Adding the MobileFirst Platform Foundation SDK to Cordova Applications

Overview

The MobileFirst Platform Foundation SDK provides a set of API methods enabling a developer to implement various MobileFirst features, such as: authentication and security mechanisms, notifications, resource requests, collecting analytics data and more.

> For a complete list of MobileFirst SDK abilities visit the user documentation (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html).

In this tutorial you will learn how to add the MobileFirst SDK to either a new or existing Cordova application created with Apache Cordova, Ionic or other thirdy-party tool. You will also learn how to configure the MobileFirst Server to recognize the application, as well as find information about the MobileFirst configuration files that are changed in the project.

The MobileFirst Cordova SDK is provided as a set of Cordova plug-ins, and is registered at NPM: (cordova-plugin-mfp (https://www.npmjs.com/package/cordova-plugin-mfp)).
The following MobileFirst Cordova SDK plug-ins are available:

- cordova-plugin-mfp
- cordova-plugin-mfp-push
- cordova-plugin-mfp-jsonstore
- cordova-plugin-mfp-fips

cordova-plug-in-mfp
The `cordova-plugin-mfp` plug-in is the core MobileFirst plug-in for Cordova, and is required. If you install any of the other MobileFirst plug-ins the `cordova-plugin-mfp` plug-in is automatically installed as well.

cordova-plugin-mfp-jsonstore
The `cordova-plugin-mfp-jsonstore` plug-in enables your app to use JSONstore. For more information on JSONstore, see the JSONStore tutorial (../client-side-development/jsonstore/).

cordova-plugin-mfp-push
The `cordova-plugin-mfp-push` plug-in provides permissions needed to use push notification from the MobileFirst Server for Android apps. Additional setup for using push notification is required. For more information on push notification, see the Push notifications tutorial (../notifications/push-notifications-overview/).

cordova-plugin-mfp-fips
The `cordova-plugin-mfp-fips` plug-in enables FIPS related features. For more information about FIPS, see FIPS (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html) in the user docuementation.

Pre-requisites:

- Apache Cordova CLI and MobileFirst developer CLI installed on the developer workstation.
- *Optional* MobileFirst Server to run a locally.
- Make sure you have read the Setting up your MobileFirst development environment (../../setting-up-the-mobilefirst-development-environment) tutorial.

Jump to:

Adding the MobileFirst Cordova Plug-in

Before starting, make sure the MobileFirst Server is running.
If using a locally installed server: From a **Command-line** window, navigate to the server's **scripts** folder and run the command: `./start.sh` in Mac and Linux or `start.cmd` in Windows.

Follow the below instructions to add the MobileFirst Cordova Plugin to either a new or existing Cordova project:

1. Create a Cordova project or use an existing one. The MobileFirst template app can be used to create a new Cordova project that already contains the MobileFirst Plug-in. To use the template, open the **Command-line** and run the command:

   ```
   cordova create myapp  --template cordova-template-mfp
   ```

   The ***myapp*** folder will be created with a copy of the cordova-template-mfp project
2. Navigate to the root of the Cordova project.
3. Add the MobileFirst Cordova Plugin (cordova-plugin-mfp (https://www.npmjs.com/package/cordova-plugin-mfp)) using the cordova CLI. This will install the plugin from NPM (https://www.npmjs.com/package/cordova-plugin-mfp):

   ```
   cordova plugin add cordova-plugin-mfp
   ```

4. The optional plug-ins `cordova-plugin-mfp-push`, `cordova-plugin-mfp-jsonstore` and `cordova-plugin-mfp-fips` can be added using the Cordova CLI command `cordova add plugin [plug-in-name]`.
   For example, to add the `cordova-plugin-mfp-push` plug-in:

   ```
   cordova plugin add cordova-plugin-mfp-push
   ```

5. Optionally add one or more platforms to the Cordova project using the Cordova CLI command `cordova platform add [platform]`.
   To add the iOS platform:

   ```
   cordova platform add ios
   ```

   To add the Android platform:

   ```
   cordova platform add android
   ```

   To add the Windows platform:

```
cordova platform add windows
```

> The platform versions supported by MobileFirst plug-ins are **cordova-ios@4.0.1**, **cordova-android@5.0.0** and **cordova-windows@4.2.0**

Registering the Cordova app in MobileFirst Server

1. Open the **Command-line** and navigate to the root of the Cordova project.
2. Register the application with MobileFirst Server with the CLI command:

   ```
   mfpdev app register
   ```

   The `mfpdev app register` CLI command first connects to the MobileFirst Server to register the application, followed by generating the `config.xml` file at the root of the Cordova project, and adding to it the metadata that identifies the MobileFirst Server. Each platform is registered as an application in MobileFirst Server.

