MobileFirst Platform {dev}

Tag and Broadcast Notifications in Native Windows 8 Applications

Relevant to:
Native Windows 8 Universal

Overview

Prerequisite: Make sure that you read the Push notifications in native Windows 8 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). This ability can be disabled by by unsubscribing from the reserved `Push.all` tag.

Agenda

Notifications configuration

Tag Notifications configuration

Setting up tags

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

```
<nativeWindows8App xmlns="http://www.worklight.com/native-windows8-descriptor"
 id="NativeWin8TagNotifications" platformVersion="7.0.0.00.20150312-0731"
 version="1.0">

<pushSender clientSecret="WNS_CLIENT_SECRET" packageSID="WNS_PACKAGE_SID"/>
...
...
...
<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

- `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.
- `WLPush.notificationListener = new MyNotificationListener();`
  Sets the implementation class of the `WLNotificationListener` interface.
- `WLPush.onReadyToSubscribeListener`
  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`.

Server-side API

`WL.Server.sendMessage(applicationId,notificationOptions)`
This method submits a notification based on the specified target parameters and takes two mandatory parameters:

- `applicationId` – (mandatory) The name of the MobileFirst application
- `notificationOptions` – (mandatory) A JSON block containing message properties

> For a full list of message properties, see the `WL.Server.sendMessage` API in the API reference of the user documentation.

Sample application

Before running the application, check the adapter's `PushAdapter-impl.js` file and verify that the `WL.Server.sendMessage()` method uses the correct application name. The correct application name can be determined from the `id` attribute in `application-descriptor.xml`.

Click to download the MobileFirst project.

Click to download the Native project.

- The `TagNotifications` project contains a MobileFirst native API that you can deploy to your

MobileFirst Server instance.

- The `TagNotificationsWin8` project contains a native Windows 8 Universal application that uses a MobileFirst native API library to subscribe to push notifications and receive notifications from Windows Notification Services (WNS).
- Make sure to update the `wlclient.properties` file in the native project with the relevant server settings.