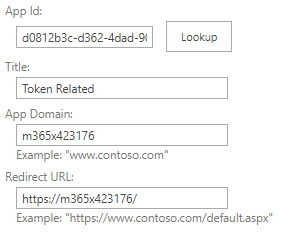
Now that your SharePoint App is registered, we need to add **Permissions** to the app so that it can access data. In order to do that, navigate through the following site,

https://**{yourtenantname}.**sharepoint.com **/\_layouts/15/appinv.aspx**

In this site grab the **Client Id** that you saved before and paste it in the **App Id** field. Then proceed and click on **“Lookup”.** This will load the details of the app we registered previously.



After clicking on Lookup, the other fields will be fill with the details of the app we registered previously.



In the “**Permission Request XML”** field we’ll paste the next XML.

**<AppPermissionRequests AllowAppOnlyPolicy="true">**

**<AppPermissionRequest**

**Scope="http://{TenantName}.sharepoint.com/content/sitecollection/web"**

**Right="FullControl"/>**

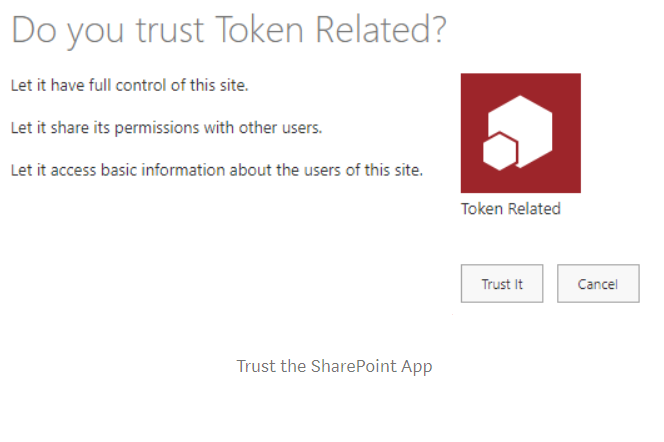
**</AppPermissionRequests>**

For this example, we’ll grant full control to the app. If you need to give different permissions then please take a look at this [Add-in Permissions in SharePoint](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/add-in-permissions-in-sharepoint) article.

Once you pasted the XML code click on **“Create”**.



In the next page click on **“Trust it”** and this will provide the app with the required permissions.



If you need to deploy multiple SharePoint Apps with *different* permissions. Follow the steps mentioned here again and don’t forget to save your **Client Id** and **Client Secret** for each app.

# 2. Writing Requests to the SharePoint Rest API in Python

Now that we successfully deployed a SharePoint App our next goal will be to generate a **Bearer token**. Bearer Token is access **security token** with the property that any party in possession of the token, a “bearer”, can use the token in any way that any other party in possession of it can.

## 2.1 Get an AccessToken for SharePoint in Python

If we want to make calls to SharePoint, we’ll need to get an AccessToken in our code.

We’ll be using the next Python Libraries:

* **requests:** This library help us make http request to the SharePoint Rest API.
* **json:** This library helps us formatting the API responses.

Let start by making a settings.py file in our working directory, in this file we’ll put some of the data we get when we registered the SharePoint App in a Python Dictionary. Here’s an example,

{YourWorkDirectory}/ SharePointRestAPISettings.py:

settings = {

 "appReg\_clientId": "{YourClientId}",

  "appReg\_clientSecret": "{YourClientSecret}",



 "targetHost": "{YourTenantName.sharepoint.com}",

 "principal": "00000003-0000-0ff1-ce00-000000000000",

 "realm": "{Azure Active Directory ID}",

 "appReg\_bearerToken": ""

}

Let’s break down the settings one by one:

Settings appReg\_clientId:



The client id we got after registered the SharePoint App.

Settings appReg\_clientSecret:



The client secret we got after registered the SharePoint App.

