Tag and Broadcast Notifications in Hybrid Applications

Overview

Prerequisite: Make sure to read the Push Notifications in Hybrid Applications (../) tutorial first.

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
- Sample application

Notifications configuration

Tag Notifications configuration

Setting up tags

Tags are defined in the `application-descriptor.xml` file:

Copy

```
<application xmlns="http://www.worklight.com/application-descriptor" id="HybridTagNotifications" platformVer
sion="7.0.0.00.20150312-0731">
...
...
...
<tags>
   <tag>
      <name>my tag 1</name>
      <description>About my tag 1</description>
   </tag>
   <tag>
      <name>my tag 2</name>
      <description>About my tag 2</description>
   </tag>
</tags>
```

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.

Copy

```
WL.Client.Push.onMessage = function (props, payload) {
   alert("Provider notification data: " + Object.toJSON(props));<
   alert("Application notification data: " + Object.toJSON(payload));
}
```

- **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`.

Server-side API

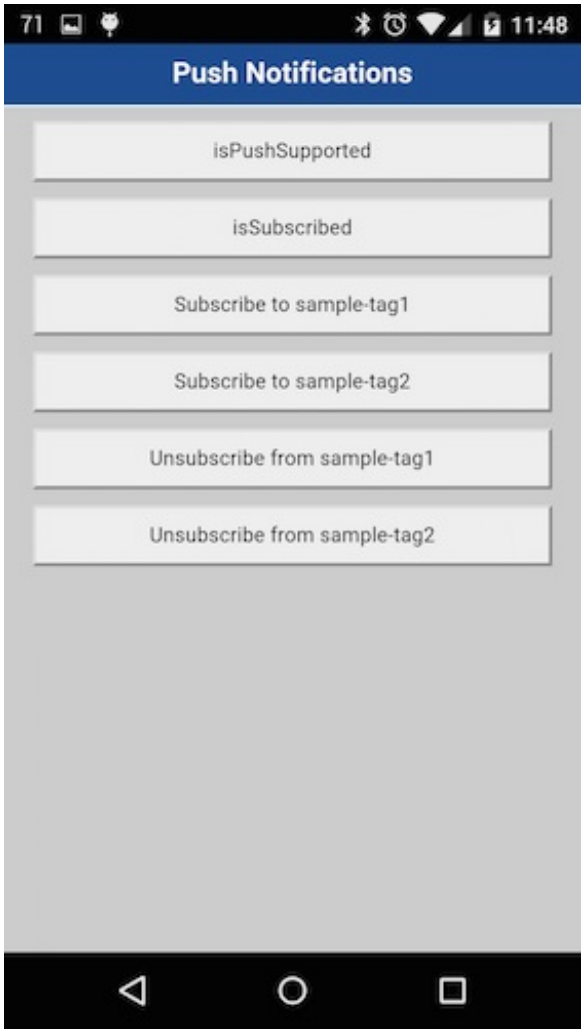
`WL.Server.sendMessage(applicationId,notificationOptions)`
This method submits a notification based on the specified target parameters.
applicationId - (mandatory) The name of the MobileFirst application.
notificationOptions - (mandatory) A JSON block containing message properties.

> For more information about tag and broadcast notifications, see the "tag-based notification" and "broadcast notification" topics in the user documentation.

Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/TagNotifications) the MobileFirst project.

The provided sample application demonstrates how to send a broadcast notification to the default auto-subscribed `push.ALL` tag.
The sample also demonstrates how to send notifications to two predefined tags, sample-tag1 and sample-tag2, to which the user can subscribe in the application.



Sending a notification

To test the application is able to receive a push notification you can perform one of the following:

1. From MobileFirst Studio, right-click the adapter folder, select **Call MobileFirst Adapter** and:
    - If selecting the "sendBroadcastNotification" procedure, provide the application ID and notification text in quotation marks.
    - If selecting the "sendTagNotification" procedure, provide the application ID, notification text and tag name in quotation marks.
    - The application ID can be determined from the `id` attribute in `application-descriptor.xml`:

      Copy

      ```
      <application ... id="HybridTagNotifications" ...>
      ```

2. If using the CLI:

   Copy

   ```
   $ mfp adapter call
   [?] Which endpoint do you want to use? PushAdapter/sendBroadcastNotification
   [?] Enter the comma-separated parameters: "HybridTagNotifications","hello"
   [?] How should the procedure be called? GET
   ```

   Or:

   Copy

   ```
   $ mfp adapter call
   [?] Which endpoint do you want to use? PushAdapter/sendTagNotification
   [?] Enter the comma-separated parameters: "HybridTagNotifications","hello","s
   ample-tag1,sample-tag2"
   [?] How should the procedure be called? GET
   ```