NOKIA Developer

Facebook authentication in Windows Phone application

This article demonstrates how to implement Facebook OAuth2.0 authentication and fetch Facebook data without using any third party library.

13 Aug 2012

Note: If you do want to use a third party library, the Facebook C# SDK is a popular choice. It supports a number of platforms and is licensed under the permissive Apache Licence v2.0

Introduction

Facebook OAuth2.0 authentication enables third party application to access user data using Facebook API. In this article we will create an application which will pass Facebook OAuth2.0 authentication to access user profile information.

Facebook Login Dialog

Facebook User Profile

Implementation

Create an empty Windows Phone Project

Launch **Visual Studio** -> click on **File->New Project**-> select **Windows Phone Application** (Visual C# Template) -> add **Name** and **Location** of the project -> click **OK** to create the project.

Create a Facebook App and get the API Key and App Secret

To create a Facebook App lets go to Facebook Developer account and click on **Create New App** button. Follow the instructions on the page to complete the process. Once the app is created on your Facebook account, you will see *App ID/API Key* and *App Secret* on the Facebook application page. We will use these *Key* and *Secret* to authenticate our application to access Facebook user profile information.

Adding controls on application page

In our application we have two pages **MainPage.xaml** and **FacebokLoginPage.xaml**. **MainPage.xaml** consist of one Button, TextBlock and a Image control.

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
     <Image Height="82" Margin="66,100,0,0" Name="image1" Stretch="Fill"
VerticalAlignment="Top" ImageFailed="image1_ImageFailed" HorizontalAlignment="Left"
Width="76" />
     <TextBlock Height="68" HorizontalAlignment="Left" Margin="153,100,0,0"
Name="textBlock1" Text="" VerticalAlignment="Top" Width="290" FontSize="42" />
     <Button Content="Login" Height="72" HorizontalAlignment="Left" Margin="66,421,0,0"
Name="button1" VerticalAlignment="Top" Width="309" Click="buttons_Click" />
</Grid>
```

The Button control in the **MainPage.xaml** is used to call the **FacebokLoginPage.xaml** which has a WebBrowser control to load the Facebook login page.

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
     <phone:WebBrowser IsScriptEnabled="True" HorizontalAlignment="Left"
Margin="9,20,0,0" x:Name="webBrowser1" VerticalAlignment="Top" Height="614" Width="441"
Navigated="BrowserNavigated"/>
</Grid>
```

When the Facebook login is successful the **FacebookLoginPage.xaml** page gets closed and the application control goes back to **MainPage.xaml** which displays user's name and profile picture.

Launching Facebook login page

When user clicks on *Login Button* on the **MainPage.xaml** it navigates to **FacebookLoginPage.xaml** which calls the Facebook login *URL* and launch the Facebook login dialog.

```
https://www.facebook.com/dialog/oauth?client_id=<Your
Key>&redirect_uri=https://www.facebook.com/connect/login_success.html
```

After user enters Facebook credentials it will ask for permissions which can be done with the *scope* parameter. So our *URL* should look like below.

```
https://www.facebook.com/dialog/oauth?client_id=<Your
Key>&redirect_uri=https://www.facebook.com/connect/login_success.html&scope=email,user_location,friends_location,user_hometown,friends_hometown,publish_stream,offline_access,read_stream,user_status,user_photos,friends_photos,friends_status,user_checkins,friends_checkins,user_events,publish_checkins&display=wap
```

Note: There is extra permissions used in this application. You can manage permissions according to your use.

Get the Code and Access Token

The login page is an HTML page, so we load it on WebBrowser control.

```
void FB_LoginPage_Loaded(object sender, RoutedEventArgs e)
{
var url = "https://www.facebook.com/dialog/oauth?client_id=<Your
Key>&redirect_uri=https://www.facebook.com/connect/login_success.html&scope=email,user_location,friends_location,user_hometown,friends_hometown,publish_stream,offline_access,read_stream,user_status,user_photos,friends_photos,friends_status,user_checkins,friends_checkins,user_events,publish_checkins&display=wap";
webBrowser1.Navigate(new Uri(url));
}
```