Settings targetHost:



Your SharePoint Site (Example: YourTenantName.sharepoint.com)

Settings principal:



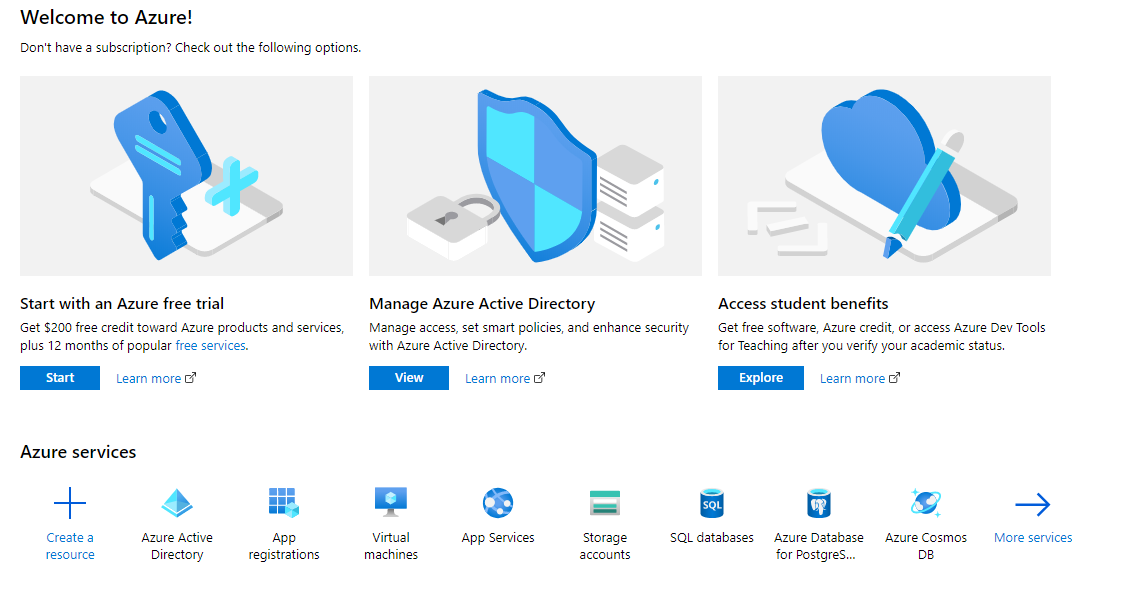
Go to: The realm will be the tenant ID for Office 365, or the farm ID for your on-premises deployment. The Principal ID will change based on the endpoint. Since we are using SharePoint, we are going to use the ID bolded below. For more information in principal take a look the [Inside SharePoint 2013 OAuth Context Tokens](https://docs.microsoft.com/en-us/archive/blogs/kaevans/inside-sharepoint-2013-oauth-context-tokens) documentation.

|  |  |
| --- | --- |
| ACS | 00000001-0000-0000-c000-00000000000 |
| Exchange | 00000002-0000-0ff1-ce00-000000000000 |
| SharePoint | **00000003-0000-0ff1-ce00-000000000000** |
| Lync | 00000004-0000-0ff1-ce00-000000000000 |
| Workflow | 00000005-0000-0000-c000-00000000000 |

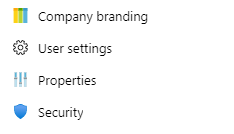
Settings realm:



The realm Refers to the **Azure Active Directory Id**. You can get it by going to Azure and Log in in your administrator account, once you logged go to the Home page, then click on **Azure Active Directory**.



Once in **Azure Active Directory** in the sidebar click on **Properties.**



In the Properties screen copy the Directory ID that’s your **Azure Active Directory Id.**



We are going to use the Azure Active Directory and use it on the realm setting from our Python code.

Last, Settings appReg\_bearerToken:



This part will take our **AccessToken** from the **API response**, for now leave it empty.

