Handling SMS Notifications in Cordova

Overview

SMS notifications are a sub-set of Push Notification, as such make sure to first go through the Push notifications in Cordova (../) tutorials.

Jump to:

- Notifications API
- Using a SMS subscribe servlet
- Sample Application

Notifications API

In SMS notifications, when registering the device, a phone number value is passed.

Register Device

Register the device to the push notifications service.

```
MFPPush.registerNotificationsCallback(notificationReceived);

function registerDevice() {
   var phoneNumber = prompt("Enter Your 10 digit phone number");
   if(phoneNumber != null && phoneNumber!="" && /^\d+$/.test(phoneNumber)) {
      var options = {};
      options.phoneNumber = phoneNumber;
      MFPPush.registerDevice(options,
      function(successResponse) {
         alert("Successfully registered");
         enableButtons();
      }, function(failureResponse) {
         alert("Failed to register");
      });
      return true;
   }

   else {
      alert("Failed to register, You have entered invalid number");
   }
}
```

Using a SMS subscribe servlet

REST APIs are used to send notifications to the registered devices. All forms of notifications can be sent: tag & broadcast notifications, and authenticated notifications

To send a notification, a request is made using POST to the REST endpoint: `imfpush/v1/apps/<application-identifier>/messages`. Example URL:

```
https://myserver.com:443/imfpush/v1/apps/com.sample.sms/messages
```

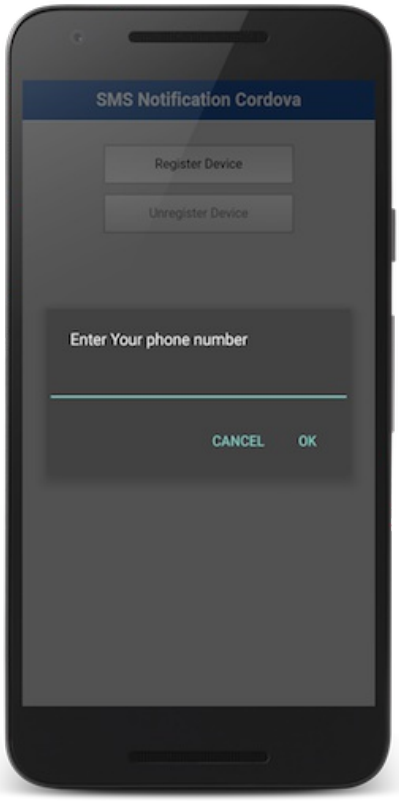
> To review all Push Notifications REST APIs, see the REST API runtime services (https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.apiref.doc/rest_runtime/c_restapi_runtime.html) topic in the user documentation.

To send a notification, see the sending notifications (../sending-push-notifications) tutorial.

Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/SMSNotificationsSwift/tree/release80) the Cordova project.

Note: The latest version of Google Play Services is required to be installed on any Android device for the sample to run.



Sample usage

1. From a **Command-line**, navigate to the project's root folder.
2. Add a platform using the `cordova platform add` command.
3. Register the application by running the command: `mfpdev app register`.
4. In the MobileFirst Operations Console
    - Setup the MobileFirst Server with either GCM key and senderId, APNS certificate or WNS credentials, and define tags.
    - Under **Applications → SMSNotificationsCordova → Security → Map scope elements to security checks**, add a mapping for `push.mobileclient`.
5. Run the application by running the `cordova run` command.

Sending a notification (../sending-push-notifications):

- Tag notification
    - Use the **MobileFirst Operations Console → [your application] → Push → Send Push tab**.