# How to upload file to OneDrive and to share using web application

## Introduction

This sample demonstrates how to upload file to OneDrive and to get a share link to share with REST using web application.

When users visit the website, they will be redirected to office 365 to finish the authentication. After that, they will go forward to our website with a code so that they can request REST API for a token.

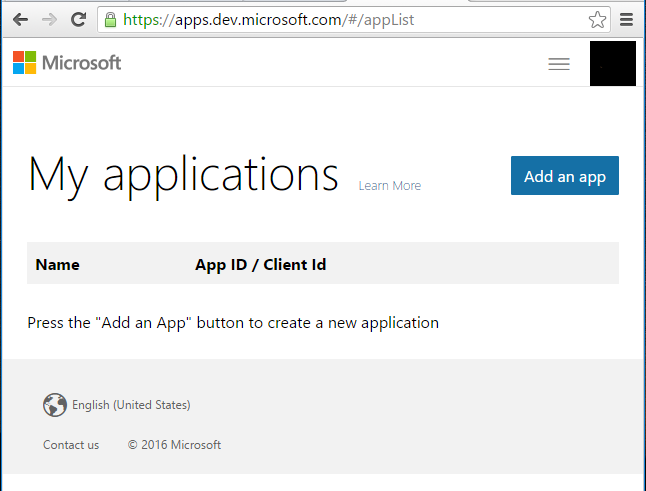
Next, users will upload the file to OneDrive and receive a file id in return.

Eventually, REST API can be requested with this file id to get the share link which can be used in web, client or mobile.

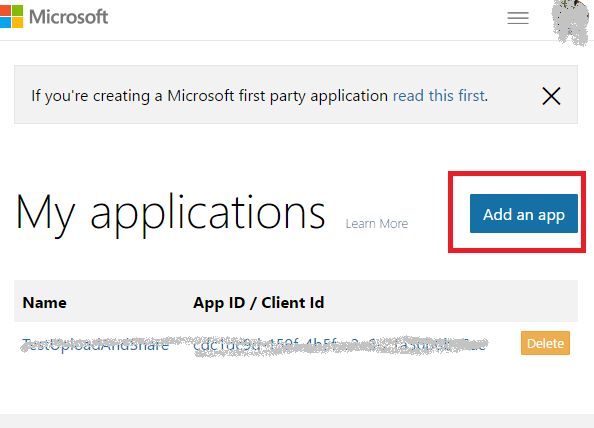
## Sample prerequisites

* Register application for your OneDrive, while the related details will be described in the next section
* Go to the [Register and manage apps](https://apps.dev.microsoft.com/) to register your application.
* After the registration has been prompted, sign in with you Microsoft account credentials in [apps.dev.microsoft.com](https://apps.dev.microsoft.com/).

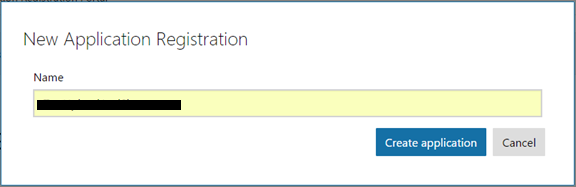
And you will see this in your browser.

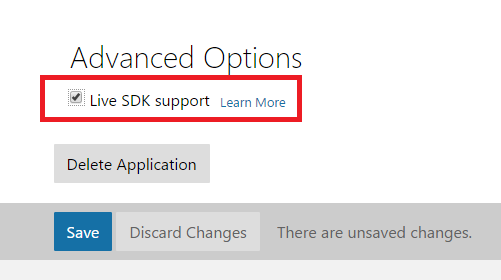


* Find **My applications** and click **Add an app**.