The result will look something like this,

User/settings.py:

settings = {

    "appReg\_clientId": "d0812b3c-d362-4dad-9035-e0fd017fe895",

    "appReg\_clientSecret": "JP7yqSPQQOiZs8jyoHr6ks9pUz4j1n8+ncvlUOUam30=",

    "targetHost": "m365x423176.sharepoint.com",

    "principal": "00000003-0000-0ff1-ce00-000000000000",

    "realm": "1b14560c-1aa5-434c-b667-c2819926093",

    "appReg\_bearerToken": ""

}

Now we are all set to obtain our bearer Token.

After we fill that data, we need to create another Python file in the same directory. In this file we are going to retrieve the AccessToken with the settings.py data and with that token we’ll make the requests to the SharePoint Rest API.

{YourWorkDirectory}/SharePointRestAction.py:

import requests

import json

import SharePointRestAPISettings

import uuid

import os

def getToken(settings):

    body = {

        "grant\_type": "client\_credentials",

        "client\_id": settings["appReg\_clientId"] + "@" + settings["realm"],

        "client\_secret": settings["appReg\_clientSecret"],

        "resource": settings["principal"] + "/" + settings["targetHost"] + "@" + settings["realm"]

    }

    response = requests.post("https://accounts.accesscontrol.windows.net/" +

                             settings["realm"] + "/tokens/OAuth/2", data=body).json()

    settings["appReg\_bearerToken"] = response["access\_token"]

Let’s break down the code for getting the AccessToken:

First, we import the libraries we are going to use to make the requests to the SharePoint Rest API. Also, we import the settings file we make before to access the settings data.

Then we construct the body of the request with the data from the settings file.

    body = {

        "grant\_type": "client\_credentials",

        "client\_id": settings["appReg\_clientId"] + "@" + settings["realm"],

        "client\_secret": settings["appReg\_clientSecret"],

        "resource": settings["principal"] + "/" + settings["targetHost"] + "@" + settings["realm"]

    }

We are going to make the request to the SharePoint Rest API using the library requests.

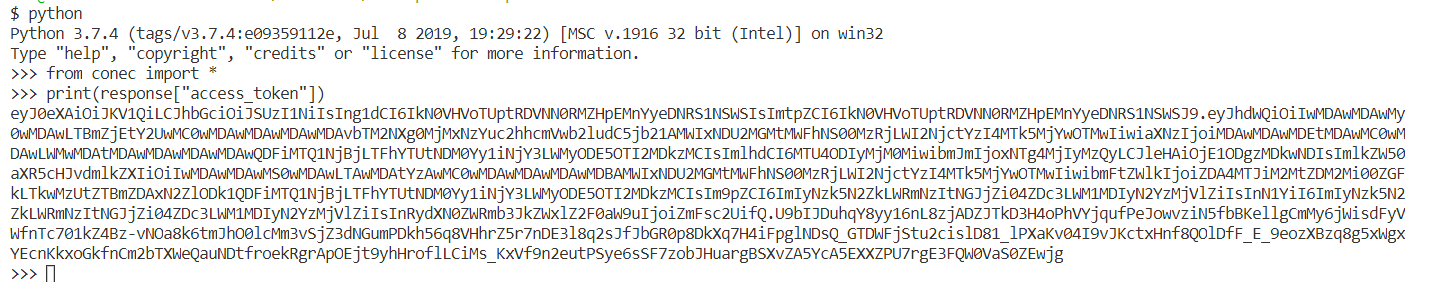
response = requests.post("https://accounts.accesscontrol.windows.net/" +

                             settings["realm"] + "/tokens/OAuth/2", data=body).json()

Finally, we save the AccessToken in the settings dictionary.

settings["appReg\_bearerToken"] = response["access\_token"]

The request returns the response with the **AccessToken** we can see it in the **Python Terminal**. Here’s an example on hot the AccesToken looks like,



You can find an example of the usage of this Function in Reference

# 3. Working with Folders and Files with SharePoint REST

Great! Now, that you have an **AccessToken** we can start making requests to the SharePoint Rest API.

