Adding the MobileFirst Platform Foundation SDK to Cordova Applications

Overview

The MobileFirst Platform Foundation SDK provides a set of API methods enabling developers 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).

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

- 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 if not already installed.

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 (../../using-the-mfpf-sdk/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 applications. 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 the user documentation topic for FIPS (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.

Prerequisites:

- 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 SDK
- Updating the MobileFirst Cordova SDK
- Running the application on emulator or on a real device
- Generated MobileFirst Native SDK artifacts
- Tutorials to follow next

Adding the MobileFirst Cordova SDK

Follow the below instructions to add the MobileFirst Cordova SDK to either a new or existing Cordova project, and registering it in the MobileFirst Server.

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.

Adding the SDK

1. Create a Cordova project or use an existing one.

   Consider creating the project using the MobileFirst Cordova **application template**. The template adds to the Cordova project's **config.xml** file requierd MobileFirst-specific plug-in entries, as well as provides a MobileFirst-specific, ready-to-use, **index.js** file adjusted for MobileFirst application development.

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

2. Navigate to the root of the Cordova project: `cd myapp`

3. Add one or more supported platforms to the Cordova project using the Cordova CLI command: `cordova platform add ios|android|windows`. For example:

   ```
   cordova platform add ios
   ```

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

Registering the Cordova application in MobileFirst Server

1. Open a **Command-line** window and navigate to the root of the Cordova project.

2. Register the application with MobileFirst Server:

   ```
   mfpdev app register
   ```

   The `mfpdev app register` CLI command first connects to the MobileFirst Server to register the application, followed by updating the **config.xml** file at the root of the Cordova project with 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 "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.

3. Run the command:

```
mfpdev app pull
```

The `mfpdev app pull` CLI command creates the **mobilefirst** folder at the root of the Xcode project and downloads into it the `application-descriptor.json` file, containing application configuration data.

These files are further explained in the Generated MobileFirst Native SDK artifacts section below.

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

Updating the MobileFirst Cordova SDK

To update the MobileFirst Cordova SDK with the latest release, the **cordova-plugin-mfp** plug-in needs to be removed using the `cordova plugin remove cordova-plugin-mfp` command, followed by re-adding it: `cordova plugin add cordova-plugin-mfp`.

SDK releases can be found in the SDK's NPM repository (https://www.npmjs.com/package/cordova-plugin-mfp).

Generated MobileFirst Cordova SDK artifacts

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

config.xml

Once the MobileFirst Cordova SDK 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,
mp3" />
  </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

With the MobileFirst Cordova SDK now integrated, you can now:

- Review the Using the MobileFirst Platform Foundation SDK tutorials (../../using-the-mfpf-sdk/)
- Review the Adapters development tutorials (../../adapters/)
- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)