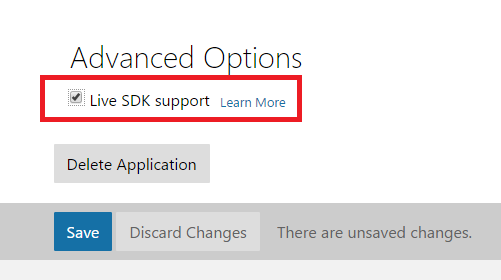


* Enter you app’s name and click **Create application**.

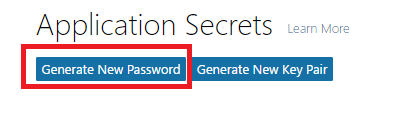


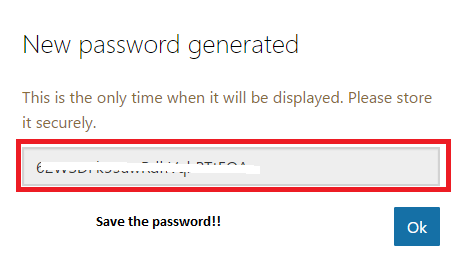


* Scroll to the page bottom and check the **Live ADK support** box.

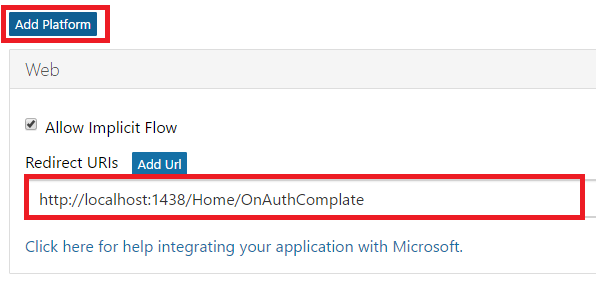


* Below the **Application Secrets**, generate new password, and then save it for later use.





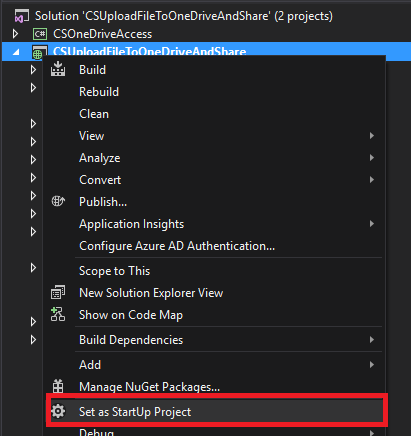
* Below the Platform's header, create a web app, and then set the Redirect URIs to your web app callback address such as <http://localhost:1438/Home/OnAuthComplate>.



* Click **Save** at the very bottom of the page.

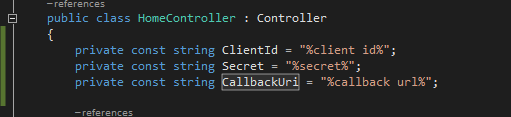
## Building the sample

* Double-click **CSUploadFileToOneDriveAndShare.sln** file to open this sample solution using Microsoft Visual Studio 2015 which has the web develop component installed.
* Set project **CSUploadFileToOneDriveAndShare** as a startup project.

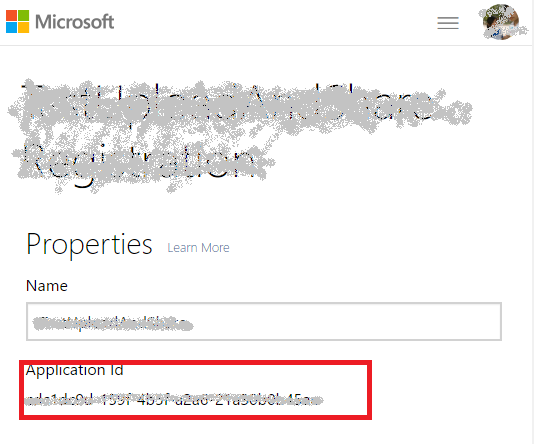


* Config under parameter in:

