

Azure Week

HOL – Universal Windows Platform Apps and Azure App Services



Contents

[Overview 4](#_Toc434234559)

[Lab 1: Authenticating Facebook in your UWP app using mobile services 5](#_Toc434234560)

[Lab 2: Push Notification using Azure Notification Hub in your UWP app 14](#_Toc434234561)

Windows Developer account creation……………………………………………………………………………..25

# Overview

This hands-on lab builds on the content of the breakout session and will provide depth information to the participants about using Azure services while developing Universal Windows Apps on Windows using VS2015.

Below are the topics covered in the session.

1. Push Notification using Azure Notification Hub in your UWP app
2. Authenticating Facebook in your UWP app using mobile services.

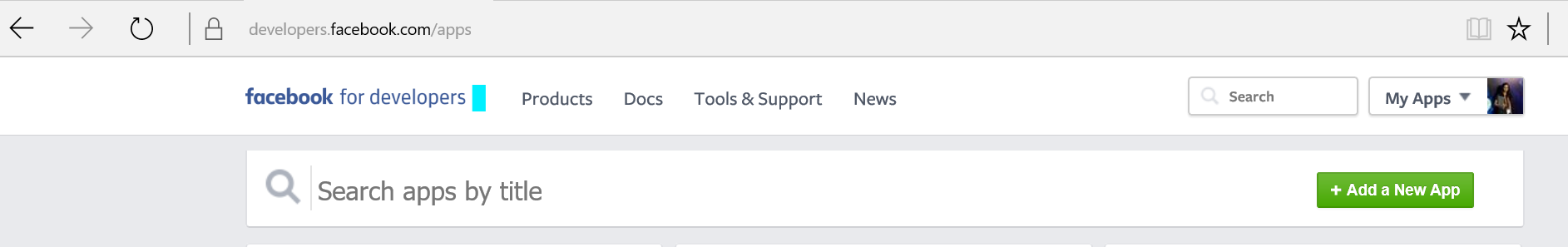
This lab will help participants develop a greater understanding of the capabilities of UWP Apps on Windows 10 and how Azure services can be leveraged.

# Lab 1: Adding Facebook Authentication in your UWP app using Azure App services

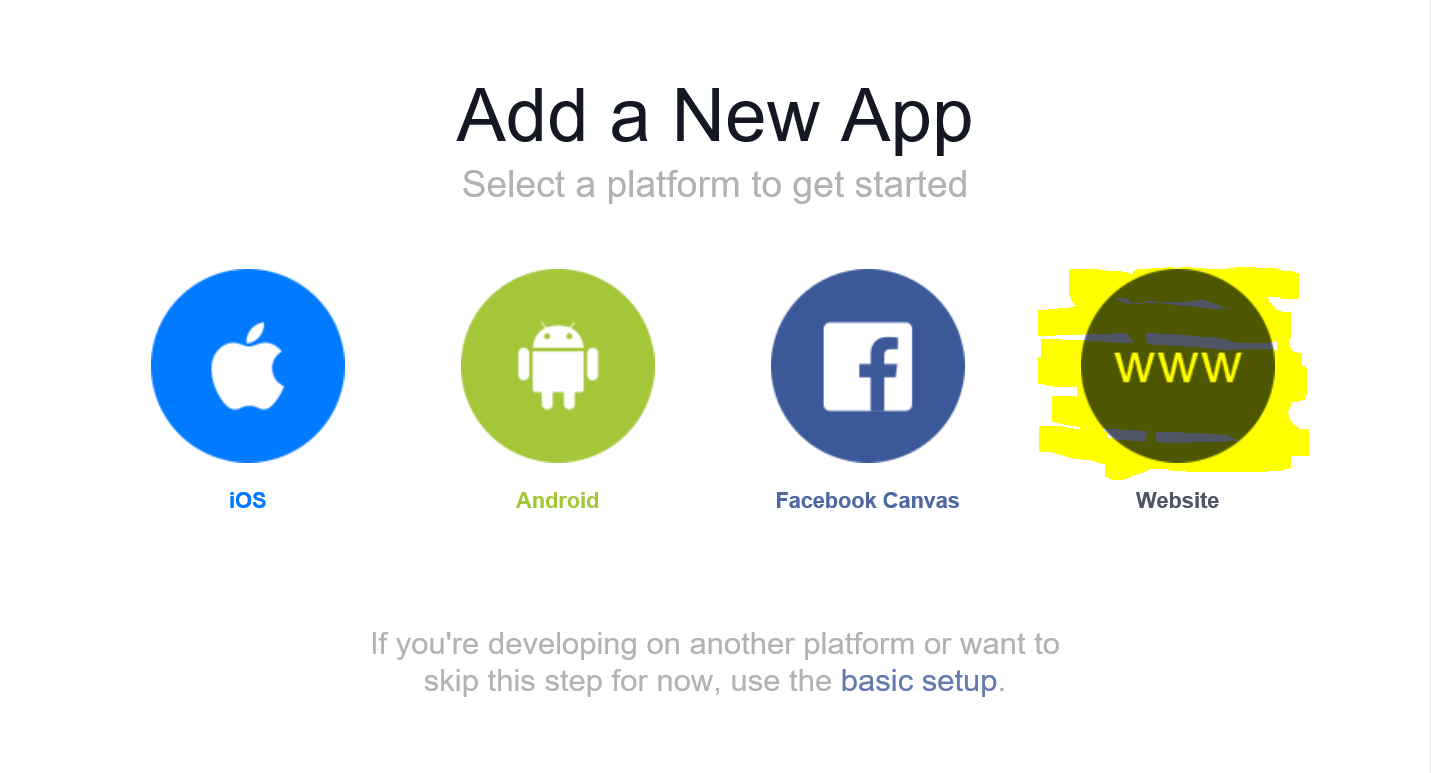
**Completion Time : ~20 minutes**

Lets create an App on facebook..

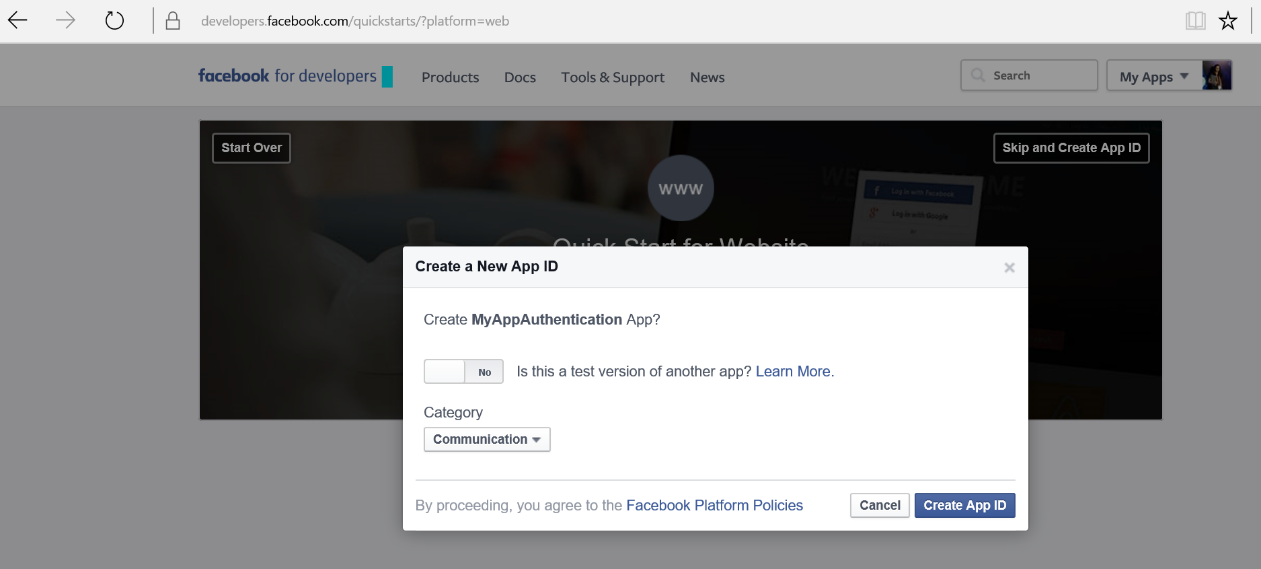
1. Go to <https://developers.facebook.com/apps/> and Login using your facebook account credentials.
2. Click on **Add a New App**



