Handling Push Notifications in Android

Overview

In this tutorial, you will be learning how to handle push notification for Android applications.

Tag notifications are notification messages that are targeted to all the devices that are subscribed to a particular tag. Tags represent topics of interest to the user and provide the ability to receive notifications according to the chosen interest.

Broadcast notifications are a form of tag push notifications that are targeted to all subscribed devices. Broadcast notifications are enabled by default for any push-enabled MobileFirst application by a subscription to a reserved Push.all tag (auto-created for every device). Broadcast notifications can be disabled by by unsubscribing from the reserved Push.all tag.

Agenda

- Notifications configuration
- Notifications API
- Handling a push notification
- Handling a secure push notification

Notifications Configuration

1. Create new app in Android Studio
2. Add classpath 'com.google.gms:google-services:2.0.0-alpha3' to Project Gradle
3. Add compile 'com.google.android.gms:play-services-gcm:8.4.0' to app gradle
4. Add compile 'com.squareup.okhttp:okhttp:2.6.0' to app gradle
5. Add apply plugin: 'com.google.gms.google-services' to app gradle
6. Add google-services.json to app folder
7. Add all mfp core jars (bcprov, android sync, worklight-android) to libs folder and the mfpclient.properties to assets folder
8. Add mfp push sdk (ibmmobilefirstplatformfoundationpush) compile dependency in app gradle compile group: 'com.ibm.mobile.foundation', name:'ibmmobilefirstplatformfoundationpush', version:'1.0.0', ext: 'aar', transitive: true
9. Copy ibmmobilefirstplatformfoundationpush-1.0.0.aar (from halpert Electra DevOps Latest integration build) to \extras\google\m2repository\com\ibm\mobile\foundation\ibmmobilefirstplatformfoundationpush\1.0.0\ibmmobilefirstplatformfoundationpush-1.0.0.aar Remove libs folder from the aar. Note: This step is not required once the lib gets to maven central/jcenter. Just need to add mavenCentral()/jcenter() in app gradle.
10. Add the push required configuration in AndroidManifest.xml (Refer the sample)
11. Use the Push SDK APIs in your application (Refer the sample)

Notifications API

API methods for tag notifications

Client-side API

- `WLPush.subscribeTag(tagName,options)` - Subscribes the device to the specified tag name.
- `WLPush.unsubscribeTag(tagName,options)` - Unsubscribes the device from the specified tag name
- `WLPush.isTagSubscribed(tagName)` - Returns whether the device is subscribed to a specified tag name

Common API methods for tag and broadcast notifications

Client-side API

- `WLNotificationListener` - Defines the callback method to be notified when the notification arrives.
- `client.getPush().setWLNotificationListener(listener)` - This method sets the implementation class of the `WLNotificationListener` interface.
- `client.getPush().setOnReadyToSubscribeListener(listener)` - This method registers a listener to be used for push notifications. This listener should implement the `onReadyToSubscribe()` method.
- The `onMessage(props,payload)` method of `WLNotificationListener` is called when a push notification is received by the device. --* *props* – A JSON block that contains the notifications properties of the platform. --* *payload* – A JSON block that contains other data that is sent from MobileFirst Server. The JSON block also contains the tag name for tag-based or broadcast notification. The tag name appears in the “tag” element. For broadcast notification, the default tag name is `Push.ALL`.

Handling a push notification

Handling a secure push notification