Once the application is authenticated it will send back us a response with a *code*. We get the response in `BrowserNavigated()` which then parse the *code* and add the *code* to the *URL* to get the access token.

```
private void BrowserNavigated(object sender,
System.Windows.Navigation.NavigationEventArgs e)
{
            if (e.Uri.IsAbsoluteUri)
            {
                string code = e.Uri.Query.ToString();
                string[] split = code.Split(new Char[] { '=' });
                string codeString = split.GetValue(0).ToString();
                string codeValue = split.GetValue(1).ToString();
                if (codeValue.Length > 0)
                {
                    var url = "https://graph.facebook.com/oauth/access_token?
client_id=<Your Key>
&redirect_uri=https://www.facebook.com/connect/login_success.html&client_secret=<Your
Secret>&code=" + codeValue;

                    //call access token
                    WebRequest request = WebRequest.Create(url); //FB access token Link
                    request.BeginGetResponse(new
AsyncCallback(this.ResponseCallback_AccessToken), request);
                }
            }
            else
                return;
}
```

An asynchronous web request is called to get the access token.

```
//call access token
WebRequest request = WebRequest.Create(url); //FB access token Link
request.BeginGetResponse(new AsyncCallback(this.ResponseCallback_AccessToken), request);
```

We get the web response and parse the access token

```
private void ResponseCallback_AccessToken(IAsyncResult asynchronousResult)
{
      try
      {
                var request = (HttpWebRequest)asynchronousResult.AsyncState;
                using (var resp =
(HttpWebResponse)request.EndGetResponse(asynchronousResult))
                {
                    using (var streamResponse = resp.GetResponseStream())
                    {
                        using (var streamRead = new StreamReader(streamResponse))
                        {
                            string responseString = streamRead.ReadToEnd();
                            string[] splitAccessToken = responseString.Split(new Char[]
{ '=' });
                            string accessTokenString =
splitAccessToken.GetValue(0).ToString();
                            string accessTokenValue =
splitAccessToken.GetValue(1).ToString();
                            if (accessTokenString == "access_token")
                            {
                                AccessToken = accessTokenValue;
                            }
                        }
                    }
                }
        }
        catch (WebException ex)
        {

        }
         GetAccessToken();
}
```

`GetAccessToken()` method checks the existence of access token and calls the `LoadUserProfile()` method.

```
void GetAccessToken()
 {
            Deployment.Current.Dispatcher.BeginInvoke(() =>
            {
                if (string.IsNullOrEmpty(AccessToken))
                {
                    MessageBox.Show("AccessToken not valid");
                }
                else
                {
                    MessageBox.Show("AccessToken= " + AccessToken);
                    LoadUserProfile();
                }
            });
 }
```

Load Facebook User Profile

(C) Copyright Nokia 2014. All rights reserved.

`LoadUserProfile()` method will put a web request to get user name and profile picture using Facebook Graph API.

```
void LoadUserProfile()
 {
            var urlProfile = "https://graph.facebook.com/me?
fields=name,picture&access_token=" + AccessToken;
            //call access token
            WebRequest request = WebRequest.Create(urlProfile); //FB access token Link
            request.BeginGetResponse(new AsyncCallback(this.ResponseCallbackProfile),
request);
 }
```

We catch the asynchronous call at `ResponseCallbackProfile()` and parse the name and profile picture of the user and displays on the **MainPage.xaml**.

```
private void ResponseCallbackProfile(IAsyncResult asynchronousResult)
{
       try
       {
                var request = (HttpWebRequest)asynchronousResult.AsyncState;
                using (var resp =
(HttpWebResponse)request.EndGetResponse(asynchronousResult))
                {
                    using (var streamResponse = resp.GetResponseStream())
                    {
                        var facebookSerializerData = new
DataContractJsonSerializer(typeof(FacebookUserProfile));
                        var facebookProfileData =
facebookSerializerData.ReadObject(streamResponse) as FacebookUserProfile;
                        this.Dispatcher.BeginInvoke(
                            (Action<FacebookUserProfile>)((facebookUserData) =>
                            {
                                thisApp.UserName = facebookUserData.Name;
                                thisApp.UserImage = facebookUserData.Picture;
                                NavigationService.GoBack();
                            }), facebookProfileData);
                        }
                }
        }
       catch (WebException ex)
       {

       }
}
```

Note: To run the code make sure you have replaced **<Your Key>** with **API Key** and **<Your Secret>** with **App Secret** that you got from Facebook Developer account in the project.

Summary

This application shows how to use the Facebook API to authenticate our application without using any third party library. The same process of Facebook authentication has been implemented in Qt/QML applications also.

- Facebook+ Facebook client with Nokia Maps - app showcase
- Display Facebook Friend position with Nokia Maps API for Qt

Related Article

- How to use Facebook Graph API in Windows Phone

Source Code

- The full source code of the example is available here: File:FacebookLoginWp7.zip
- The updated version of the source code is available here: File:FacebookLoginWp7 v1.1.zip

(C) Copyright Nokia 2014. All rights reserved.