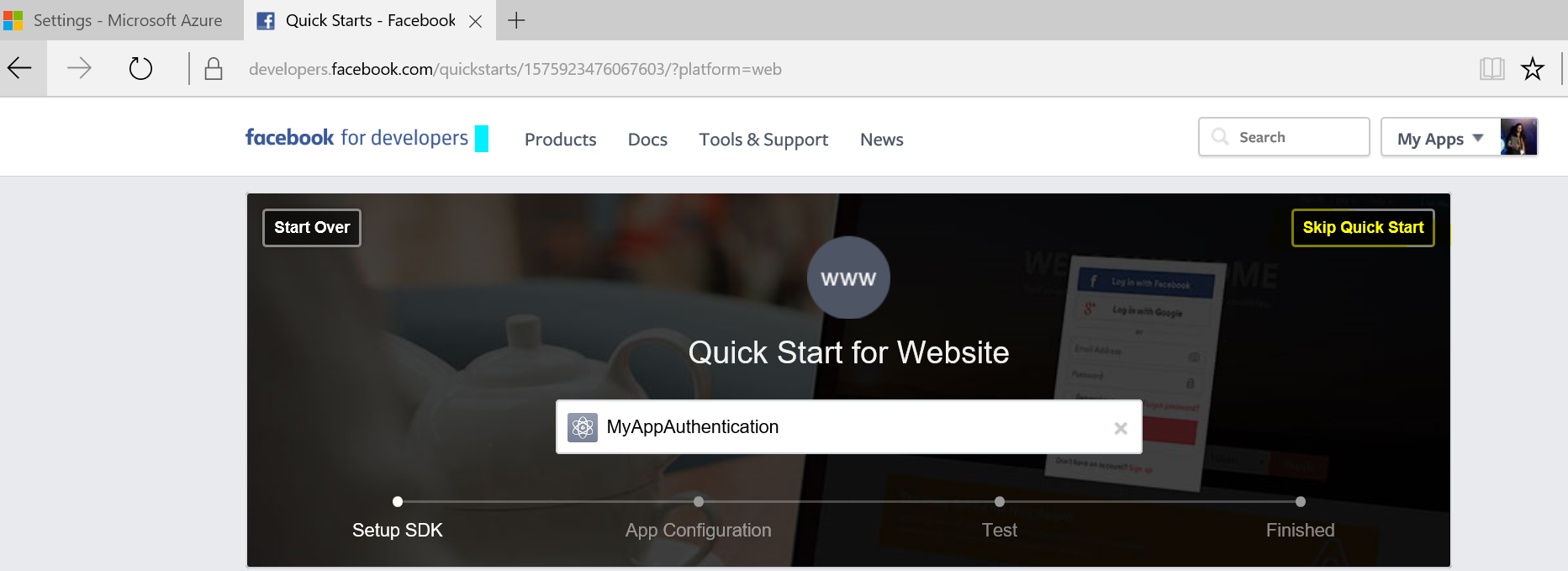
1. Choose **Websites**

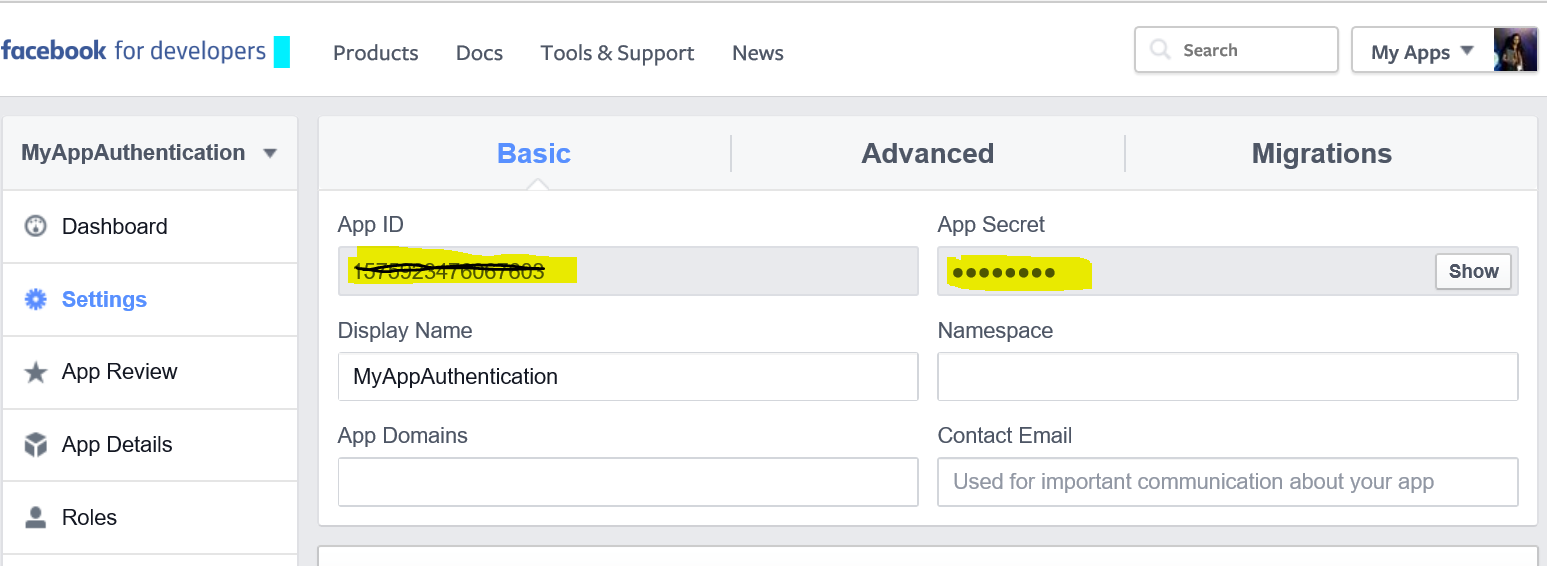


1. Choose a **Name** for your app > **Create new Favebook App ID** > Choose a **category** > Click on **Create App ID**



1. Click on **Skip Quick Start**

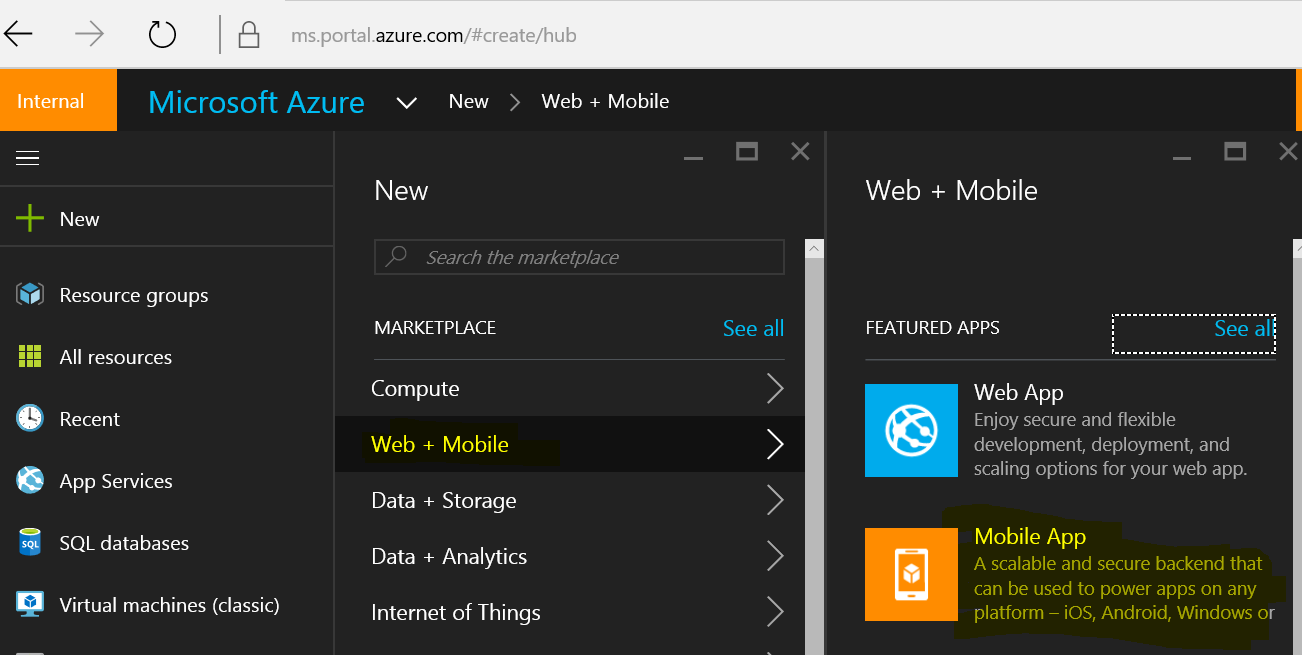


1. Copy your **AppID** and **Secret** in a notepad

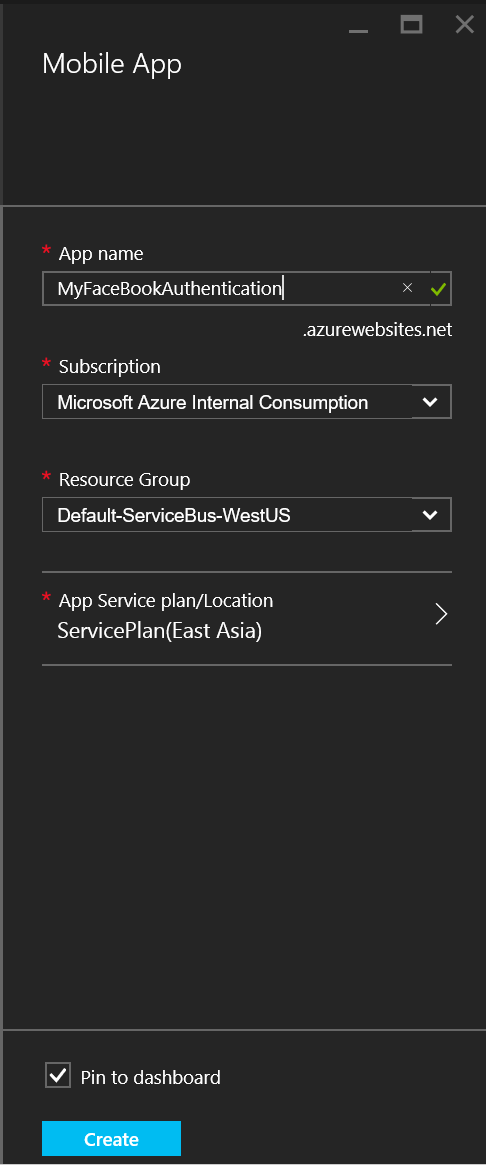
You have an app created on facebook now. Hold on a bit, we’ll get back here again!

Here comes the Azure part. . .

