# Integrate Facebook to Your Windows Phone Application

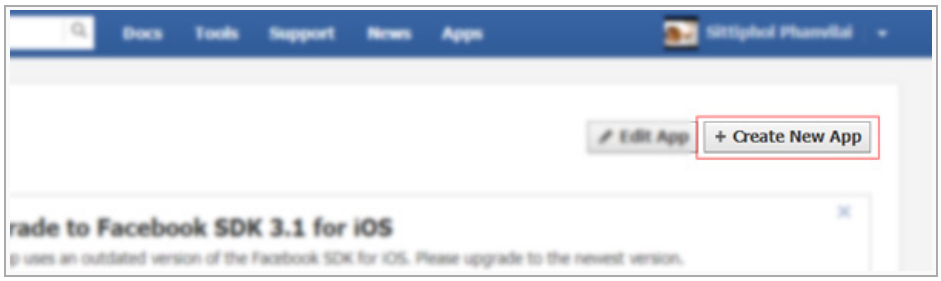
This article explains how to integrate Facebook to your Windows Phone Application.

## Introduction

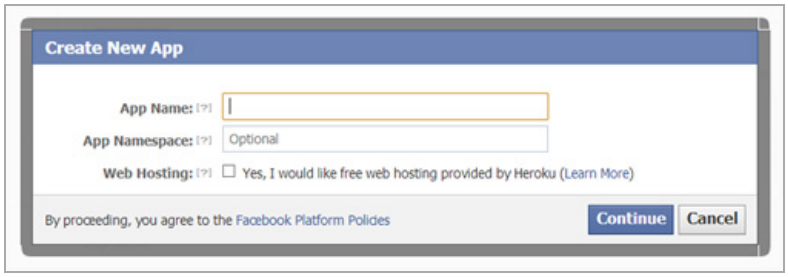
Facebook is now the most popular social network on earth. You can get the benefit from integrating Facebook to your application for example gaining more user, publish thing in the name of your application, etc. This article will show you how to easily integrate Facebook to your Windows Phone 8 application.

## Create Facebook App

First of all, you need to create Facebook Application on website. Here is the link to do so https://developers.facebook.com/apps
Click at **"Create New App"** button

