Push notifications in native Android applications

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/7.0/notifications/push-notifications-native-android-applications.html) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

Overview

IBM MobileFirst Platform Foundation provides a unified set of API methods to send, or push, notifications to devices where the MobileFirst application is installed. It is possible to send a notification in 3 distinct types: event source notifications, broadcast notifications and tag notifications.

In this tutorial, the concept, API, and usage of push notifications are explained in the context of Native Android applications.
To create and configure an Android native project, first follow these tutorials:

- Creating your first Native Android MobileFirst application (../../hello-world/creating-first-native-android-mobilefirst-application/)
- Invoking adapter procedures from native Android applications (../../server-side-development/invoking-adapter-procedures-native-android-applications/)

The following topics are covered:

- Notification types
- Setting up the project

Notification types

Event source notifications

Event source notifications are notification messages that are targeted to devices with a user subscription.

Broadcast notifications

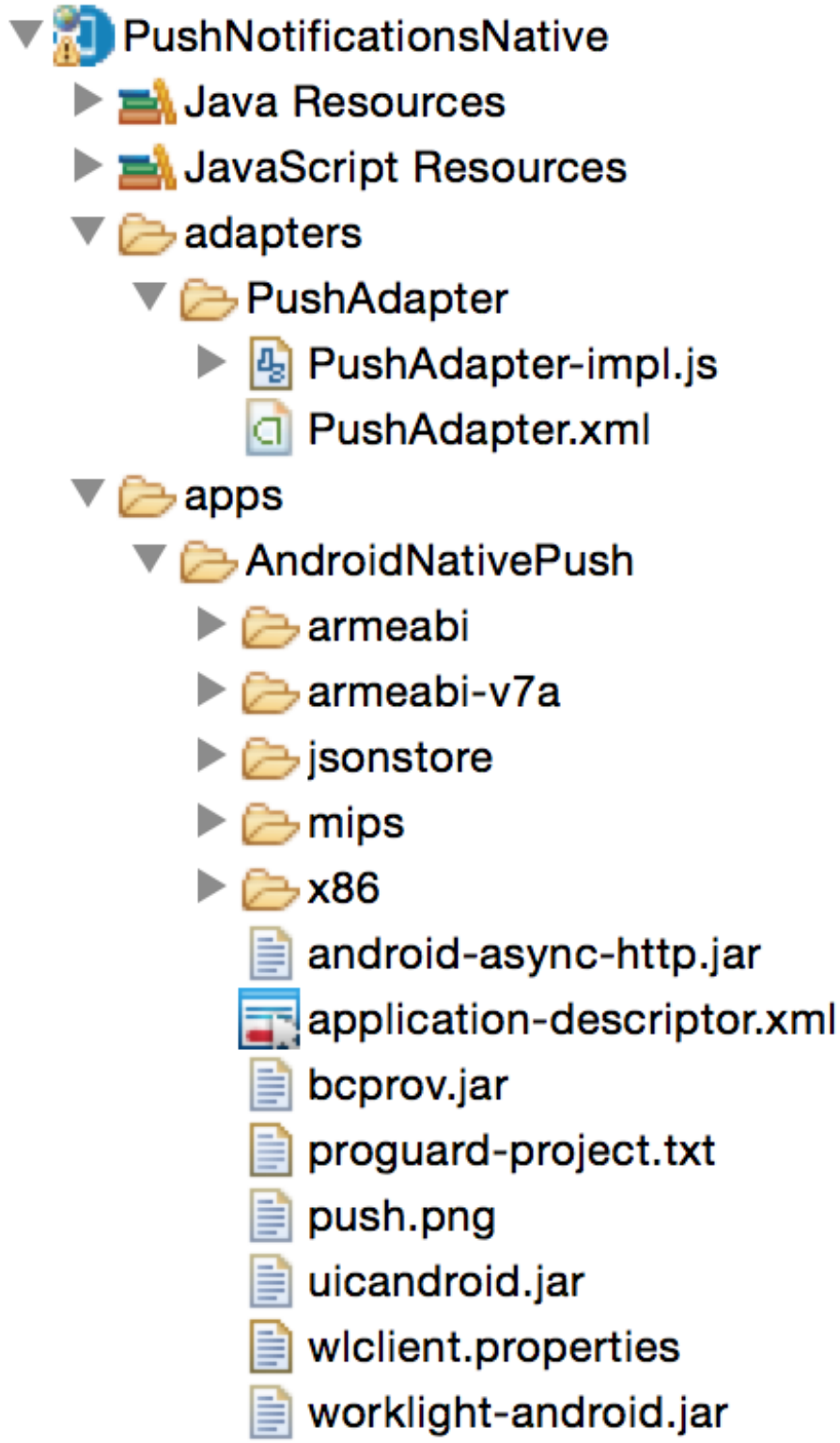
Broadcast notifications are notification messages that are targeted to all subscribed devices.