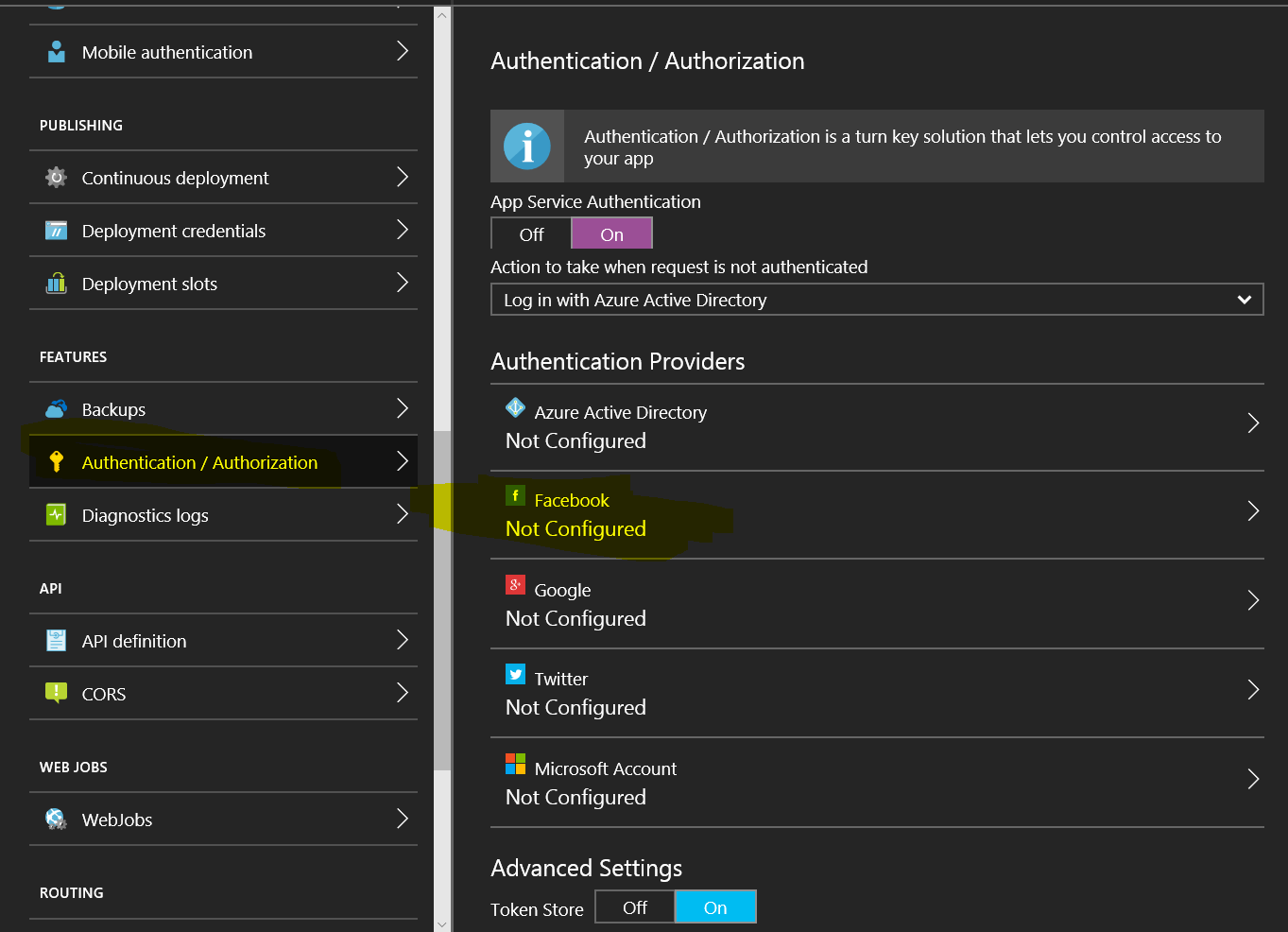
1. Go to <https://ms.portal.azure.com/>
2. Choose **NEW** > **Web +Mobile** > **Mobile App**



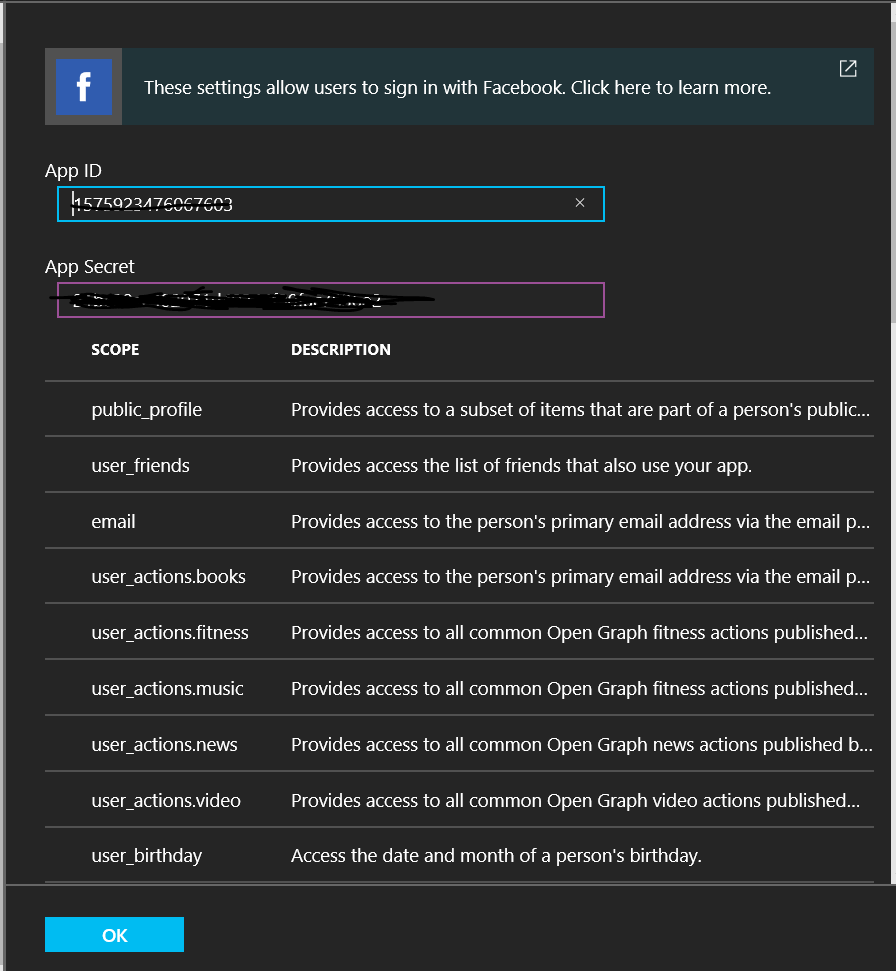
1. Give a unique name for your app and click Create (You can leave the rest of the settings to default or if it is null, choose the appropriate options).



1. Go to **Authentication / Authorization** > Click **ON** for App Service Authentication > **Facebook Configure**



1. Enter your App ID and App Secret which you have copied from Facebook App settings > **OK**.

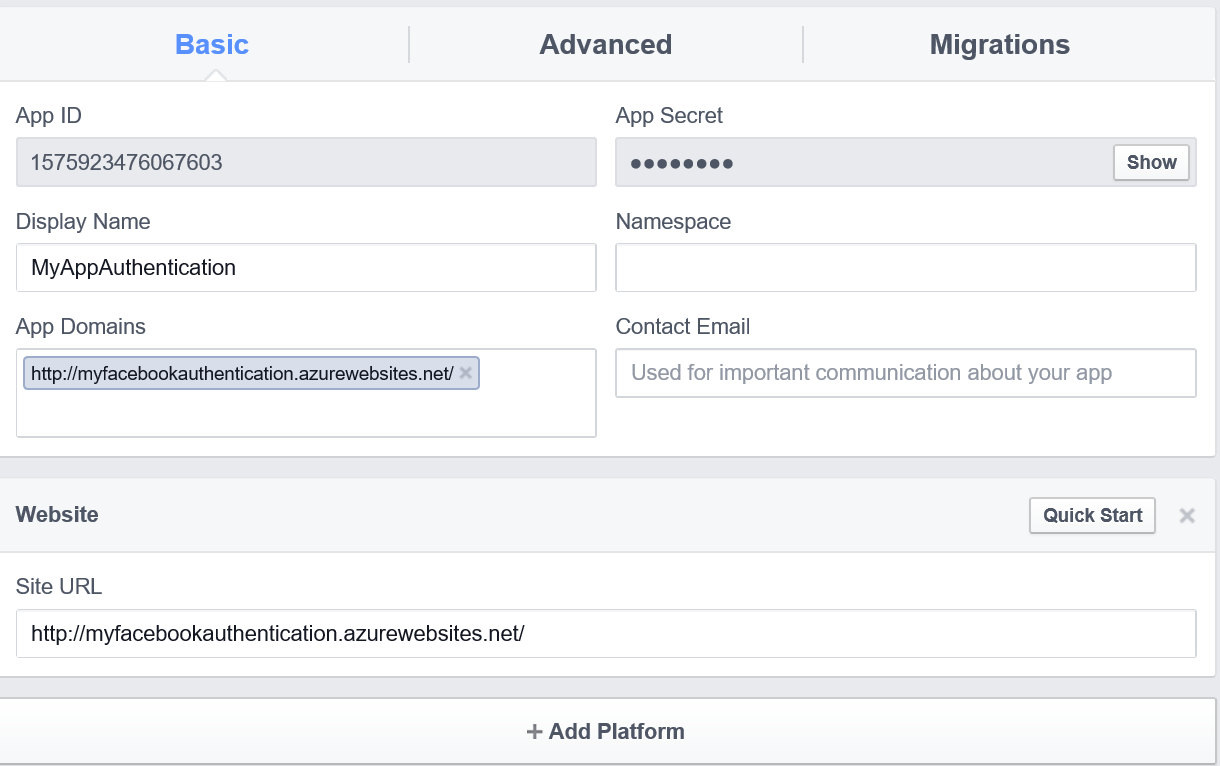


You should see **Configured** for Facebook now.

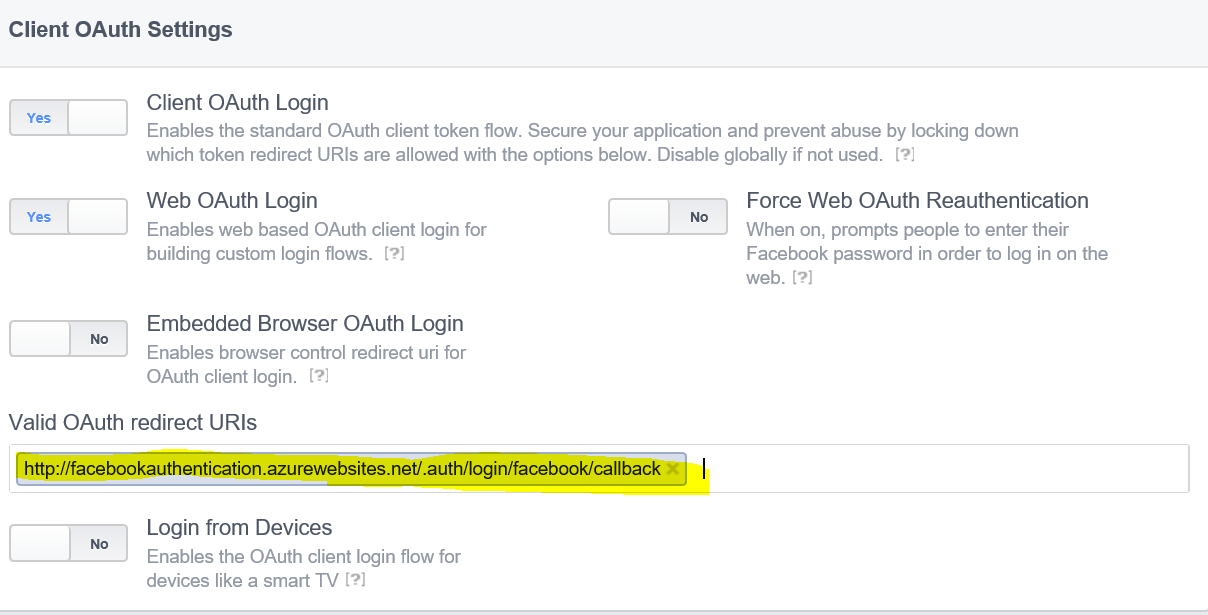


1. Go back to your <https://developers.facebook.com/apps/> and choose your app which you created.
2. Give the **name** of the Mobile app which you just created under App Domain name.