Keep in mind that **refresh tokens** are valid for **90 days**, and with continuous use, they can be valid until revoked. For more information please visit this [Session timeouts for Office 365](https://docs.microsoft.com/en-us/office365/enterprise/session-timeouts) documentation

## 3.1 Working with Files and Folder by using REST

The SharePoint REST service supports combining multiple requests into a single call to the service by using the OData $batch query option. For details and links to code samples, see [Make batch requests with the REST APIs](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/make-batch-requests-with-the-rest-apis).

Below we are going to work with **GET** and **POST** REST examples.

Keep in mind that the code samples below are working examples using Python 3.7.

### 3.1.1 Retrieve all the Files in a Folder.

The next code makes a request to the SharePoint Rest API and retrieve all the files in a folder.

{YourWorkDirectory}/SharePointRestAction.py:

def getAllFiles(settings, relativeURL, siteURL):

    header = {

        "Authorization": "Bearer " + settings["appReg\_bearerToken"],

        "Accept": "application/json; odata=verbose",

    }

    response = requests.delete("https://" + settings["targetHost"] + siteURL + "/\_api/web/GetFolderByServerRelativeUrl('" +

                               relativeURL + "')/Files", headers=header).json()

    print(response)

You can find an example of the usage of this Function in Reference

### 3.1.2 Retrieve a Specific File.

The next code makes a request to the SharePoint Rest API and retrieve a specific file in a folder.

def getFile(settings, fileName, relativeURL, siteURL):

    header = {

        "Authorization": "Bearer " + settings["appReg\_bearerToken"],

        "Accept": "application/json; odata=verbose",

    }

response = requests.get("https://" + settings["targetHost"] + siteURL + "/\_api/web/GetFolderByServerRelativeUrl('" +

                  relativeURL + "')/Files('" + fileName + "')/", headers=header).json()

    print(response)

You can find an example of the usage of this Function in Reference

### 3.1.3 Create a File and Add it to a Folder.

The next code makes a post request to the SharePoint Rest API that add a file to a folder  
If the file its larger than 10mb, the file its split into multiple parts and send with the   
/startupload, /continueupload and /finishupload endpoints

.

def uploadFile(settings, fileName, relativeURL, siteURL):

    offset = 0

    chunckSize = 3 \* 1024 \* 1024

    header = {

        "Authorization": "Bearer " + settings["appReg\_bearerToken"],

        "Accept": "application/json; odata=verbose",

    }

    with open("./" + fileName, "rb") as file:

        fileLength = os.path.getsize("./" + fileName)

        if (fileLength <= chunckSize):

            response = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/GetFolderByServerRelativeUrl('"   
+ relativeURL + "')/Files/add(url='" + fileName + "', overwrite=true)", headers=header, data=file).json()

            print(response)

        else:

            response = requests.post("https://" + settings["targetHost"]

+ siteURL +

                "/\_api/web/GetFolderByServerRelativeUrl('" +

                relativeURL + "')/Files/add(url='"

                + fileName + "', overwrite=true)", headers=header).json()

            print(response)

            uploadID = response['d']['UniqueId']

            first = True

            totalBytesRead = 0

            for chunk in read\_in\_chunks(file, chunckSize):

                totalBytesRead = totalBytesRead + len(chunk)

                if (first):

                    first = False

                    r = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/getfilebyserverrelativeurl('"   
+ siteURL + "/" + relativeURL + "/" + fileName + "')/startupload(uploadId=guid'" +

                   str(uploadID) + "')", headers=header, data=(chunk)).json()

print(r)

                elif (totalBytesRead == fileLength):

                    r = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/getfilebyserverrelativeurl('"   
+ siteURL + "/" + relativeURL + "/" + fileName + "')/finishupload(uploadId=guid'" +

str(uploadID) + "', fileOffset=" + str(offset) + ")",   
headers=header, data=(chunk)).json()

print(r)

                else:

                    r = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/getfilebyserverrelativeurl('"   
+ siteURL + "/" + relativeURL + "/" + fileName + "')/continueupload(uploadId=guid'" +

str(uploadID) + "', fileOffset=" + str(offset) + ")",   
headers=header, data=(chunk)).json()

print(r)

                offset += len(chunk)

                print("%" + "{:.2f}".format(offset \* 100 / fileLength) + " Completed")

You can find an example of the usage of this Function in Reference

### 3.1.4 Create a New Folder.

The next code makes a post request to the SharePoint Rest API that add a new folder.