Tag notifications

Tag notifications are notification messages that are targeted to all subscribed devices to a particular tag.

For more information, select a notification type.

Setting up the project



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 project and put it in your project `res/drawable` folders.

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

- Replace the `key` and `senderId` values with your API key and project number respectively in the `pushSender` tag. In case you do not have these, you can get them from the Google Developer Console (https://console.developers.google.com).
  - Your project number is the `senderId`.
  - Your Android key is the GCM `key`. You can generate it in **API & Auth > Credentials**.

```
<pushSender key="GCM-KEY" senderId="GCM-ID"/>
```

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.
   - wlServerContext – The context root of your MobileFirst Server instance.
   - GcmSenderId – The project number that you obtained through the Google API console.

```
wlServerProtocol = http
wlServerHost =
wlServerPort = 10080
wlServerContext = /PushNotificationsNative/
wlAppId = AndroidNativePush
wlAppVersion = 1.0
wlEnvironment = Androidnative
wlUid = wY/mbnwKTDDYQUvuQCdSgg==
wlPlatformVersion = 6.3.0.00.20141012-0730
#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 =
```

4. ## Add Google Play Services (optional)

   - From **Android SDK Manager > Extras**, add the **Google Play Services** option.
   - Import Google Play Services as a library to the Eclipse workspace:
     1. Select **File > Import**, select **Android > Existing Android Code into workspace**, and browse to the google-play-services_lib project @ `*android_sdk_location*\extras\google\google_play_services\libproject\google-play-services_lib`
     2. After successfully importing google-play-services_lib into the workspace, mark it as an Android library project: Right-click **imported-project > properties > Android** and select the **IsLibrary** checkbox.
   - Right-click the **Android project > properties > Android >** and click **Add**.
     1. In the Project Selection dialog, select the **google-play-services_lib project** and click **OK**.
     2. Click **Apply** and **OK** in the Properties window.
   - Add a reference to the google-play-services version in `your-`

`app\android\native\AndroidManifest.xml` as the first child of the `application` element:

```
<meta-data
   android:name="com.google.android.gms.version"
   android:value="@integer/google_play_services_version" /
>
```

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

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

```
<permission android:name="com.imf.androidnativepush.permission.C2D_MESSAGE" android:pr
otectionLevel="signature" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="com.worklight.androidnativepush.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"/>
```

   Add the `launchMode` attribute to the main `AndroidNativePush` activity. Set its value to `singleTask`.

```
<activity
   android:name="com.worklight.androidnativepush.AndroidNativePush
"
   android:label="@string/app_name"
   android:theme="@android:style/Theme.Black.NoTitleBar"
   android:launchMode="singleTask">
```

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

```
<intent-filter>
   <action android:name="com.worklight.androidnativepush.AndroidNativePush.NOTIFICATION"
/>
   <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

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

```
<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.worklight.androidnativepush" />
   </intent-filter>
   <intent-filter>
      <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
      <category android:name="com.worklight.androidnativepush" />
   </intent-filter>
</receiver>
```