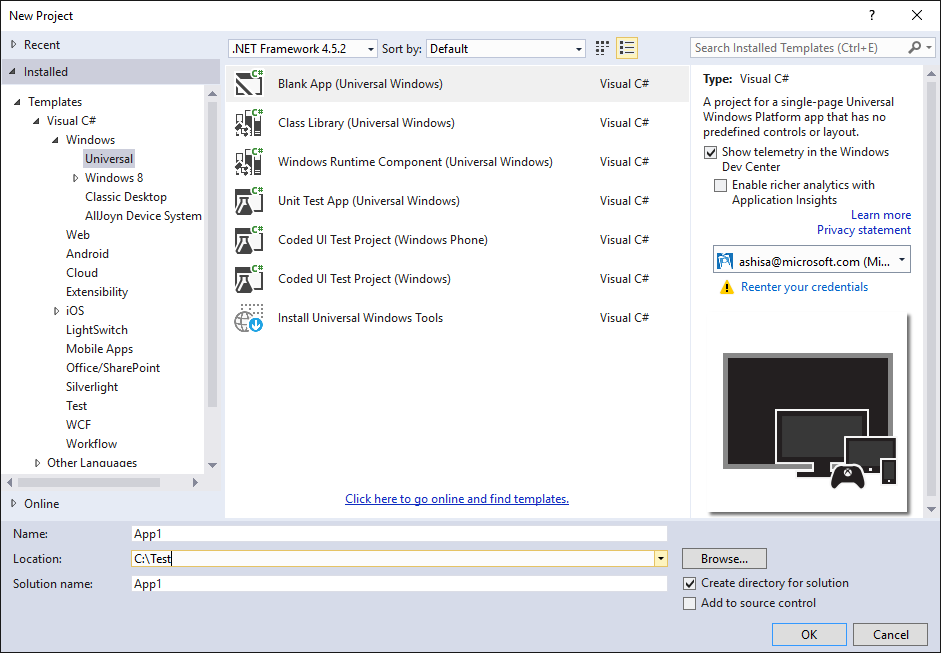
Add a **Platform** and choose **Website. Save changes.**

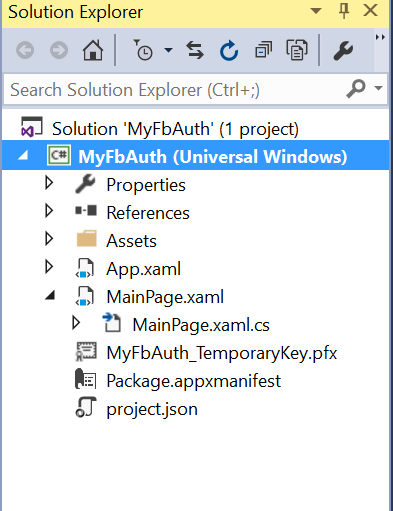


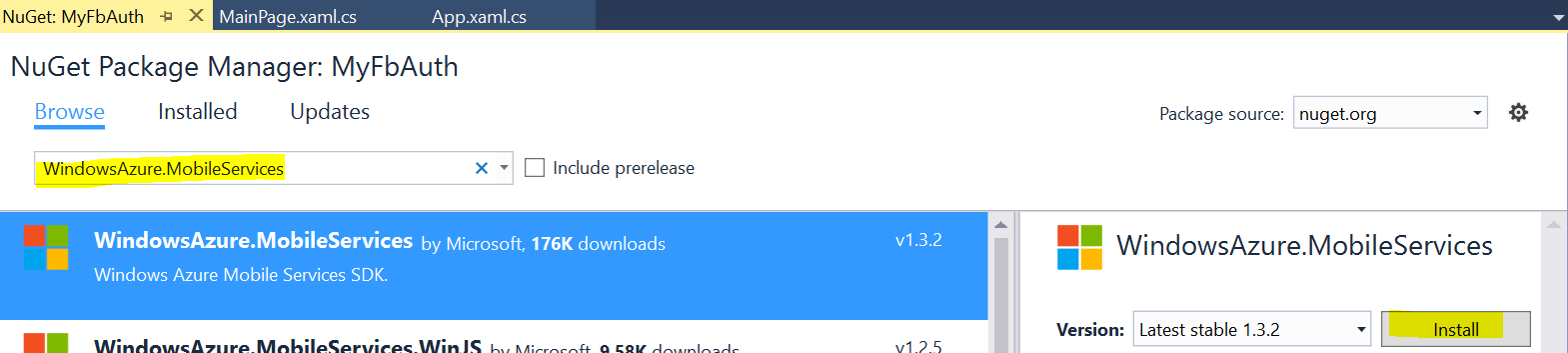
1. Go to **Advanced** tab **> Valid OAuth redirect URIs > Add your Mobile App URL append with *.auth/login/facebook/callback***



Now we have Azure and Facebook part ready.. Lets start our UWP app.

1. Start **Visual Studio 2015 > File > New > Project > Visual C# Universal (template)**
2. Create a blank app and give it a Name-  
   
3. Right Click the project from Solution Explorer> **Manage Nuget Packages** > Browse – “**WindowsAzure.MobileServices”** and Install it.





1. Go to MainPage.xaml.cs and paste the below code inside your MainPage **class** :

private MobileServiceUser user;

public static MobileServiceClient MobileService =

new MobileServiceClient("https://<Your Azure website URL>.azurewebsites.net");

private async System.Threading.Tasks.Task<bool> AuthenticateAsync()

{

string message;

bool success = false;

try

{

user = await MobileService.LoginAsync(MobileServiceAuthenticationProvider.Facebook);

message =

string.Format("You are now signed in - {0}", user.UserId);

success = true;

}

catch (InvalidOperationException)

{

message = "You must log in. Login Required";

}

var dialog = new MessageDialog(message);

dialog.Commands.Add(new UICommand("OK"));

await dialog.ShowAsync();

return success;

}

1. Add th below NameSpaces : -

using Microsoft.WindowsAzure.MobileServices;

using Windows.UI.Popups;

1. Go to MainPage.xaml and insert the below code inside the grid

<Button Name="ButtonLogin" Click="ButtonLogin\_Click"

Visibility="Visible" HorizontalAlignment="Center" VerticalAlignment="Center">Sign in</Button>

1. Add the below code for ButtonClick inside MainPage.xaml.cs

private async void ButtonLogin\_Click(object sender, RoutedEventArgs e)

{

if (await AuthenticateAsync())

{

ButtonLogin.Visibility = Windows.UI.Xaml.Visibility.Collapsed;

}

}

1. Run the app (Press F5) > Click on Sign In – You’ll see that Facebook Login appears in the app.



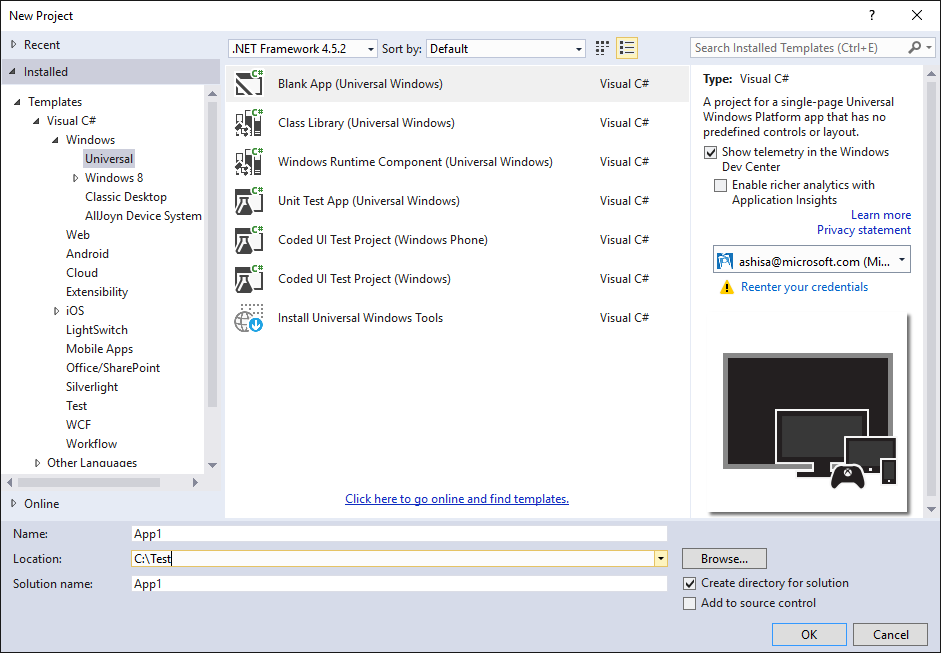
\*\* Try on further experiments to check how you can store the logged in user and get details like – Users posts / Status / Friends etc in your app! \*\*