# addFolder send a request to create a Folder

def addFolder(settings, relativeURL, siteURL, folderName):

    header = {

        "Authorization": "Bearer " + settings["appReg\_bearerToken"],

        "Accept": "application/json; odata=verbose",

        "Content-Type": "application/json;odata=verbose"

    }

    response = requests.post(

        "https://" + settings["targetHost"] + siteURL + "/\_api/Web/Folders/add('"  + relativeURL + "/" + folderName + "')", headers=header).json()

    print(response)

You can find an example of the usage of this Function in Reference

## 3.2 Working with large files by using REST

When you need to upload a binary file that is larger than 10 megabytes, you need to split the code in chunks and send it with the /startupload, /continueupload and /finishupload endpoints.

The next part of the uploadFile() Function shows when the File size its larger than 10mb, in that case the codes gets into the else part and performs the algorithm for upload the file in chunks.

def uploadFile(settings, fileName, relativeURL, siteURL):

.

.

.

        else:

            response = requests.post(

                "https://" + settings["targetHost"] + siteURL +

                "/\_api/web/GetFolderByServerRelativeUrl('" +

                relativeURL + "')/Files/add(url='"

                + fileName + "', overwrite=true)", headers=header).json()

            print(response)

            uploadID = response['d']['UniqueId']

            first = True

            totalBytesRead = 0

            for chunk in read\_in\_chunks(file, chunckSize):

                totalBytesRead = totalBytesRead + len(chunk)

                if (first):

                    first = False

                    r = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/getfilebyserverrelativeurl('"

+ siteURL + "/" + relativeURL + "/" + fileName + "')/startupload(uploadId=guid'" + str(uploadID) + "')", headers=header, data=(chunk)).json()

print(r)

                elif (totalBytesRead == fileLength):

                    r = requests.post("https://" + settings["targetHost"]

 + siteURL + "/\_api/web/getfilebyserverrelativeurl('"

+ siteURL + "/" + relativeURL + "/" + fileName + "')/finishupload(uploadId=guid'" + str(uploadID) + "', fileOffset="

+ str(offset) + ")", headers=header, data=(chunk)).json()

print(r)

                else:

                    r = requests.post("https://" + settings["targetHost"]

+ siteURL + "/\_api/web/getfilebyserverrelativeurl('"

+ siteURL + "/" + relativeURL + "/" + fileName + "')/continueupload(uploadId=guid'" + str(uploadID) + "', fileOffset=" + str(offset) + ")", headers=header, data=(chunk)).json()

print(r)

                offset += len(chunk)

                print("%" + "{:.2f}".format(offset \*100 / fileLength) + " Completed")

For more information about this case go to:

<https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/upload-large-files-sample-app-for-sharepoint>

# 4. References

In this section you will be able to find the sample code that was used on this documentation.

### Create the SharePointRestAction.py File

In the SharePointRestAction.py File we are going to create a settings object.

settings = {

    "appReg\_clientId": "d0812b3c-d362-4dad-9035-e0fd017fe895",

    "appReg\_clientSecret": "JP7yqSPQQOiZs8jyoHr6ks9pUz4j1n8+ncvlUOUam30=",

    "targetHost": "m365x423176.sharepoint.com",

    "principal": "00000003-0000-0ff1-ce00-000000000000",

    "realm": "1b14560c-1aa5-434c-b667-c28199260930",

    "appReg\_bearerToken": ""

}

We’ll be using the settings object in the requests to the SharePoint Rest API.

