# How to log in office 365 and get account info with REST API using web application

## Introduction

This sample demonstrates how to log in office 365 and get the account information with REST using web application.

When users visit the website, they will be redirected to office 365 to finish the authentication. Then, they will go forward to our website with a code to request REST API for a token. At last, they will get the user account information and picture with the token.

## Sample prerequisites

* Register the 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/Disambiguation?ru=https%3a%2f%2fapps.dev.microsoft.com%2f) to register you application.
* When prompted, sign in with you Microsoft account credentials.
* Find **My applications** and click **Add an app**.

{1}

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

{2}

{3}

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

{4}

* Generate the new password below the **Application Secrets**, and save it for later use.

{5}

{6}

* Create a web app below the Platforms header, and then set the Redirect URIs to you web app callback address such as <http://localhost:1438/Home/OnAuthComplate>.

{7}

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

## Building the sample

* Double-Click **CSLoginOnOffice365AndGetAccountInfo.sln** file to open this sample solution using Microsoft Visual Studio 2015 which has the web develop component installed.
* Set project LoginOnOffice365AndGetAccountInfo as startup project.

{8}

* Config the following parameter in:

Project: **LoginOnOffice365AndGetAccountInfo/Controllers/HomeController.cs**

{9}

About the ClientId you can find it here:

{10}

Secret is the key for the application where you can set password only once after it shows up.

{11}

For CallbackUri, you can find it here:

{12}

## Running the sample

* Open the sample solution using Visual Studio, then press **F5 key** or select **Debug -> Start Debugging** in menu.
* When the site is started, you can see this:

{13}

After you have filled all the fields and clicked the Sign in button, you can see this:

{14}

## Using the code

The Office 365 Rest API access base class

public abstract class OAuthAccessBase

{

public string ClientId { get; }

public string ClientSecret { get; }

//when user complated the authenticate, will retrun this code

public string AccessCode { get; protected set; }

//use this token to request Office 365 API

public string AccessToken { get; protected set; }

//when accessToken had expires, can use this token to refresh accessToken

public string RefreshToken { get; protected set; }

public string UserId { get; protected set; }

public DateTime RefreshTime { get; protected set; }

public TimeSpan RefreshTimeSpan { get; protected set; }

public string RedirectURI { get; set; }

public OAuthAccessBase(string clientId, string clientSecret, string redirectURI)

{

this.ClientId = clientId;

this.ClientSecret = clientSecret;

this.RedirectURI = redirectURI;

}

//retrun authenticate url for redirect

public string GetLoginUrl(string scopes)

{

string urlStr =

"https://login.live.com/oauth20\_authorize.srf" +

"?client\_id=" + ClientId +

"&scope=offline\_access " + scopes +

"&response\_type=code" +

"&redirect\_uri=" + RedirectURI;

return urlStr.ToString();

}

//get token use code

public async Task RedeemTokensAsync(string code)

{

this.AccessCode = code;

string url = "https://login.live.com/oauth20\_token.srf";

string paramStr =

"client\_id=" + ClientId +

"&redirect\_uri=" + RedirectURI +

"&client\_secret=" + ClientSecret +

"&code=" + AccessCode +

"&grant\_type=authorization\_code";

APIRequest request = GetRequest(url, HTTPMethod.Post, paramStr.ToString());

string response = await request.GetResponseToStringAsync();

JObject jo = JObject.Parse(response);

this.RefreshToken = jo.SelectToken("refresh\_token").Value<string>();

this.AccessToken = jo.SelectToken("access\_token").Value<string>();

this.UserId = jo.SelectToken("user\_id").Value<string>();

this.RefreshTimeSpan = new TimeSpan(0, 0, Convert.ToInt32(jo.SelectToken("expires\_in").Value<string>()));

this.RefreshTime = DateTime.Now;

}

protected async Task<string> AuthRequestToStringAsync(string url, HTTPMethod httpMethod = HTTPMethod.Get, string data = "")

{

await RefreshTokenIfNeededAsync();

APIRequest request = GetRequest(url, httpMethod, data);

return await request.GetResponseToStringAsync();