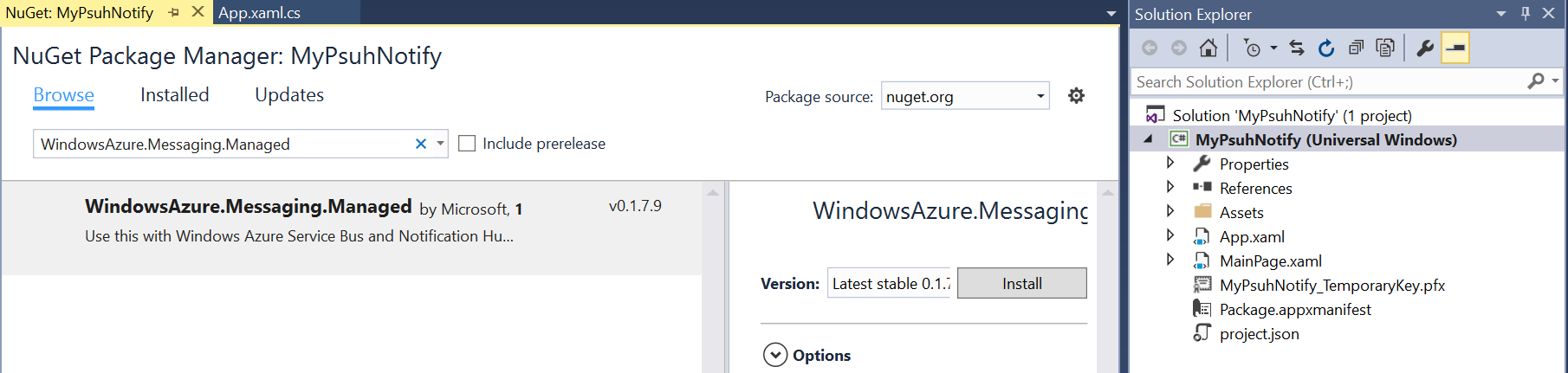
Reference Links: <https://azure.microsoft.com/en-us/documentation/articles/mobile-services-how-to-register-facebook-authentication/>

# Lab 2: Push Notification using Azure Notification Hub in your UWP app

**Completion Time : ~30 minutes**

**Pre-Requisite**: You need to have a [**Windows App Publisher Account**](#_Windows_Developer_Account). (refer Section 3 in this document)

1. Start **Visual Studio 2015 > File > New > Project > Visual C# Universal (template)**
2. Create a blank app and give it a Name-  
   
3. Right Click the project from Solution Explorer> **Manage Nuget Packages** > Browse – “**WindowsAzure.Messaging.Managed”** and Install it.



1. Go to MainPage.Xaml.cs and paste the below code. Lets replace the connecting string and Hub name in a while.

private async void InitNotificationsAsync()

{

var channel = await PushNotificationChannelManager.CreatePushNotificationChannelForApplicationAsync();

var hub = new NotificationHub("<HubName>", "<Your Connection String>");

var result = await hub.RegisterNativeAsync(channel.Uri);

// Displays the registration ID so you know it was successful

if (result.RegistrationId != null)

{

var dialog = new MessageDialog("Registration successful: " + result.RegistrationId);

dialog.Commands.Add(new UICommand("OK"));

await dialog.ShowAsync();

}

}

1. Add the below name spaces:-

using Microsoft.WindowsAzure.Messaging;

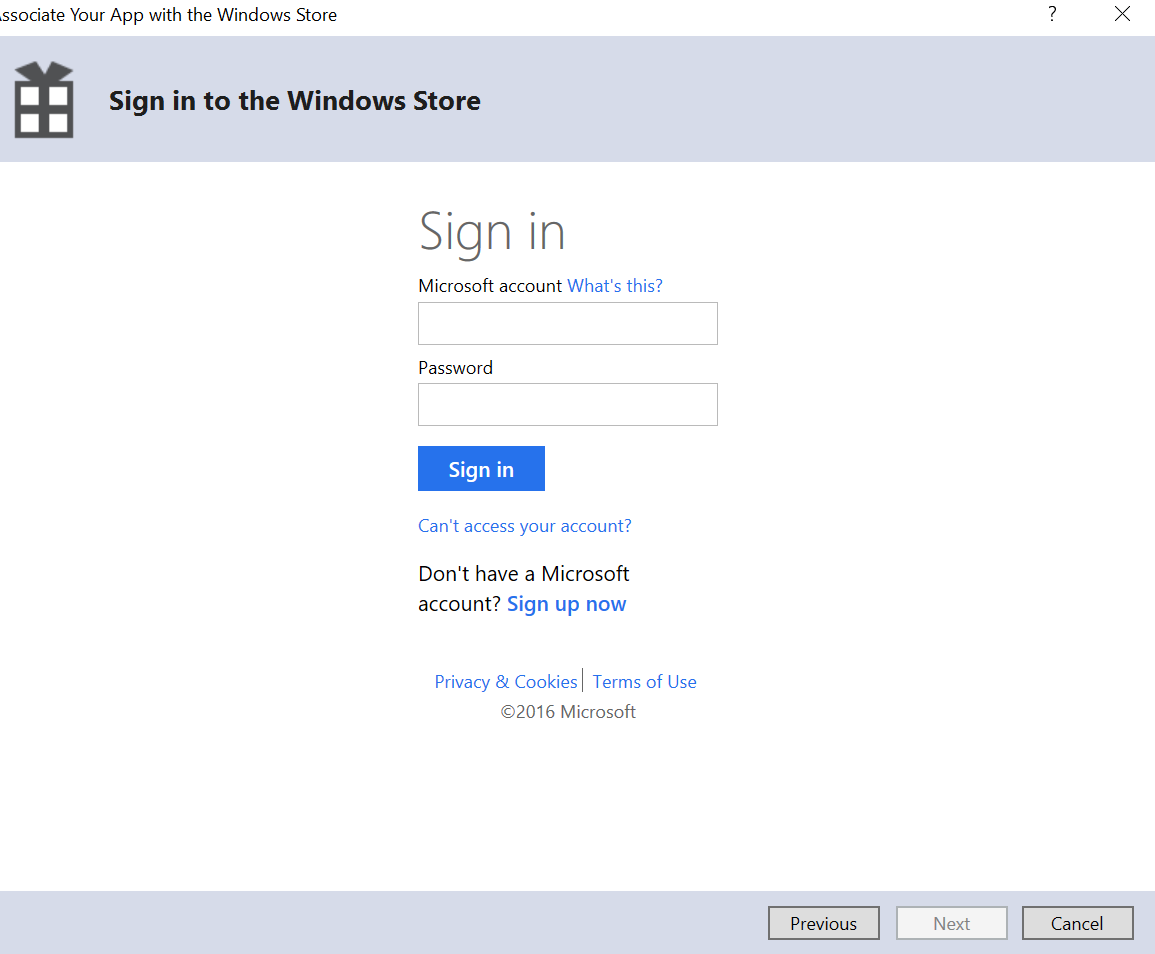
using Windows.Networking.PushNotifications;

using Windows.UI.Popups;

1. Lets associate the app with the store.

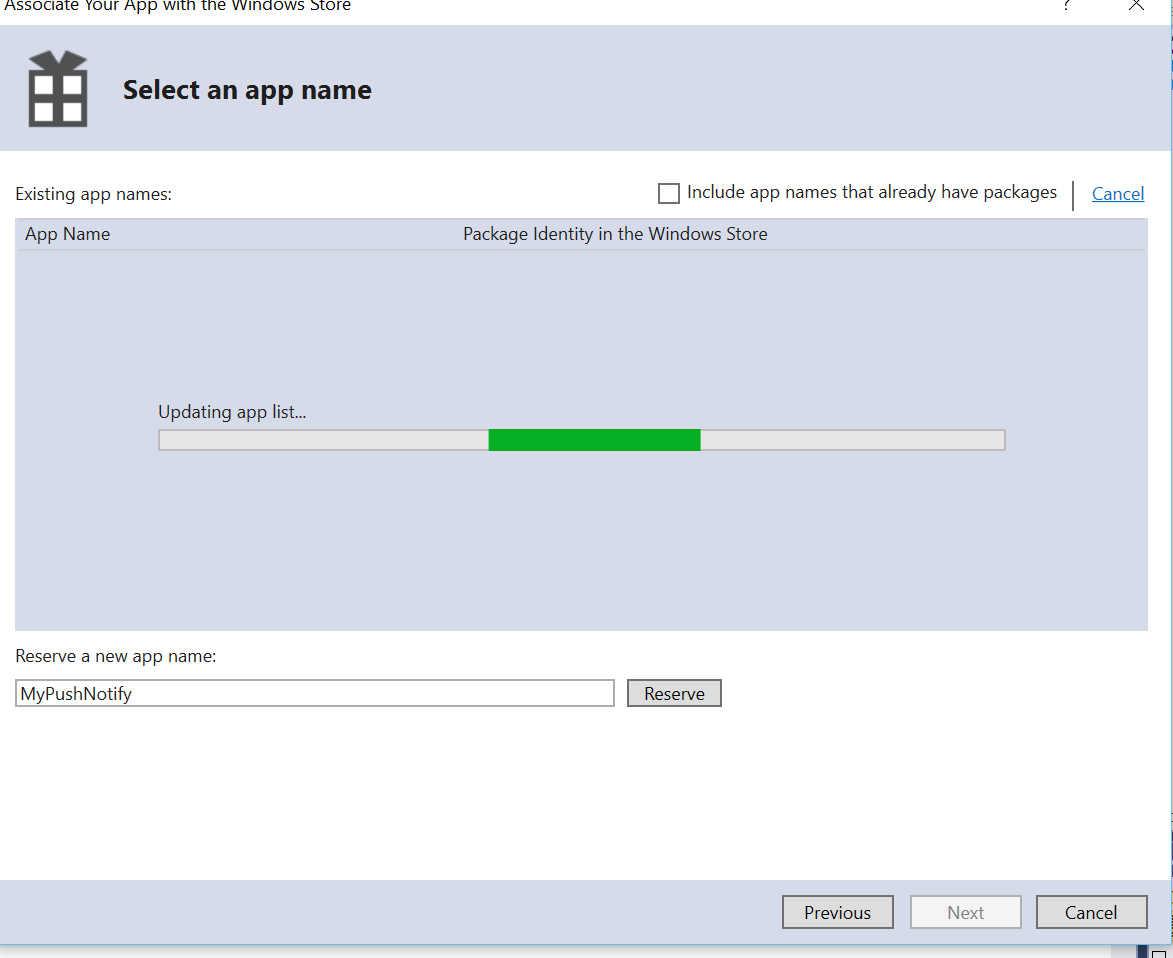
Right Click Project > **Store** > **Associate App with the Store**