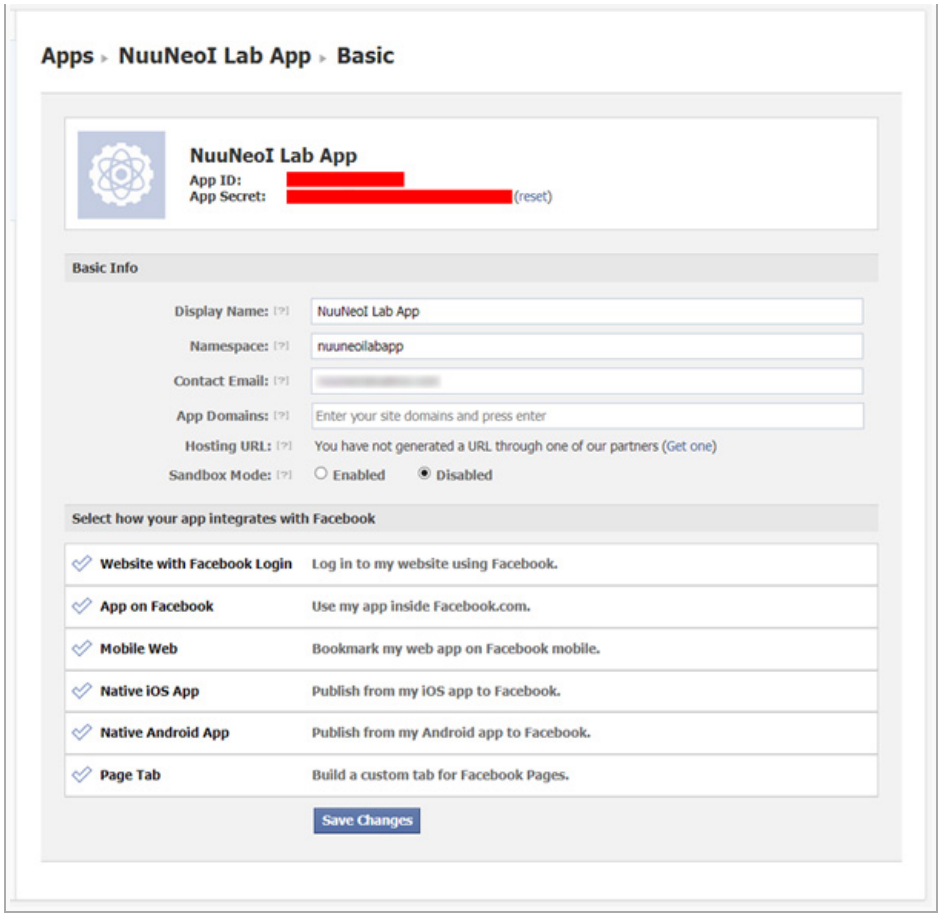



Enter the App Name and namespace (Optional)




Note down the **App ID** and **App Secret**. You will need them in coding part.


(C) Copyright Nokia 2014. All rights reserved.




# Set up the Project

## Prepare the Pages

Place the **Connect/Disconnect** as well as **Post Message on Wall** button on MainPage. To connect your application to Facebook, you need to press **Connect** button. Application will navigate to next page, **ConnectPage**

MainPage.xaml

(C) Copyright Nokia 2014. All rights reserved.

The ConnectPage is just a simple XAML page with **WebBrowser** embedded inside

ConnectPage.xaml

```
<phone:WebBrowser x:Name="mWebBrowser"
 HorizontalAlignment="Left"
 VerticalAlignment="Top"
 Height="696"
 Width="480"
 IsScriptEnabled="True"
 Navigated="WebBrowser_Navigated"/>
```

## Add Library

You need 4 small files below to work together. For the first file **FacebookClient.cs**, you need to put the **App ID** and **App Secret** to the **appId** and **clientSecret** variables to make these things work.

```
// FacebookUtils/FacebookClient.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.IO.IsolatedStorage;

namespace FacebookUtils
{
    public class FacebookClient
    {
        private static FacebookClient instance;
        private string accessToken;

        private static readonly IsolatedStorageSettings appSettings =
IsolatedStorageSettings.ApplicationSettings;

        private String appId = "xxx";
        private String clientSecret = "xxx";
        private String scope = "publish_stream";

        public FacebookClient()
        {
```


(C) Copyright Nokia 2014. All rights reserved.

```
            try
            {
                accessToken = (string)appSettings["accessToken"];
            }
            catch (KeyNotFoundException e)
            {
                accessToken = "";
            }
        }

        public static FacebookClient Instance
        {
            get
            {
                if (instance == null)
                    instance = new FacebookClient();
                return instance;
            }
            set
            {
                instance = value;
            }
        }

        public string AccessToken
        {
            get
            {
                return accessToken;
            }
            set
            {
                accessToken = value;
                if (accessToken.Equals(""))
                    appSettings.Remove("accessToken");
                else
                    appSettings.Add("accessToken", accessToken);
            }
        }

        public virtual String GetLoginUrl()
        {
            return "https://m.facebook.com/dialog/oauth?client_id=" + appId +
"&redirect_uri=https://www.facebook.com/connect/login_success.html&scope=" + scope +
"&display=touch";
        }

        public virtual String GetAccessTokenRequestUrl(string code)
        {
            return "https://graph.facebook.com/oauth/access_token?client_id=" + appId +
"&redirect_uri=https://www.facebook.com/connect/login_success.html&client_secret=" +
clientSecret + "&code=" + code;
        }

        public virtual String GetAccessTokenExchangeUrl(string accessToken)
        {
            return "https://graph.facebook.com/oauth/access_token?client_id=" + appId +
```


