Handling Push Notifications in Cordova

Overview

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

To get the application running for Android

1. Create cordova app using cordoav create and mfp cordova template
2. Add android platform
3. Add cordova-plugin-mfp-push plugin. Take it from latest halpert Electra DevOps integration build till its published in npm registry
4. Run cordova build
5. Import the app/platforms/android in Android Studio
6. Add classpath 'com.google.gms:google-services:2.0.0-alpha3' to Module:android gradle. Add jcenter() to repositories in buildscript block
7. Add compile 'com.google.android.gms:play-services-gcm:8.4.0' to app/platforms/android/cordova-plugin-mfp-push/-build-extras.gradle
8. Add compile 'com.squareup.okhttp:okhttp:2.6.0' to app/platforms/android/cordova-plugin-mfp-push/-build-extras.gradle
9. Add apply plugin: 'com.google.gms.google-services' to app/platforms/android/cordova-plugin-mfp-push/-build-extras.gradle
10. Add google-services.json to app/platforms/android folder
11. Add the Push SDK APIs to your application (Refer the sample application)
12. Disable the old push plugin in config.xml. This is reqired till its removed
13. If you want to change the notification title, then add push *notification*tile in strings.xml

To get the application running for iOS

1. Create Cordova project without using cordova mfp template
2. In the cordova-plugin-mfp-push, comment out the dependeny on cordova-plugin-mfp in plugin.xml
3. Run cordova build
4. Open in XCode
5. Add #import "HelloCordova-Swift.h" in AppDelegate.m
6. Declare the notification methods in AppDelegate.m (Refer Sample)
7. Use the Push SDK APIs (Refer Sample)
8. Modify the server and port details in mfpclient.plist

Notifications API

API methods for tag notifications

Client-side API

- `WL.Client.Push.subscribeTag(tagName,options)` - Subscribes the device to the specified tag name.
- `WL.Client.Push.unsubscribeTag(tagName,options)` - Unsubscribes the device from the specified tag name.
- `WL.Client.Push.isPushSupported()` - Returns `true` if push notifications are supported by the platform, or `false` otherwise.
- `WL.Client.Push.isTagSubscribed(tagName)` - Returns whether the device is subscribed to a specified tag name.

Common API methods for tag and broadcast notifications

Client-side API

- `WL.Client.Push.onMessage (props, payload)` - This method is called when a push notification is received by the device.
- **props** - A JSON block that contains the notification 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`.

```
WL.Client.Push.onMessage = function (props, payload) {
    WL.Client.Push.onMessage = function (props, payload) {
        WL.SimpleDialog.show("Tag Notifications", "Provider notification data: " + JSON.stringify(props), [ {
            text : 'Close',
            handler : function() {
                WL.SimpleDialog.show("Tag Notifications", "Application notification data: " + JSON.stringify(paylo
ad), [ {
                    text : 'Close',
                    handler : function() {}
                  }]);
            }
        }]);
    };
}
```

Handling a push notification

Handling a secure push notification