1. Login to your Windows Store Account

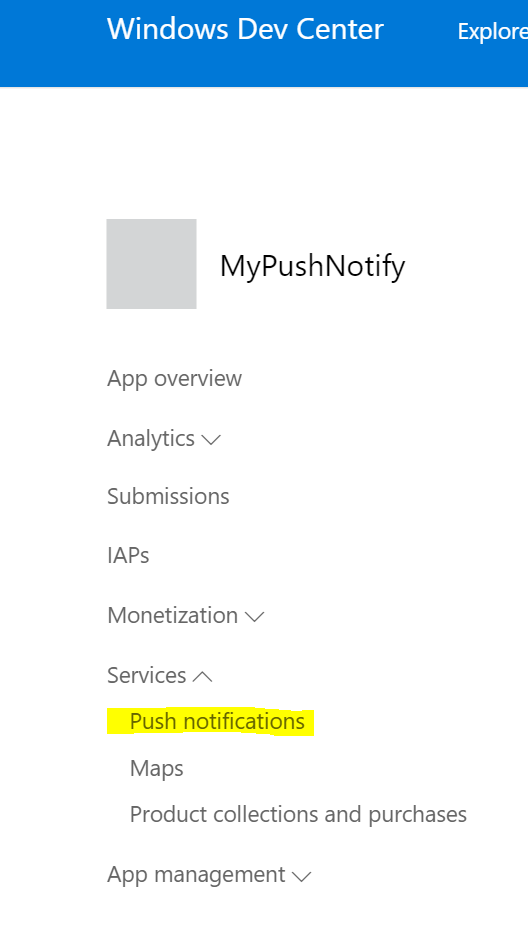


1. Reserve a **new App Name** for your Notification App:

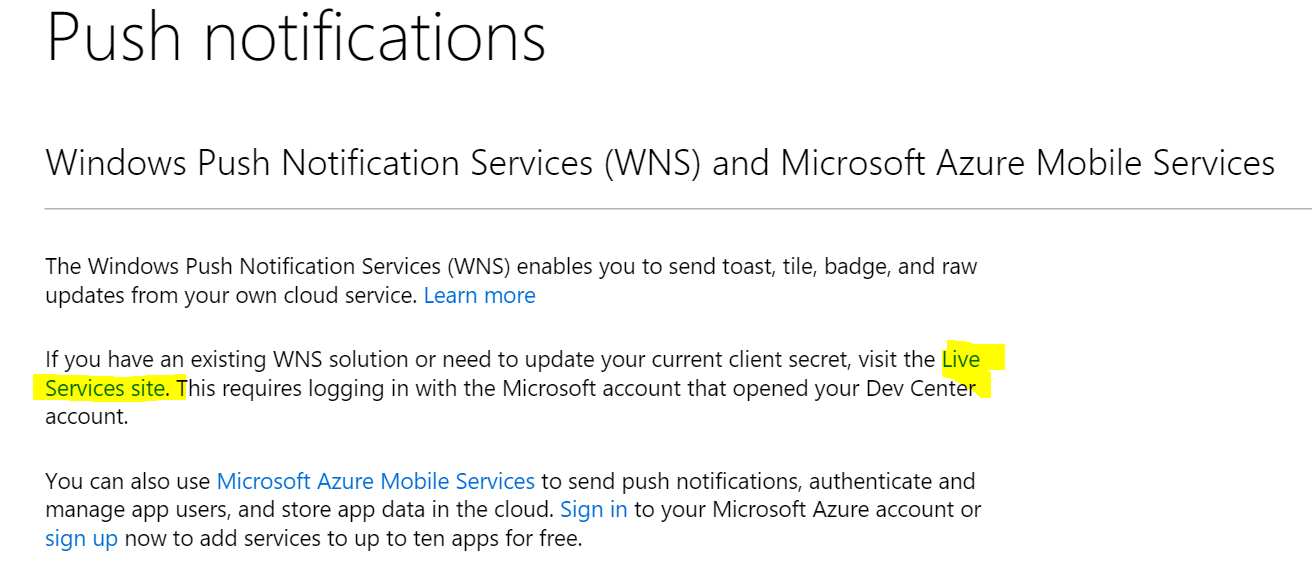
Choose it and **Associate**



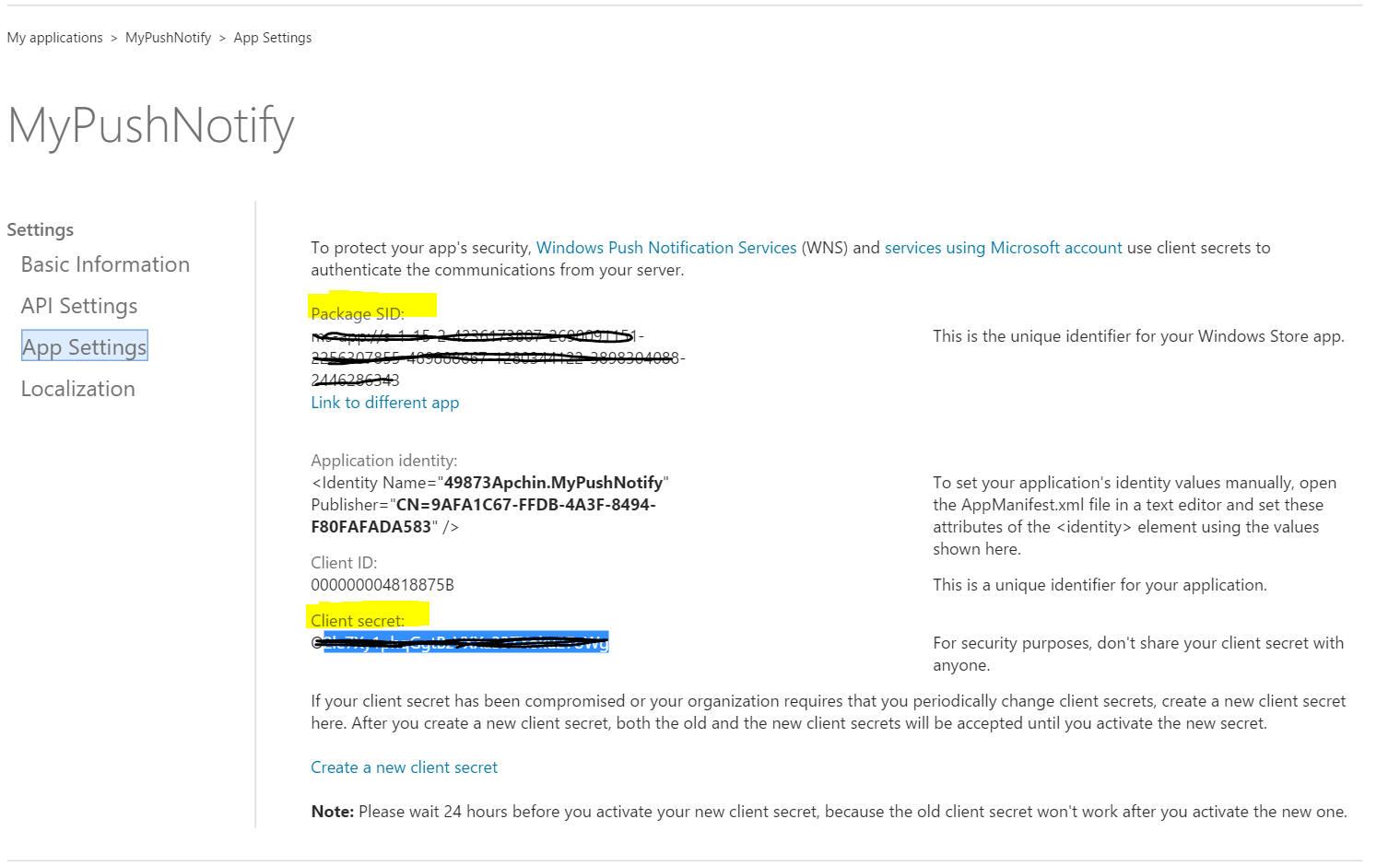
1. Go to <https://dev.windows.com/en-us> > Sign in > **Dashboard** > Go to your App Namne which you just **reserved** from Visual Studio > **Services** > **Push Notification**



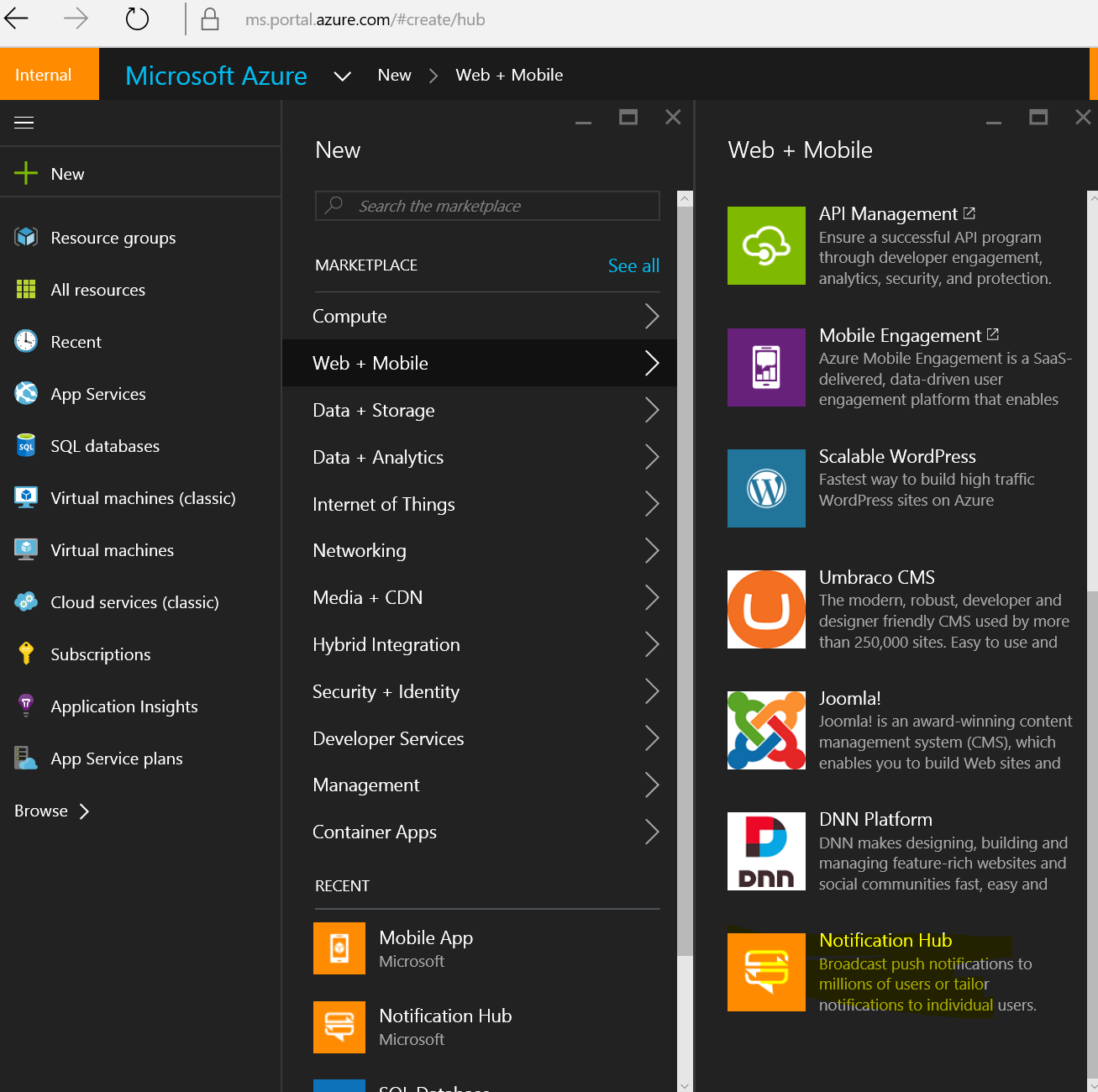
1. Go to **Live Services Site**.