(C) Copyright Nokia 2014. All rights reserved.

```
"&client_secret=" + clientSecret + "&grant_type=fb_exchange_token&fb_exchange_token=" +
accessToken;
        }

        public void PostMessageOnWall(string message, UploadStringCompletedEventHandler
handler)
        {
            WebClient client = new WebClient();
            client.UploadStringCompleted += handler;
            client.UploadStringAsync(new Uri("https://graph.facebook.com/me/feed"),
"POST", "message=" + HttpUtility.UrlEncode(message) + "&access_token=" +
FacebookClient.Instance.AccessToken);
        }

        public void ExchangeAccessToken(UploadStringCompletedEventHandler handler)
        {
            WebClient client = new WebClient();
            client.UploadStringCompleted += handler;
            client.UploadStringAsync(new
Uri(GetAccessTokenExchangeUrl(FacebookClient.Instance.AccessToken)), "POST", "");
        }

    }
}
```

```
// FacebookUtils/ResponseData.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FacebookUtils
{
    public class ResponseData
    {
        public string id { get; set; }
        public ErrorData error { get; set; }
    }

    public class ErrorData
    {
        public int code { get; set; }
        public int error_subcode { get; set; }
    }
}
```

```
// Tools/UriToolKits.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
```

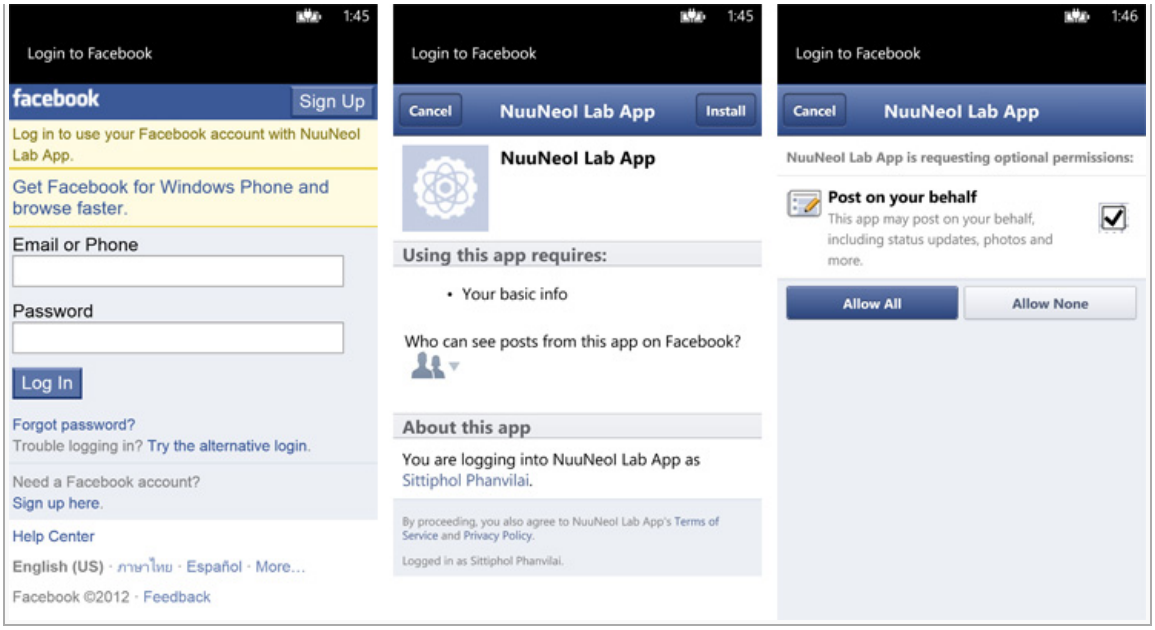

(C) Copyright Nokia 2014. All rights reserved.

```
namespace Tools
{
    public static class UriToolKits
    {
        private static readonly Regex QueryStringRegex = new Regex(@"[\?&](?
<name>[^&=]+)=(?<value>[^&=]+)");

        public static IEnumerable<KeyValuePair<string, string>> ParseQueryString(this
string uri)
        {
            if (uri == null)
                throw new ArgumentException("uri");

            var matches = QueryStringRegex.Matches(uri);
            for (var i = 0; i < matches.Count; i++)
            {
                var match = matches[i];
                yield return new KeyValuePair<string,
string>(match.Groups["name"].Value, match.Groups["value"].Value);
            }
        }
    }
}
```

```
// Tools/KeyValuePairUtils.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Tools
{
    public static class KeyValuePairUtils
    {
        public static TValue GetValue<TKey, TValue>(this IEnumerable<KeyValuePair<TKey,
TValue>> pairs, TKey key)
        {
            foreach (KeyValuePair<TKey, TValue> pair in pairs)
            {
                if (key.Equals(pair.Key))
                    return pair.Value;
            }

            throw new Exception("the value is not found in the dictionary");
        }
    }
}
```