   > The application registration can also be done from the MobileFirst Operations Console:
   > 1. Open your browser of choice and load the MobileFirst Operations Console using the address `http://localhost:9080/mfpconsole/`. You can also open the console from the **Command-line** using the CLI command `mfpdev server console`.
   > 2. Click on the "Create new" button next to "Applications" to create a new application. Follow the on-screen instructions.
   > 3. After successfully registering your application you can optionally download a "skeleton" Cordova project pre-bundled with the MobileFirst Cordova SDK.

> **Tip:** Learn more about the various CLI commands in the Using CLI to manage MobileFirst artifacts (../../client-side-development/using-cli-to-manage-mobilefirst-artifacts/) tutorial.

Running the application on emulator or on a real device

Preview the Application

The application can be previewed with the CLI command

```
mfpdev app preview
```

The preview can be done in two ways, with Simple Browser Rendering or with the IBM Mobile Browser Simulator.
With Simple Browser Rendering the app is presented as a web page in your browser while the IBM Mobile Browser Simulator will simulate a Mobile Device in your browser and the app is presented inside the simulated device.

```
? Select how to preview your app: (Use arrow keys)
❯ browser: Simple browser rendering
mbs: Mobile Browser Simulator
```

Running the application on emulator or on a real device

Use the Cordova CLI to run the application on a emulator or on a real device.
To emulate the application execute the Cordova CLI command `cordova emulate [platform]`.

Preview the application in the iOS Simulator:

```
cordova emulate ios
```

Preview the application in the Android Emulator:

```
cordova emulate android
```

To run the application on a real device attached to the development workstation, execute the Cordova CLI command `cordova run [platform]`

Run the application on an iOS device:

```
cordova run ios
```

Run the application on an Android device:

```
cordova run android
```

Generated MobileFirst Native SDK artifacts

Two MobileFirst-related artifacts are available in the Cordova project after it has been integrated with the MobileFirst Cordova SDK: the `config.xml` and the `application-descriptor.json` file.

config.xml

When the MobileFirst Cordova plug-in is added to the project, the Cordova-generated `config.xml` file receives a set of new settings identified with the namespace `mfp:`. The added elements contain information related to MobileFirst features and the MobileFirst Server. Here is an example of MobileFirst settings added to the `config.xml` file:

```
<mfp:android>
   <mfp:sdkChecksum>3563350808</mfp:sdkChecksum>
   <mfp:appChecksum>0</mfp:appChecksum>
   <mfp:security>
      <mfp:testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, m
p3" />
   </mfp:security>
</mfp:android>
<mfp:platformVersion>8.0.0.00-20151214-1255</mfp:platformVersion>
<mfp:clientCustomInit enabled="false" />
<mfp:server runtime="mfp" url="http://10.0.0.1:9080" />
<mfp:directUpdateAuthenticityPublicKey />
<mfp:languagePreferences>en</mfp:languagePreferences>
```

- **mfp:android:** Root element for settings specific for the android platform
- **mfp:ios:** Root element for settings specific for the ios platform
- **mfp:windows:** Root element for settings specific for the windows platform
- **mfp:sdkChecksum:** Checksum of the SDK in use
- **mfp:appChecksum:** Checksum of the app
- **mfp:security:** Root element for security configurations
- **mfp:testWebResourcesChecksum:** Enables or Disables the test for web resources checksum
- **mfp:platformVersion:** The version of the MobileFirst SDK in use
- **mfp:clientCustomInit:** Enables or Disables custom initialization, when WL.Client.init is not automatically executed
- **mfp:server:** MobileFirst Server URL and Runtime definition
- **mfp:directUpdateAuthenticityPublicKey:** The public key used for direct update authenticity
- **mfp:languagePreferences:** Default language for client sdk system messages (en, fr, es)

Editing MobileFirst settings in config.xml with MobileFirst developer CLI
The MobileFirst developer CLI can be used to edit the above settings with the command:

```
mfpdev app config
```

application-descriptor.json

Located in the **<cordova-project-root-directory>/mobilefirst/[platform]** folder, this file contains application configuration settings such as its `bundleId` and `version` and is user-editable.

The file can be edited either locally or via the MobileFirst Operations Console.
If edited locally, the MobileFirst Server can be updated by running the CLI command: `mfpdev app push`.
The file can also be updated by pulling from the server its latest revision by running the CLI command: `mfpdev app pull`.

```
{
   "applicationKey": {
      "packageName": "com.samplePackage",
      "version": "1.0",
      "clientPlatform":"android"
   }

   ...
   ...
   ...
}
```

Tutorials to follow next

Now that the MobileFirst Cordova plugin is added to the application you can continue reading tutorials for Cordova development (../../hybrid-tutorials/) to learn more about authentication and security, server-side development, notifications, and more.