Handling Push Notifications in iOS applications

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/notifications/handling-push-notifications-in-ios/index.md) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

The handling in client side tutorials should explain: - how to setup push notifications support in iOS Xcode project (editing the podfile?) - how to setup push notifications support in Andrid Studio project (editing the builde.gradle file?) - how to setup push notifications support in Cordova applications - how to intercept and display notifications in the client

Overview

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

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.

Jump to:

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

Notifications Configuration

1. Create native project using XCode
2. Add the MobileFirst Platform Foundation SDk, for detailed instructions see Adding the MobileFirst Platform Foundation SDK to iOS Applications (../../adding-the-mfpf-sdk/ios.md)
3.

Old tutorial

Notifications Configuration

1. Create native project using XCode
2. Get the IBMMobileFirstPlatformFoundation.framework and IBMMobileFirstPlatformFoundationPush.framework from halpert and include them in yout project
3. Declare the notification methods in AppDelegate.m (Refer Sample)
4. Use the Push SDK APIs (Refer Sample)
5. Modify the server and port details in mfpclient.plist

Notifications API

API methods for tag notifications

Client-side API

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

Common API methods for tag and broadcast notifications

Client-side API

- `didReceiveRemoteNotification` - When a notification is received by a device, the `didReceiveRemoteNotification` method in the app delegate is called. The logic to handle the notification should be defined here.

```
-(void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo{
    NSLog(@"Received Notification %@",userInfo.description);
}
```

··*`userInfo` – A JSON block that contains the payload field. This field holds other data that is sent from the MobileFirst Platform server. It also contains the tag name for tag and broadcast notification. The tag name appears in the tag element. For broadcast notification, the default tag name is `Push.ALL`. * `setOnReadyToSubscribeListener` - This method registers a listener to be used for push notifications. This listener should implement the `OnReadyToSubscribe()` method. `[[WLPush sharedInstance] setOnReadyToSubscribeListener:readyToSubscribeListener];`

Handling a push notification

Handling a secure push notification