}

protected async Task<byte[]> AuthRequestToBytesAsync(string url, HTTPMethod httpMethod = HTTPMethod.Get, string data = "")

{

await RefreshTokenIfNeededAsync();

APIRequest request = GetRequest(url, httpMethod, data);

return await request.GetResponseTobytesAsync();

}

private APIRequest GetRequest(string url, HTTPMethod httpMethod = HTTPMethod.Get, string data = "")

{

APIRequest apiRequest = new APIRequest(url, httpMethod, data);

if (!string.IsNullOrEmpty(AccessToken))

{

apiRequest.Request.Headers.Add("Authorization", "bearer " + AccessToken);

}

return apiRequest;

}

private async Task RefreshTokenIfNeededAsync()

{

if (RefreshTimeSpan == null || DateTime.Now - RefreshTime > RefreshTimeSpan)

{

string url = "https://login.live.com/oauth20\_token.srf";

string paramStr = "client\_id=" + ClientId +

"&redirect\_uri=" + RedirectURI +

"&client\_secret=" + ClientSecret +

"&refresh\_token=" + RefreshToken +

"&grant\_type=refresh\_token";

APIRequest request = GetRequest(url, HTTPMethod.Post, paramStr.ToString());

string response = await request.GetResponseToStringAsync();

JObject jo = JObject.Parse(response);

this.RefreshToken = jo.SelectToken("refresh\_token").Value<string>();

this.AccessToken = jo.SelectToken("access\_token").Value<string>();

this.UserId = jo.SelectToken("user\_id").Value<string>();

this.RefreshTimeSpan = new TimeSpan(0, 0, Convert.ToInt32(jo.SelectToken("expires\_in").Value<string>()));

this.RefreshTime = DateTime.Now;

}

}

}

The Office 365 Rest API access class:

public class OAuthAccess : OAuthAccessBase

{

public OAuthAccess(string clientId, string clientSecret, string redirectURI) : base(clientId, clientSecret, redirectURI)

{

}

public async Task<Dictionary<string, string>> GetAccountInfoAsync()

{

string response = await AuthRequestToStringAsync("https://apis.live.net/v5.0/me?suppress\_response\_codes=true&suppress\_redirects=true");

JObject jo = JObject.Parse(response);

return jo.ToObject<Dictionary<string, string>>();

}

public async Task<byte[]> GetAccountPicture()

{

byte[] response = await AuthRequestToBytesAsync("https://apis.live.net/v5.0/me/picture");

return response;

}

}

The Controller class:

public class HomeController : Controller

{

private const string ClientId = "cdc1dc9d-159f-4b5f-a2a6-21a30b0b45ae";

private const string Secret = "xChUsnbQthVgYN5VoAjEcYy";

private const string CallbackUri = "http://localhost:1438/Home/OnAuthComplate";

public OAuthAccess OfficeAccess

{

get

{

var officeAccess = Session["OfficeAccess"];

if (officeAccess == null)

{

officeAccess = new OAuthAccess(ClientId, Secret, CallbackUri);

Session["OfficeAccess"] = officeAccess;

}

return officeAccess as OAuthAccess;

}

}

public async Task<ActionResult> Index()

{

//if user is not login, redirect to office 365 for authenticate

if (string.IsNullOrEmpty(OfficeAccess.AccessCode))

{

string url = OfficeAccess.GetLoginUrl("onedrive.appfolder");

return new RedirectResult(url);

}

//when user is authenticated get user account info

ViewBag.UserInfo = await OfficeAccess.GetAccountInfoAsync();

return View();

}

//when user complate authenticate, will be call back this url with a code

public async Task<RedirectResult> OnAuthComplate(string code)

{

//get token by the code

await OfficeAccess.RedeemTokensAsync(code);

return new RedirectResult("Index");

}

//download user picture

public async Task<ActionResult> UserPicture()

{

var btyes = await OfficeAccess.GetAccountPicture();

return base.File(btyes, "image/jpeg");

}

}

## More information

Onedrive develp documentation: <https://dev.onedrive.com/getting-started.htm>