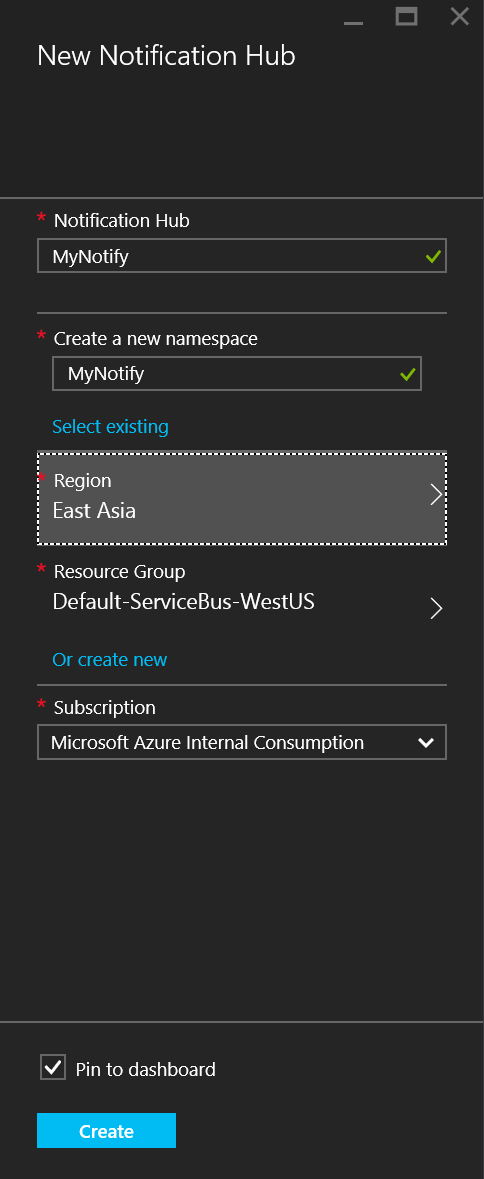
1. Go to App Settings and Note down the **Package SID and Client Secret**



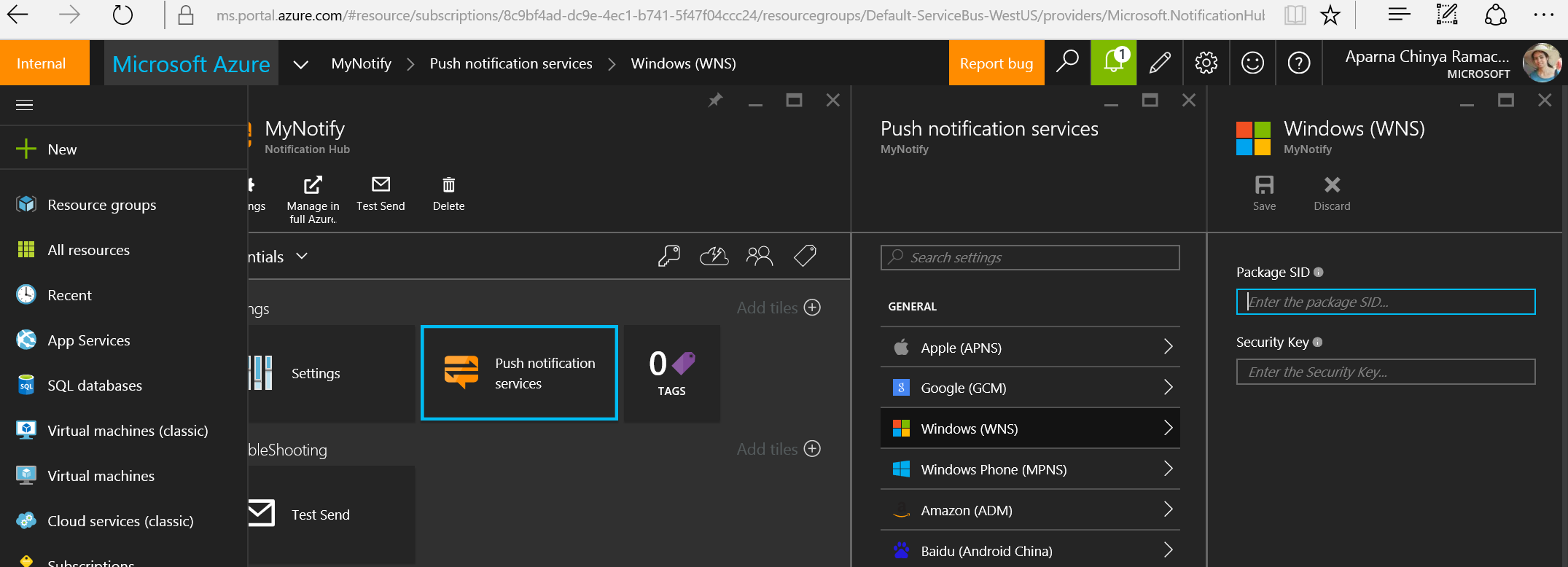
1. Go to <https://ms.portal.azure.com/>
2. **New** > **Web +Mobile** > **Notification Hub**



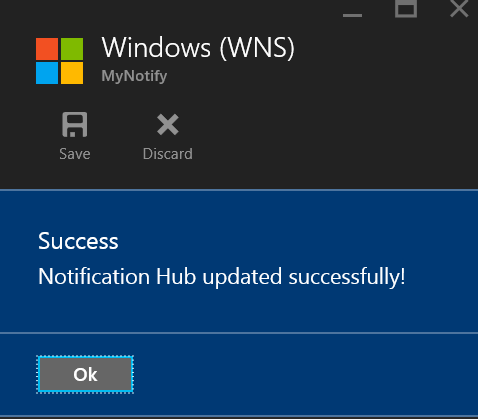
1. Give a unique Name and Namespace for your Notification Hub and click Create.



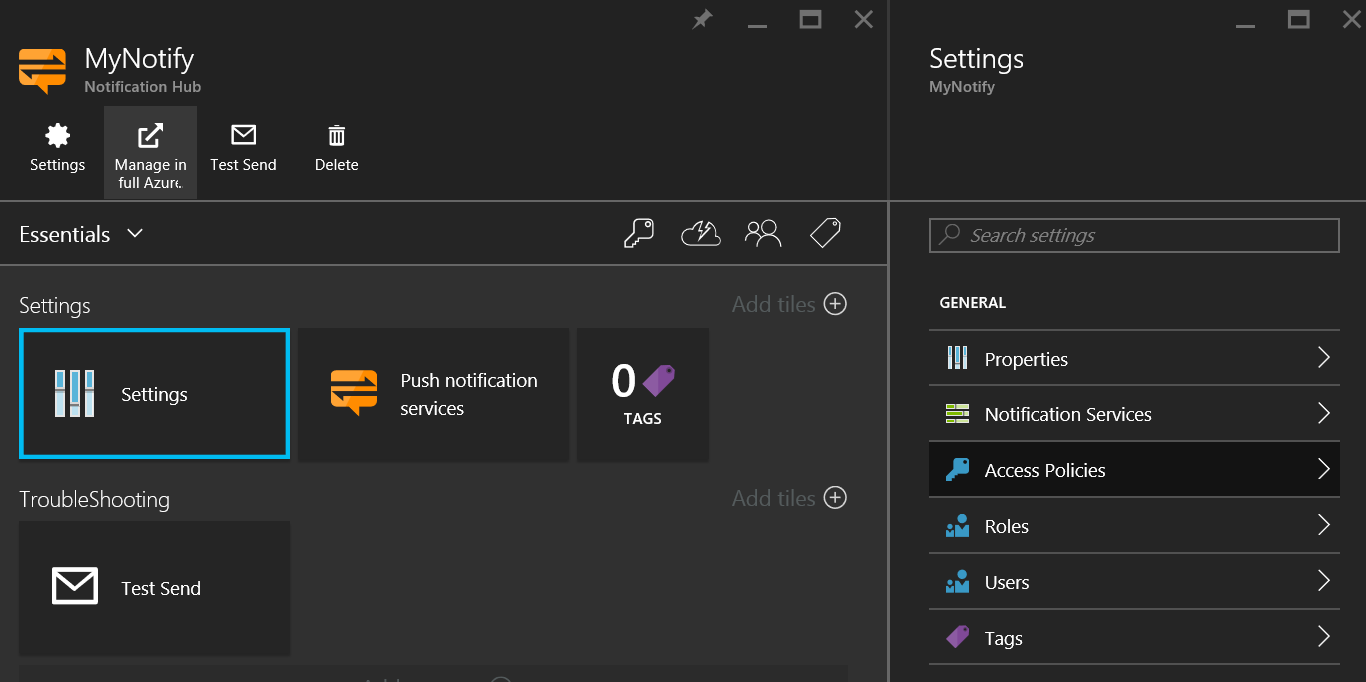
1. Once deployed go to Push Notification Services > Windows WNS.
2. Enter the **Package SID and Security Key** obtained from associating your UWP to the Windows Store.