## Work with Facebook Login Page


(C) Copyright Nokia 2014. All rights reserved.




Once **Connect** button is pressed, application will navigate to **ConnectPage**. Once this page is started, you need to set the source of WebBrowser by the url prepared by FacebookClient. Please note that you need to clear cookies before set the source to clear the logged in user data.

```
public ConnectPage()
{
            InitializeComponent();

            // Clear Cookie to remove current logged in user data
            mWebBrowser.ClearCookiesAsync();

            // Go to Login url
            mWebBrowser.Source = new Uri(FacebookClient.Instance.GetLoginUrl());
}
```

Now all you need to do is to snip the current url of WebBrowser using **Navigated** event. If logging-in process has been done, url will finally end up like "http://www.facebook.com/connect/login_success.html" or "https://www.facebook.com/connect/login_success.html". Once you got it, extract the **code** parameter out from query string to process to next step.

```
private void WebBrowser_Navigated(object sender, NavigationEventArgs e)
{
            String uri = e.Uri.ToString();

            if (uri.StartsWith("https://www.facebook.com/connect/login_success.html") ||
uri.StartsWith("http://www.facebook.com/connect/login_success.html"))
            {
                // Remove junk text added by facebook from url
                if (uri.EndsWith("#_=_"))
                    uri = uri.Substring(0, uri.Length - 4);

                String queryString = e.Uri.Query.ToString();

                // Acquire the code from Query String
                IEnumerable<KeyValuePair<string, string>> pairs =
UriToolKits.ParseQueryString(queryString);
                string code = KeyValuePairUtils.GetValue(pairs, "code");
```


(C) Copyright Nokia 2014. All rights reserved.

```
                // Get access_token from code using Asynchronous HTTP Request
                WebClient client = new WebClient();
                client.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(AccessTokenDownloadCompleted);
                client.DownloadStringAsync(new
Uri(FacebookClient.Instance.GetAccessTokenRequestUrl(code)));
            }
}
```

The **code** parameter is needed to exchange as access token which is the real parameter that you need to do anything with the Facebook account via Facebook App. To exchange code for access token, you need to make POST request to server using **WebClient** as shown above. Once data is available, extract the **access_token** parameter and store it to persistent storage and then return back to MainPage.... You are almost there.

```
void AccessTokenDownloadCompleted(object sender, DownloadStringCompletedEventArgs e)
{
            string data = e.Result;
            data = "?" + data;

            // Acquire access_token and expires timestamp
            IEnumerable<KeyValuePair<string, string>> pairs =
UriToolKits.ParseQueryString(data);
            string accessToken = KeyValuePairUtils.GetValue(pairs, "access_token");
            string expires = KeyValuePairUtils.GetValue(pairs, "expires");

            // Save access_token
            FacebookClient.Instance.AccessToken = accessToken;

            // Back to MainPage
            var rootFrame = Application.Current.RootVisual as PhoneApplicationFrame;
            if (rootFrame != null)
              rootFrame.GoBack();
}
```

## Post to Wall

Actually **access_token** is the key of Facebook Connect. Once you got it, you could do anything with the API. For now, I will show you how to post to the wall with specific message. To post to the wall, just simply call FacebookClient.Instance.PostMessageOnWall

```
FacebookClient.Instance.PostMessageOnWall(TextToPost, new
UploadStringCompletedEventHandler(PostMessageOnWallCompleted));
```

