

Sopra Steria, Microsoft Meetup Community

**Push Notifications**

Azure Mobile Apps

Enabling push notifications for your app

# Configure a Notification Hub

The Mobile Apps feature of Azure App Service uses Azure Notification Hubs to send pushes, so you will be configuring a notification hub for your mobile app.

1. In the Azure portal, go to **App Services**, and then select your app back end. Under **Settings**, select **Push**.
2. To add a notification hub resource to the app, select **Connect**. You can either create a hub or connect to an existing one.
3. Now you have connected a notification hub to your Mobile Apps back-end project. Later you configure this notification hub to connect to a platform notification system (PNS) to push to devices.

# Register your app for push notifications

You need to submit your app to the Microsoft Store, then configure your server project to integrate with Windows Notification Services (WNS) to send push.

1. In Visual Studio Solution Explorer, right-click the UWP app project, click Store > Associate App with the Store&hellip.
2. In the wizard, click **Next**, sign in with your Microsoft account, type a name for your app in **Reserve a new app name**, then click **Reserve**.
3. After the app registration is successfully created, select the new app name, click **Next**, and then click **Associate**. This adds the required Microsoft Store registration information to the application manifest.
4. Navigate to the **Windows Dev Center**, sign-in with your Microsoft account, click the new app registration in **My apps**, then expand Services > Push notifications.
5. In the Push notifications page, click **Live Services site** under **Microsoft Azure Mobile Services.**
6. In the registration page, make a note of the value under **Application secrets** and the **Package SID**, which you will next use to configure your mobile app backend.

**Important**: The client secret and package SID are important security credentials. Do not share these values with anyone or distribute them with your app. The Application Id is used with the secret to configure Microsoft Account authentication.

# Configure the backend to send push notifications

1. In the Azure portal, select Browse All > App Services. Then select your Mobile Apps back end. Under Settings, select App Service Push. Then select your notification hub name.

2. Go to Windows (WNS). Then enter the Security key (client secret) and Package SID that you obtained from the Live Services site. Next, select Save

3. Your back end is now configured to use WNS to send push notifications.

# Update the server to send push notifications

**Node.js backend projects**

1. If you haven't already done so, download the quickstart project or else use the online editor in the Azure portal.

2. Replace the existing code in the todoitem.js file with the following:

var azureMobileApps = require('azure-mobile-apps'),   
promises = require('azure-mobile-apps/src/utilities/promises'),   
logger = require('azure-mobile-apps/src/logger');

var table = azureMobileApps.table();

table.insert(function (context) {   
// For more information about the Notification Hubs JavaScript SDK,   
// see http://aka.ms/nodejshubs   
logger.info('Running TodoItem.insert');

// Define the WNS payload that contains the new item Text.   
var payload = "<toast><visual><binding template=\ToastText01\><text id=\"1\">" + context.item.text + "</text></binding></ visual></toast>";

// Execute the insert. The insert returns the results as a Promise,   
// Do the push as a post-execute action within the promise flow.   
return context.execute()   
 .then(function (results) {   
 // Only do the push if configured   
 if (context.push) {   
 // Send a WNS native toast notification. context.push.wns.sendToast(null, payload, function (error) {   
if (error) {   
logger.error('Error while sending push notification: ', error); } else {   
logger.info('Push notification sent successfully!'); } });   
}   
// Don't forget to return the results from the context.execute()   
return results;   
})   
.catch(function (error) {   
logger.error('Error while running context.execute: ', error);   
});   
});

module.exports = table;

3. This sends a WNS toast notification that contains the item.text when a new todo item is inserted. 4. When editing the file on your local computer, republish the server project

# Add pus notifications to your app

Next, your app must register for push notifications on start-up. When you have already enabled authentication, make sure that the user signs-in before trying to register for push notifications.

1. Open the App.xaml.cs project file and add the following using statements:   
using System.Threading.Tasks;   
using Windows.Networking.PushNotifications;

2. In the same file, add the following InitNotificationsAsync method definition to the App class: private async Task InitNotificationsAsync() {

// Get a channel URI from WNS.   
 var channel = await PushNotificationChannelManager .CreatePushNotificationChannelForApplicationAsync();

// Register the channel URI with Notification Hubs  
await App.MobileService.GetPush().RegisterAsync(channel.Uri);

}

3. This code retrieves the ChannelURI for the app from WNS, and then registers that ChannelURI with your App Service Mobile App.

4. At the top of the OnLaunched event handler in App.xaml.cs, add the async modifier to the method definition and add the following call to the new InitNotificationsAsync method, as in the following example:

protected async override void OnLaunched(LaunchActivatedEventArgs e) {

await InitNotificationsAsync();

// ...   
}

5. This guarantees that the short-lived ChannelURI is registered each time the application is launched.

6. Rebuild your UWP app project. Your app is now ready to receive toast notifications.