You can’t import this File in the SharePointRestAction.py File with the next code:

import SharePointRestAPISettings

### Get AccessToken

def getToken(settings):

    body = {

        "grant\_type": "client\_credentials",

        "client\_id": settings["appReg\_clientId"] + "@" + settings["realm"],

        "client\_secret": settings["appReg\_clientSecret"],

        "resource": settings["principal"] + "/" + settings["targetHost"] + "@" + settings["realm"]

    }

    response = requests.post("https://accounts.accesscontrol.windows.net/" +

                             settings["realm"] + "/tokens/OAuth/2", data=body).json()

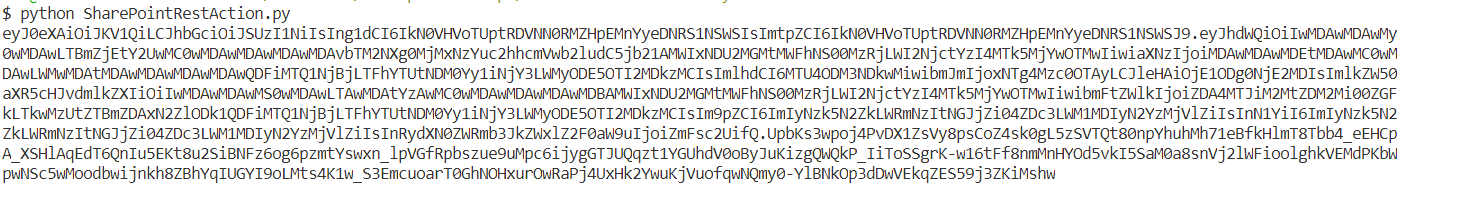
    settings["appReg\_bearerToken"] = response["access\_token"]

import SharePointRestAPISettings

# Example of usage (getToken)

getToken(SharePointRestAPISettings.settings)

Python Console:



### Retrieve all Files in a folder

import SharePointRestAPISettings

relative\_url = "Your Folder/"

site\_url = "/sites/Site Name"

getToken(SharepointSettings.settings)

# Example of usage (getAllFiles)

getAllFiles(SharePointRestAPISettings.settings, relative\_url, site\_url)

### Retrieve a specific File

import SharePointRestAPISettings

file\_name = "Example.txt"

relative\_url = "Your Folder/"

site\_url = "/sites/Site Name"

getToken(SharepointSettings.settings)

getFile(SharePointRestAPISettings.settings, file\_name, relative\_url, site\_url)

### Add a new Folder

import SharePointRestAPISettings

relative\_url = "Your Folder/"