1. Notification Hub Should be successafully created!

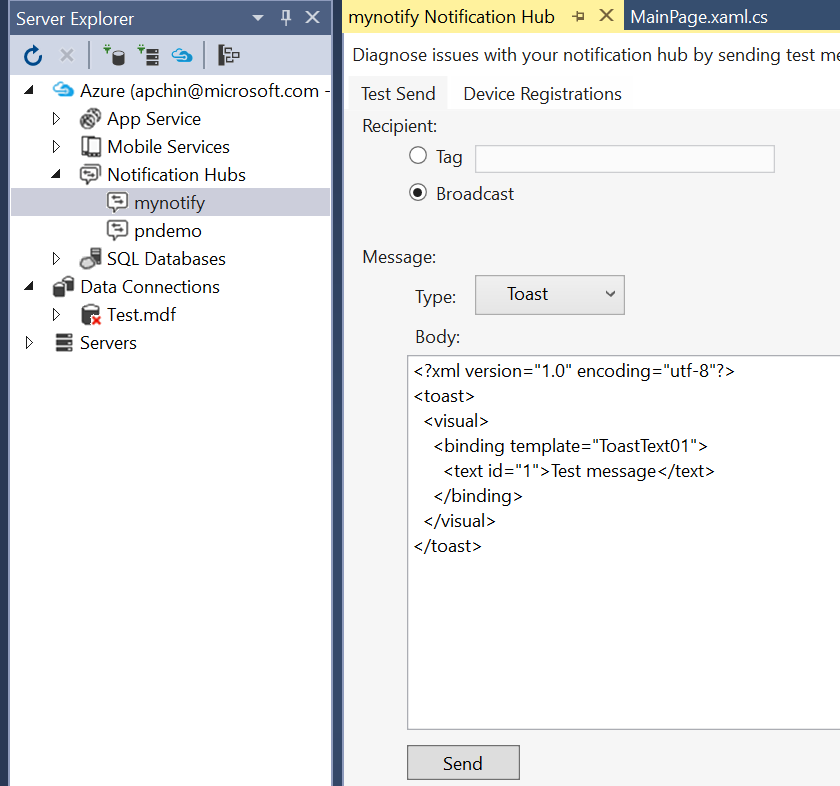


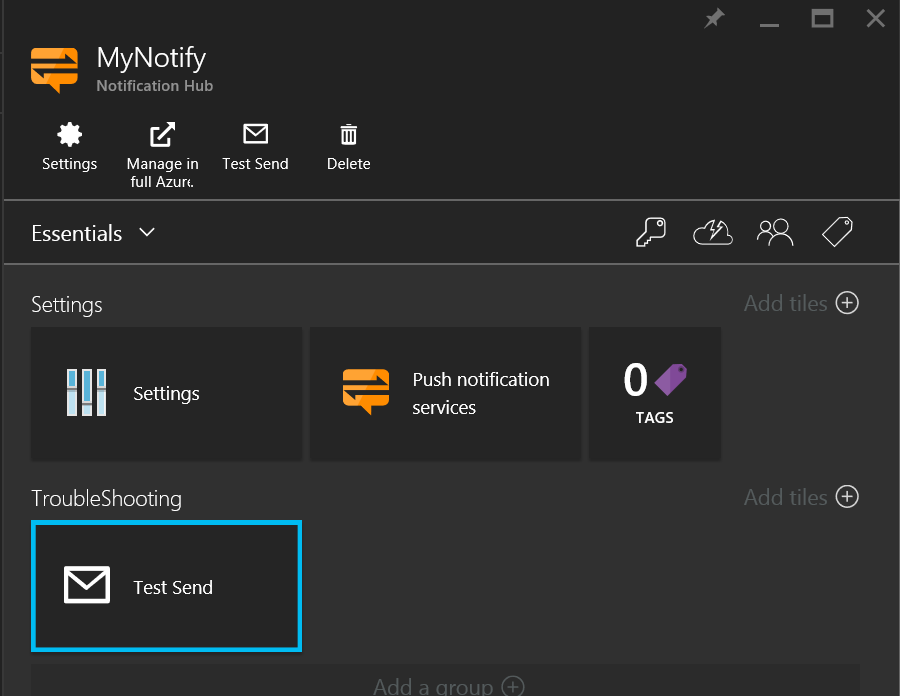
1. Go to the Settings of your Notification Hub > Access Policies > Copy value of **connection string** of **DefaultFullSharedAccessSignature.**

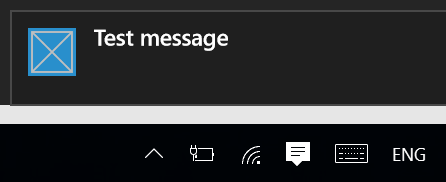


1. Go to your UWP app and replace the Notification Hub Name and connection String in the line : - var hub = new NotificationHub("<HubName>", "<Your Connection String>");
2. Call the InitNotificationsAsync() function in the constructor.
3. Run your UWP app. You should see a successful registeration Message.
4. Test your Notifications.

Go to Server Explorer > Azure > Notification Hubs > Your Notification Hub > **Test Send** > Type: **Toast** > **Send**



1. You can also send test notifications from the Azure Portal 
2. You should see a toast notification appearing on all the devices where your UWP was installed.



**NOTE**: You can also test sending a push notification from a console app.

Create a **Console application** from Visual studio and use the below function to send push notifications on all the registered devices.

private static async void SendNotificationAsync()

{

NotificationHubClient hub = NotificationHubClient.CreateClientFromConnectionString(

"<Connection String>", "<HubName>");

var toast = @"<toast><visual><binding template=""ToastText01""><text id=""1"">Hello from a .NET App!</text></binding></visual></toast>";

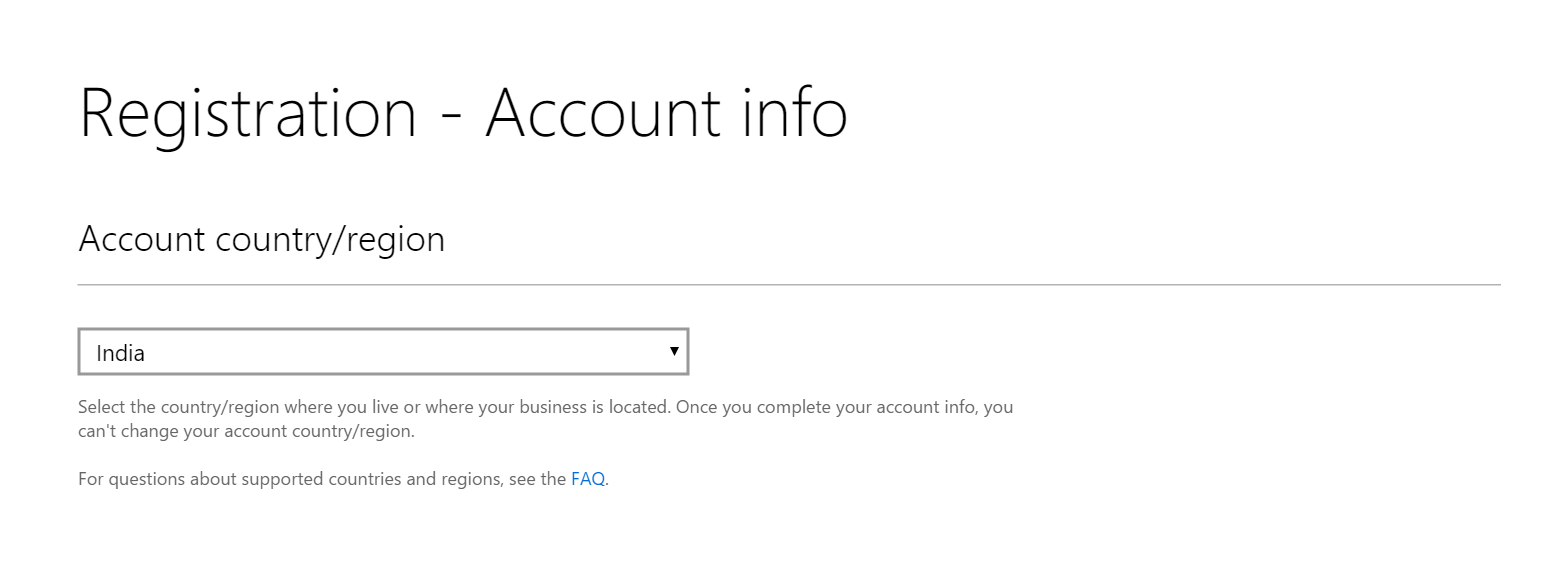
await hub.SendWindowsNativeNotificationAsync(toast);

}

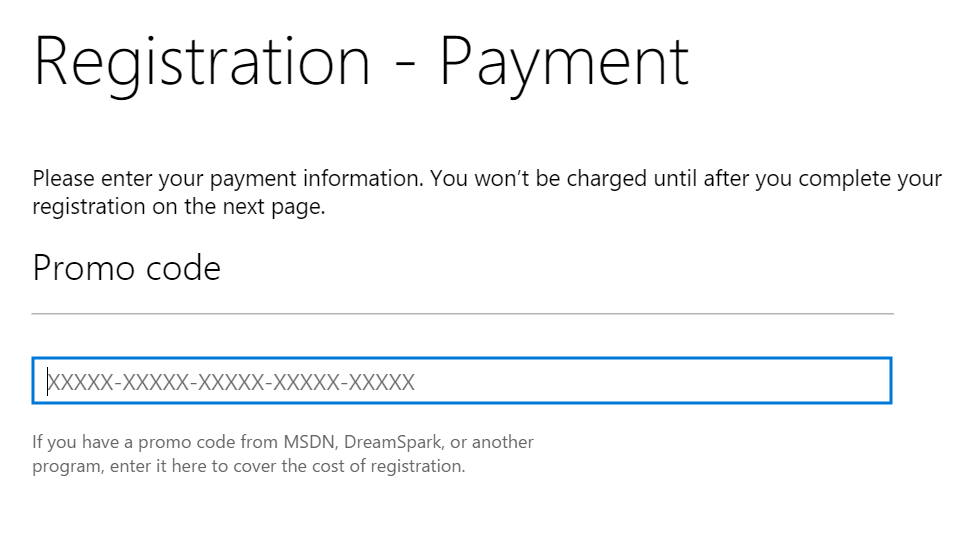
# Section 3 - Windows Developer Account

Go to : <https://dev.windows.com/en-us/programs/join> and register as an **App Developer**.

\*\*You can create a free Publisher account using the **Coupon** from your **Microsoft Subscriptions.**



Enter PromoCode which you can get from your Microsoft Subscriptions.



**Your Developer account is successfully created.**