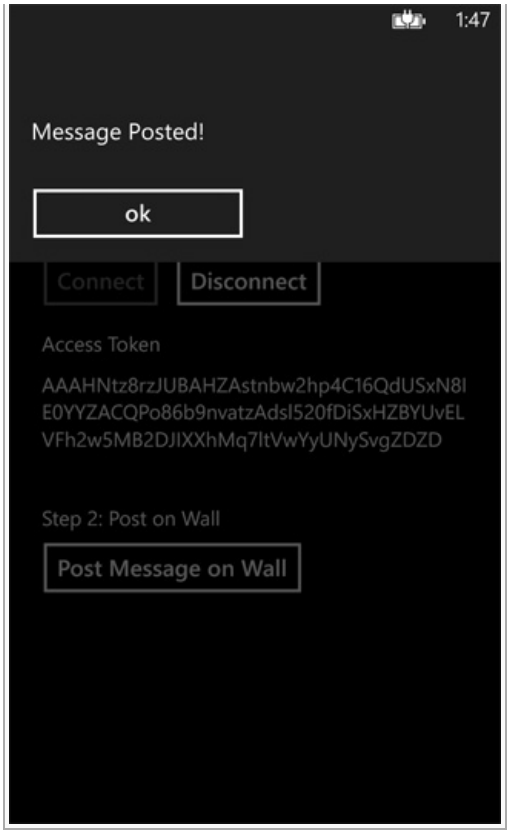
PostMessageOnWall is async request, you need to handle the result after all. Please note that access_token has expire date. In that case you need to exchange for new access token before doing thing.

```
void PostMessageOnWallCompleted(object sender, UploadStringCompletedEventArgs e)
{
            if (e.Cancelled)
                return;
            if (e.Error != null)
            {
                MessageBox.Show("Error Occurred: " + e.Error.Message);
                return;
```


(C) Copyright Nokia 2014. All rights reserved.

```
            }

            System.Diagnostics.Debug.WriteLine(e.Result);

            string result = e.Result;
            byte[] data = Encoding.UTF8.GetBytes(result);
            MemoryStream memStream = new MemoryStream(data);
            DataContractJsonSerializer serializer = new
DataContractJsonSerializer(typeof(ResponseData));
            ResponseData responseData = (ResponseData)serializer.ReadObject(memStream);

            if (responseData.id != null && !responseData.id.Equals(""))
            {
                // Success
                MessageBox.Show("Message Posted!");
            }
            else if (responseData.error != null && responseData.error.code == 190)
            {
                if (responseData.error.error_subcode == 463)
                {
                    // Access Token Expired, need to get new token
                    FacebookClient.Instance.ExchangeAccessToken(new
UploadStringCompletedEventHandler(ExchangeAccessTokenCompleted));
                }
                else
                {
                    // Another Error with Access Token, need to clear the Access Token
                    FacebookClient.Instance.AccessToken = "";
                    SetLoggedInState(false);
                }
            }
            else
            {
                // Error
            }
}

void ExchangeAccessTokenCompleted(object sender, UploadStringCompletedEventArgs e)
{
            // Acquire access_token and expires timestamp
            IEnumerable<KeyValuePair<string, string>> pairs =
UriToolKits.ParseQueryString(e.Result);
            string accessToken = KeyValuePairUtils.GetValue(pairs, "access_token");

            if (accessToken != null && !accessToken.Equals(""))
            {
                MessageBox.Show("Access Token Exchange Failed");
                return;
            }

            // Save access_token
            FacebookClient.Instance.AccessToken = accessToken;
            FacebookClient.Instance.PostMessageOnWall(TextToPost, new
UploadStringCompletedEventHandler(PostMessageOnWallCompleted));
}
```


(C) Copyright Nokia 2014. All rights reserved.




And this is what you will get on Facebook wall !

Congratulations. Your application is now connected to Facebook ! To make it more functional, you could modify the **FacebookClient.cs** to add more function for example upload the photo. For the API reference, you could find it from https://developers.facebook.com/docs/reference/api/

## Code Snippets

You can download the code for this example from File:FacebookConnect.zip


(C) Copyright Nokia 2014. All rights reserved.