Adding the MobileFirst Foundation SDK to Cordova Applications

Overview

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 (https://www.npmjs.com/package/cordova-plugin-mfp).
Available plug-ins are:

- **cordova-plugin-mfp** - the core SDK plug-in
- **cordova-plugin-mfp-push** - provides push notifications support
- **cordova-plugin-mfp-jsonstore** - provides JSONStore support
- **cordova-plugin-mfp-fips** - *Android only*. Provides FIPS support
- **cordova-plugin-mfp-encrypt-utils** - *iOS only*. Provides encrypt/decrypt support

> Learn more about the Cordova plug-ins (https://www.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/dev/c_cord_plugins.html) that compose the MobileFirst Cordova SDK in the user documentation.

Support level: The platform versions supported by the MobileFirst plug-ins, at **minimum**, are **cordova-ios@4.0.1**, **cordova-android@5.1.1** and **cordova-windows@4.2.0**.

Jump to:

Cordova SDK components

cordova-plugin-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.

> The following cordova plug-ins will be installed as a dependency of cordova-plugin-mfp:
>
> - cordova-plugin-device
> - cordova-plugin-dialogs
> - cordova-plugin-globalization
> - cordova-plugin-okhttp

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/).

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/).

Prerequisites:

- Apache Cordova CLI 6.x (https://www.npmjs.com/package/cordova), and MobileFirst CLI installed on the developer workstation.
- MobileFirst Server to run locally, or a remotely running MobileFirst Server.
- Make sure you have read the Setting up your MobileFirst development environment (../../setting-up-your-development-environment/mobilefirst-development-environment) and Setting up your Cordova development environment (../../setting-up-your-development-environment/cordova-development-environment) tutorials.

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 folder and run the command: `./run.sh`.

> **Note:** If adding the SDK to an existing Cordova application, the plug-in overwrites the `MainActivity.java` for Android and `Main.m` for iOS.

Adding the SDK

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

New Application

1. Create a Cordova project:

   ```
   cordova create Hello com.example.helloworld HelloWorld --template cordova-template-mfp
   ```

   - "Hello" is the folder name of the application
   - "com.example.helloworld" is the ID of the application
   - "HelloWorld" is the Name of the application
   - --template modifies the application with MobileFirst-specific additions

   > The templated **index.js** enables use of additional MobileFirst features as such Multilingual application translation (../../using-the-mfpf-sdk/translation) and initialization options (see the user documentation for more information).

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
   ```

   > **Note:** Because the application was configured using the MobileFirst template, the MobileFirst core Cordova plug-in is added automatically as the platform is added in step 3.

Existing Application

1. Navigate to the root of your existing Cordova project and add the MobileFirst core Cordova plug-in:

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

2. Navigate to the **www\js** folder and select the **index.js** file.
3. Add the following function:

```
function wlCommonInit() {

}
```

The MobileFirst-provided API methods are available once the MobileFirst client SDK has been loaded. Once loaded, the `wlCommonInit` function is then called.
Use this function when trying to call the various MobileFirst-provided API methods.

Registering the application

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
   ```

   - If a remote server is used, use the command `mfpdev server add` (../../using-the-mfpf-sdk/using-mobilefirst-cli-to-manage-mobilefirst-artifacts/#add-a-new-server-instance) to add it.

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.

> **Tip:** The application registration can also be performed from the MobileFirst Operations Console:
>
> 1. Load the MobileFirst Operations Console.
> 2. Click the "New" button next to "Applications" to register a new application and follow the on-screen instructions.

Using the SDK

The MobileFirst-provided API methods are available once the MobileFirst client SDK has been loaded. Once loaded, the `wlCommonInit` function is then called.
Use this function when trying to call the various MobileFirst-provided API methods.

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

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

The MobileFirst CLI can be used to edit the above settings with the command:

```
mfpdev app config
```

Tutorials to follow next

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

- Review the Using the MobileFirst 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)