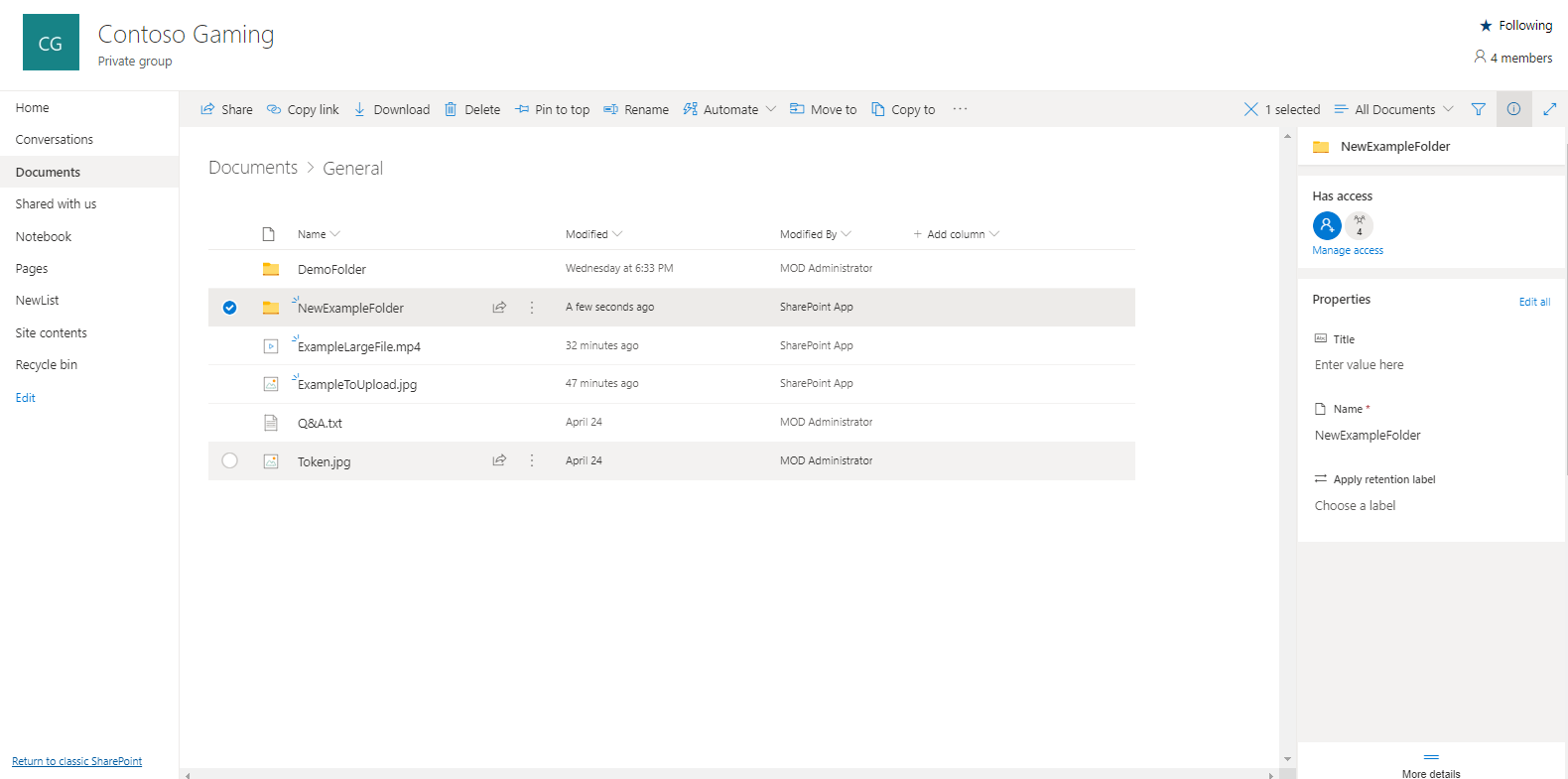
site\_url = "/sites/Site Name"

folder\_name = "NewExampleFolder"

getToken(SharepointSettings.settings)

# Example of usage (addFolder)

addFolder(SharePointRestAPISettings.settings, relative\_url, site\_url, folder\_name)



### Add a new File inside a Folder (File size less than 10mb)

import SharePointRestAPISettings

relative\_url = "Your Folder/"