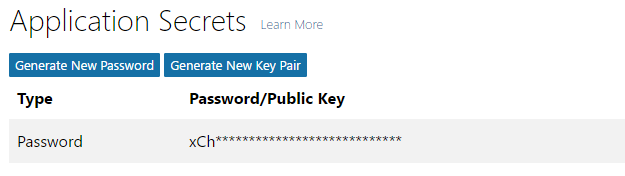
Project: **CSUploadFileToOneDriveAndShare/Controllers/HomeController.cs**



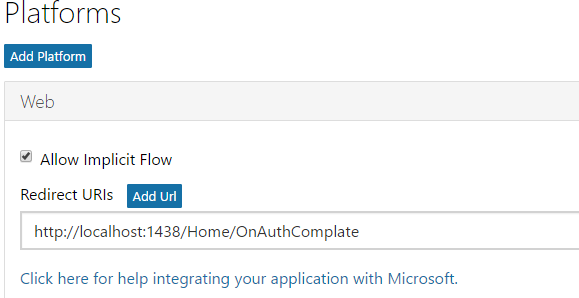
You can find the Clirntld here:



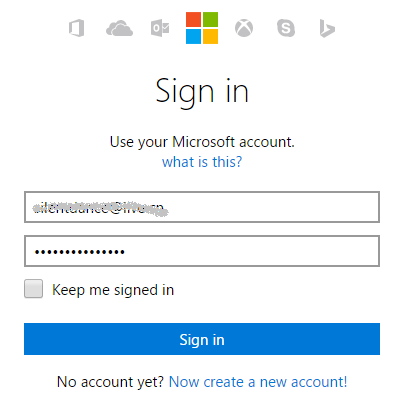
Secret is the key for the application whose password has been shown **once** when being set up. Therefore, it is better for you to copy it.

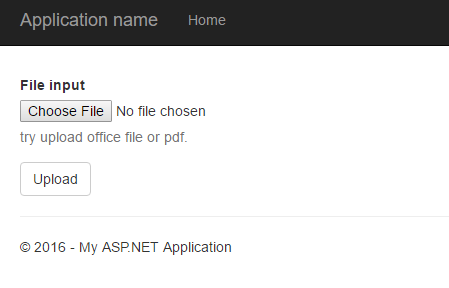


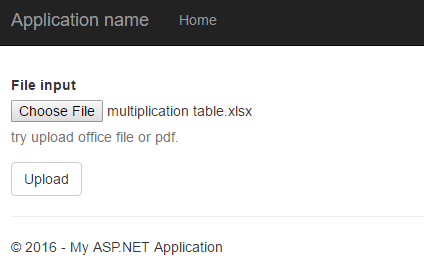
You can find CallbackUri here:



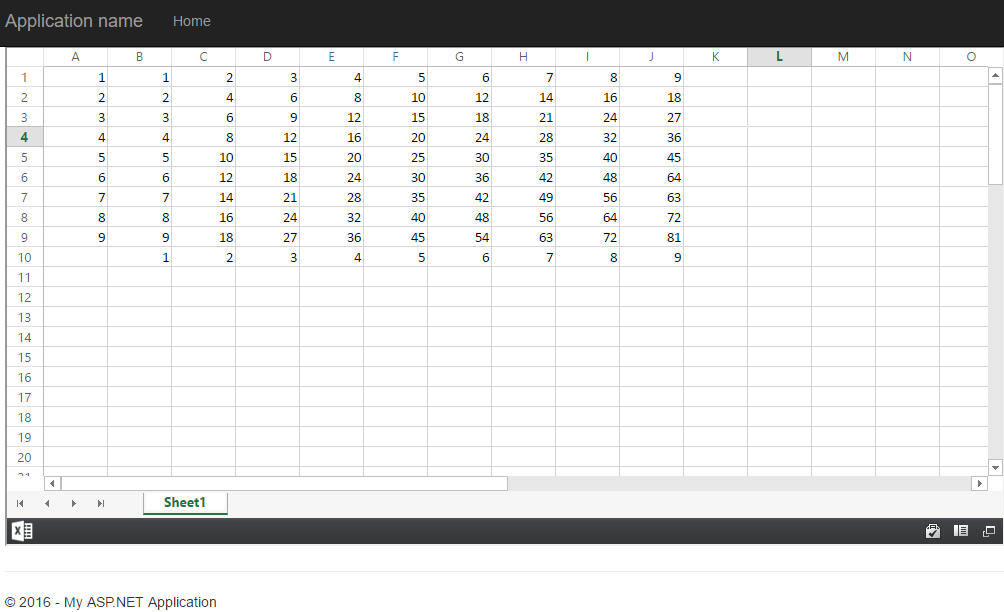
## Running the sample

* Open the sample solution using Visual studio, then press **F5 key** or select **Debug -> Start Debugging** from the menu.
* After the site has been started, you will see this:
* 
* When you have filled all the fields, click Sign in. Then it will go back to our web application, then you can see this:



* Choose an office file, and then click upload button.
* 

The file will be uploaded to OneDrive, and a preview page will be rendered.



## Using the code

**Public field**

public string OneDriveApiRoot { get; set; } = "https://api.onedrive.com/v1.0/";

**Upload file to OneDrive**

//this is main method of upload file to OneDrive

public async Task<string> UploadFileAsync(string filePath, string oneDrivePath)

{

//get the upload session,we can use this session to upload file resume from break point

string uploadUri = await GetUploadSession(oneDrivePath);

//when file upload is not finish, the result is upload progress,

//When file upload is finish, the result is the file information.

string result = string.Empty;

using (FileStream stream = File.OpenRead(filePath))

{

long position = 0;

long totalLength = stream.Length;

int length = 10 \* 1024 \* 1024;

//In one time, we just upload a part of file

//When all file part is uploaded, break out in this loop

while (true)

{

//Read a file part

byte[] bytes = await ReadFileFragmentAsync(stream, position, length);

//check if arrive file end, when yes, break out with this loop

if (position >= totalLength)

{

break;

}

//Upload the file part

result = await UploadFileFragmentAsync(bytes, uploadUri, position, totalLength);

position += bytes.Length;

}

}

return result;

}

private async Task<string> GetUploadSession(string oneDriveFilePath)

{

var uploadSession = await AuthRequestToStringAsync(

uri: $"{OneDriveApiRoot}drive/root:/{oneDriveFilePath}:/upload.createSession",

httpMethod: HTTPMethod.Post,

contentType: "application/x-www-form-urlencoded");

JObject jo = JObject.Parse(uploadSession);

return jo.SelectToken("uploadUrl").Value<string>();

}

private async Task<string> UploadFileFragmentAsync(byte[] datas, string uploadUri, long position, long totalLength)

{

var request = await InitAuthRequest(uploadUri, HTTPMethod.Put, datas, null);

request.Request.Headers.Add("Content-Range", $"bytes {position}-{position + datas.Length - 1}/{totalLength}");

return await request.GetResponseStringAsync();

}

**Get Share Link**

//This method use to get ShareLink, you can use the link in web or client terminal

public async Task<string> GetShareLinkAsync(string fileID, OneDriveShareLinkType type, OneDrevShareScopeType scope)

{

string param = "{type:'" + type + "',scope:'" + scope + "'}";

string result = await AuthRequestToStringAsync(

uri: $"{OneDriveApiRoot}drive/items/{fileID}/action.createLink",

httpMethod: HTTPMethod.Post,

data: Encoding.UTF8.GetBytes(param),

contentType: "application/json");

return JObject.Parse(result).SelectToken("link.webUrl").Value<string>();

}

## More information

* Upload large file to onedrive using REST API: <https://dev.onedrive.com/items/upload_large_files.htm>
* Get share link from onedrive using REST API: <https://dev.onedrive.com/items/sharing_createLink.htm>