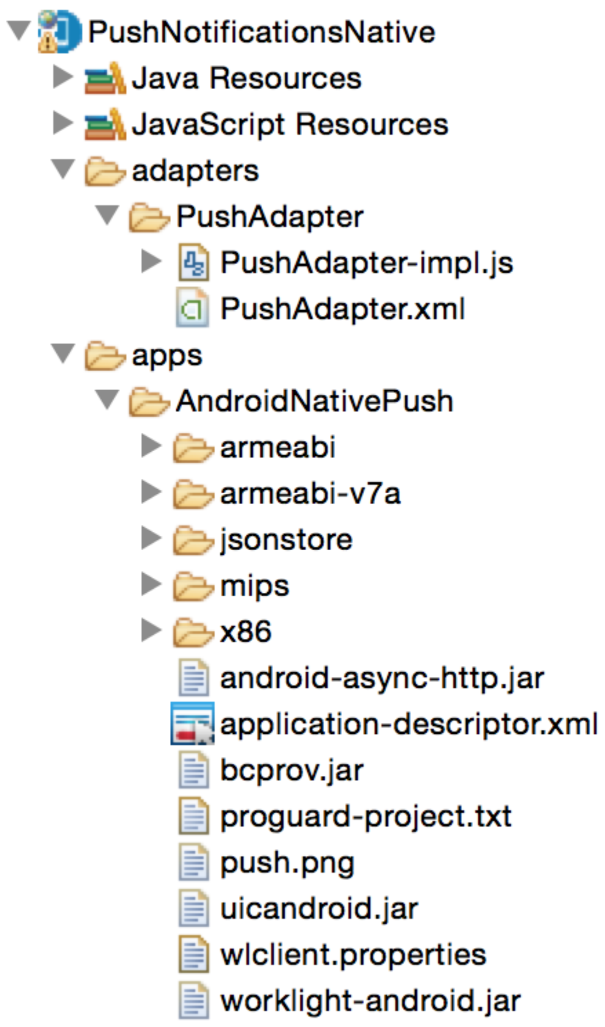
Push Notifications in Native Android Applications

Overview

This tutorial explains how to configure a MobileFirst Native Android application to support push notifications.
Also mentioned are the addresses and ports that are required for notifications to arrive to the supported Android Push Notification Service vendor (GCM).

Prerequisite: Make sure that you read the Configuring a native Android application with the MobileFirst Platform SDK (../../../configuring-the-mfpf-sdk/configuring-a-native-android-application-with-the-mfp-sdk/) tutorial first.

Setting up the project



(http://developer.ibm.com/mobilefirstplatform/wp-content/uploads/sites/32/2014/12/NativeAndroidPushAdapterExpanded-70.png)

1. **Create a MobileFirst project and add a MobileFirst Android Native API.**

   In this tutorial and the accompanying sample, the application is called "androidnativepush". Be sure to replace this value with your own application name.

   The native API includes the following push-related file:

   - The `push.png` file is an icon file that is displayed when a push notification arrives. Copy this file from your native API folder and put it in your native project's `res/drawable` folders.

2. **Edit the `application-descriptor.xml` file.**

   These settings are also editable with the Application Descriptor Editor in Design mode.

   - Add the `pushSender` child-element inside the `nativeAndroidApp` element . Replace the `key` and `senderId` values with your API key and project number respectively. In case you do not have these, you can get them from the Google Developer Console (https://console.developers.google.com).
     [code lang="xml" highlight="9"]
     <nativeAndroidApp id="AppName" platformVersion="7.1.0.00.20150812-0731" version="1.0" xmlns="http://www.worklight.com/native-android-descriptor">
     <displayName>AppName</displayName>
     <description>AppName</description>
     <publicSigningKey></publicSigningKey>
     <packageName></packageName>
     <accessTokenExpiration>3600</accessTokenExpiration>
     <userIdentityRealms></userIdentityRealms>
     <pushSender key="" senderId=""/>
     </nativeAndroidApp>
     [/code]

3. **Edit the `wlclient.properties` file.**

   Edit the `wlclient.properties` file in your native Android project and enter appropriate values for the following fields:

   - wlServerHost – The hostname or IP address of MobileFirst Server.
   - wlServerPort – The port on which MobileFirst Server is listening.
   - GcmSenderId – The project number that you obtained through the Google API console.

   [code lang="xml"]
   wlServerProtocol = http
   wlServerHost =
   wlServerPort = 10080
   wlServerContext = /MobileFirst-Project-Name/
   wlAppId = MobileFirst-App-Name
   wlAppVersion = 1.0

wlEnvironment = Androidnative
wlUid = wY/mbnwKTDDYQUvuQCdSgg==
wlPlatformVersion = 7.1.0.0
#languagePreferences = Add locales in order of preference (e.g. en, fr, fr-CA)
#For Push Notifications,uncomment below line and assign value to it
GcmSenderId =
[/code]

4. **Add Google Play Services.**

   For instructions about how to setup Google Play Services review the Setting Up Google Play Services (http://developer.android.com/google/play-services/setup.html) topic at the Android Developer website.

5. **Modify the native Android project.**

   Verify that the following permissions exist in the `AndroidManifest.xml` file of your Android project.

   [code lang="xml"]
   <permission android:name="com.example.project.permission.C2D_MESSAGE"
   android:protectionLevel="signature" />

   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission android:name="android.permission.GET_TASKS"/>
   <uses-permission android:name="com.example.project.permission.C2D_MESSAGE" />
   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
   <uses-permission android:name="android.permission.WAKE_LOCK" />
   <uses-permission android:name="android.permission.GET_ACCOUNTS" />
   <uses-permission android:name="android.permission.USE_CREDENTIALS" />
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>[/code]

   Replace all **com.example.project** with your Android project package name.

   Add the `launchMode` attribute to the `application` element. Set its value to `singleTask`.

   [code lang="xml"]
   <application
   android:allowBackup="true"
   android:icon="@drawable/ic_launcher"
   android:label="@string/app_name"
   android:theme="@style/AppTheme"
   android:launchMode="singleTask" >
   [/code]

   Add an `intent-filter` to the `activity` element for notifications.

```
[code lang="xml"]
<activity
android:name=".ActivityName"
android:label="@string/app_name" >
<intent-filter>
<action android:name="com.example.project.app_name.NOTIFICATION" />
<category android:name="android.intent.category.DEFAULT" />
</intent-filter>
[/code]
```

Replace the **com.example.project.app_name** with your Android project package name and the `app_name` string as it appears in the `res/values/string.xml` file. For example: com.sample.notifications.NativeAndroidNotifications.

Add the `GCMIntentService` and add an intent-filter for RECEIVE and REGISTRATION of notifications.

```
[code lang="xml"]
<service android:name="com.worklight.wlclient.push.GCMIntentService" />
<receiver android:name="com.worklight.wlclient.push.WLBroadcastReceiver"
android:permission="com.google.android.c2dm.permission.SEND">
<intent-filter>
<action android:name="com.google.android.c2dm.intent.RECEIVE" />
<category android:name="com.example.project" />
</intent-filter>
<intent-filter>
<action android:name="com.google.android.c2dm.intent.REGISTRATION" />
<category android:name="com.example.project" />
</intent-filter>
</receiver>[/code]
```

Replace all **com.example.project** with your Android project package name.

Android Push Notifications Service

The following ports must be open: 443, 5228, 5229, and 5230. GCM typically uses only 5228, but it sometimes uses 5229 and 5230.
GCM does not provide specific IP addresses, so you must allow your firewall to accept outgoing connections to all IP addresses that are contained in the IP blocks listed in Google ASN of 15169.

Notification Types