

# 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](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 Native SDK as an Apache Cordova plugin (cordova-plugin-mfp (<https://www.npmjs.com/package/cordova-plugin-mfp>)) using NPM to either a new or existing Cordova application. 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.

**Pre-requisites:** Apache Cordova CLI and MobileFirst CLI installed on the developer workstation. 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 Plugin
- Registering the Cordova app in MobileFirst Server
- Running the application on emulator or on a real device
- Generated MobileFirst Native SDK artifacts
- Tutorials to follow next

## Adding the MobileFirst Cordova Plugin

Before starting, make sure the MobileFirst Server is running.

From **Terminal** run the command:

```
mfpdev server start
```

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. To use the template app, open **Terminal** and run the command: `bash cordova create myapp com.ibm.app 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): bash
cordova plugin add cordova-plugin-mfp
```

4. Add one or more platforms to the project using the Cordova CLI To add ios platform:

```
cordova platform add ios
```

To add android platform:

```
cordova platform add android
```

## Registering the Cordova app in MobileFirst Server

1. Register the application with MobileFirst Server with the 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 a native app in MobileFirst Server.

> TODO: Not sure if the following is valid for Cordova

> 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:10080/mfpconsole/`. You can also open the console from **Terminal** 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" Android Studio project pre-bundled with the MobileFirst Native 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.



## Preview, emulate or run application on a device

### Preview the Application

1. After registered in a MobileFirst Server, the app can be previewed with the command

```
mfpdev app preview
```

The preview can be done in two ways, with Simple Browser Rendering or with Mobile Browser Simulator. With Simple Browser Rendering the app is presented as a web page in your browser while 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

1. Use the Cordova CLI to run the application on a emulator or on a real device.

To emulate the application execute the command `cordova emulate`

example for ios:

```
cordova emulate ios
```

example for android:

```
cordova emulate android
```

To run the application on a real device attached to the development machin, execute the command `cordova run` example for ios:

```
cordova run ios
```

example for android:

```
cordova run android
```

## Generated MobileFirst Native SDK artifacts

Two MobileFirst-related artifacts are available in the Xcode project after it has been integrated with the MobileFirst Native SDK: the `mfpclient.plist` and the `application-descriptor.json` file.

### mfpclient.plist

Located at the root of the project, this file contains server configuration properties and is user-editable:

- `protocol` – The communication protocol to MobileFirst Server. Either `HTTP` or `HTTPS`.
- `host` – The hostname of the MobileFirst Server instance.
- `port` – The port of the MobileFirst Server instance.
- `wlsServerContext` – The context root path of the application on the MobileFirst Server instance.
- `languagePreference` - Sets the default language for client sdk system messages

### application-descriptor.json

Located in the **<xcode-project-root-directory>/mobilefirst** folder, this file contains application configuration settings such as its `bundleIdversion` and is user-editable.

If edited, be sure to update the MobileFirst Server by running the CLI command: `mfpdev app push`. The file can also be updated by pulling from the server its latest variation by running the CLI command: `mfpdev app pull`.

The file can be edited via the MobileFirst Operations Console.

```
{
  "applicationKey": {
    "bundleId": "com.sampleone.bankApp",
    "version": "1.0",
    "clientPlatform": "ios"
  },
  ...
  ...
  ...
}
```

## Tutorials to follow next

Now that the application is integrated with the MobileFirst Native SDK you can continue reading tutorials for Native iOS development ([../ios-tutorials/](#)) to learn more about authentication and security, server-side development, notifications, and more.