site\_url = "/sites/Site Name

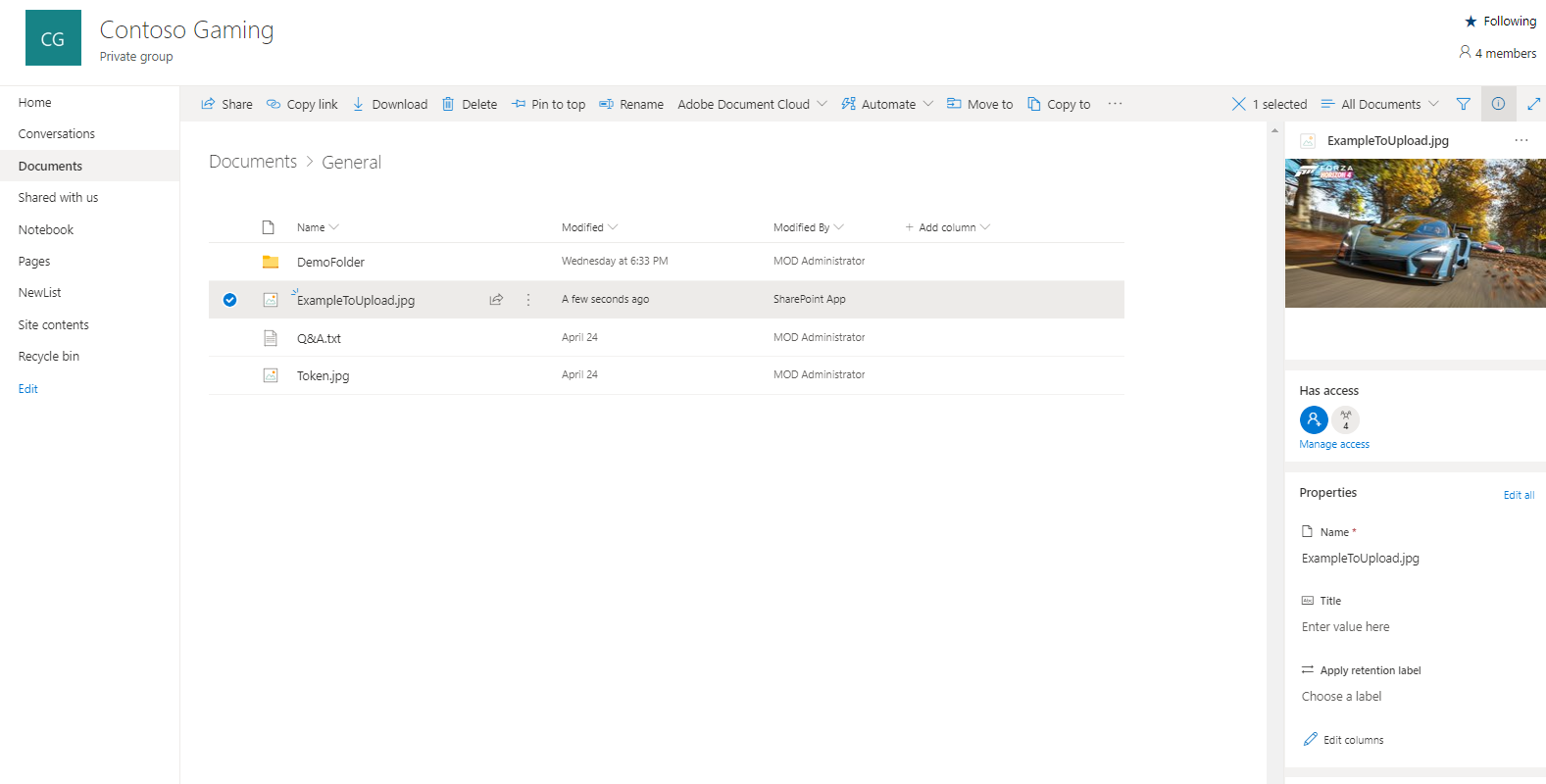
file\_to\_upload = "ExampleToUpload.jpg" # File size less than 10mb.

getToken(SharepointSettings.settings)

# Example of usage (uploadFile) File size less than 10mb

uploadFile(SharePointRestAPISettings.settings,

           file\_to\_upload, relative\_url, site\_url)



### Add a large File (more than 10mb) inside a Folder

import SharePointRestAPISettings

relative\_url = "Your Folder/"

site\_url = "/sites/Site Name

large\_file\_to\_upload = "ExampleLargeFile.mp4" # File size larger than 10mb.

getToken(SharepointSettings.settings)

# Example of usage (uploadFile) File size larger than 10mb

uploadFile(SharePointRestAPISettings.settings,

           large\_file\_to\_upload, relative\_url, site\_url)

