Cordova end-to-end demonstration

fork and edit tutorial (https://github.ibm.com/MFPSamples/DevCenter/tree/master/tutorials/en/foundation/8.0/quick-start/cordova/index.md) | report issue (https://github.ibm.com/MFPSamples/DevCenter/issues/new)

Overview

The purpose of this demonstration is to experience an end-to-end flow where an application and an adapter are registered using the MobileFirst Operations Console, an "skeleton" Cordova project is downloaded and edited to call the adapter, and the result is displayed - verifying a successful connection with the MobileFirst Server.

Prerequisites:

- Xcode for iOS, Android Studio for Android or Visual Studio 2013/2015 for Windows 8.1 Universal / Windows 10 UWP
- MobileFirst Developer CLI (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))
- *Optional*. Stand-alone MobileFirst Server (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))

1. Starting the MobileFirst Server

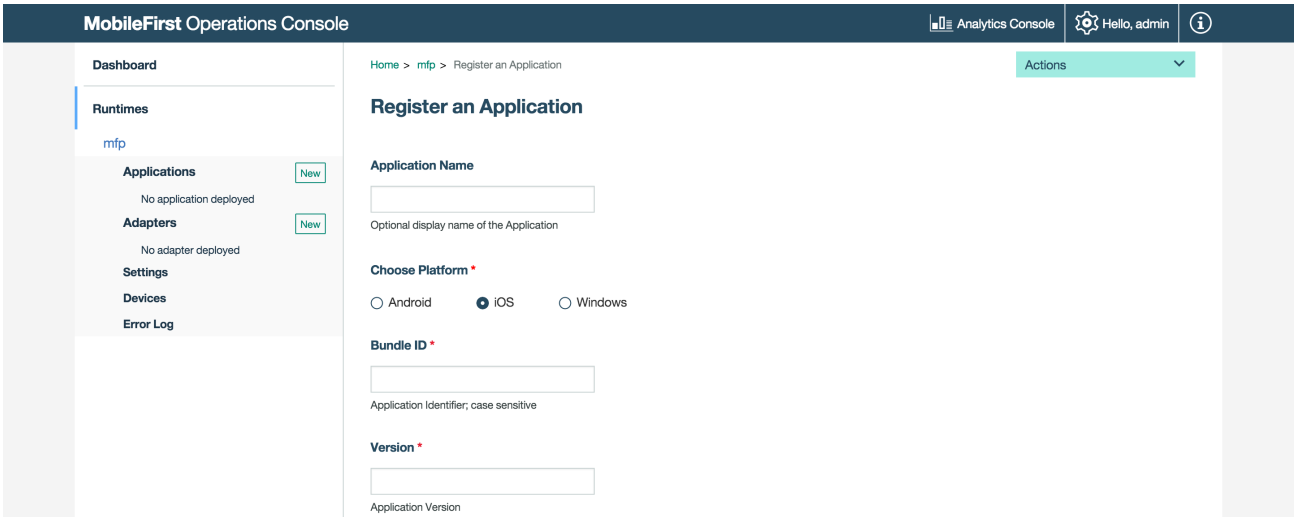
> If a remote server was already set-up, skip this step.

From a **Command-line** window, navigate to the server's folder and run the command: `./run.sh` in Mac and Linux or `run.cmd` in Windows.

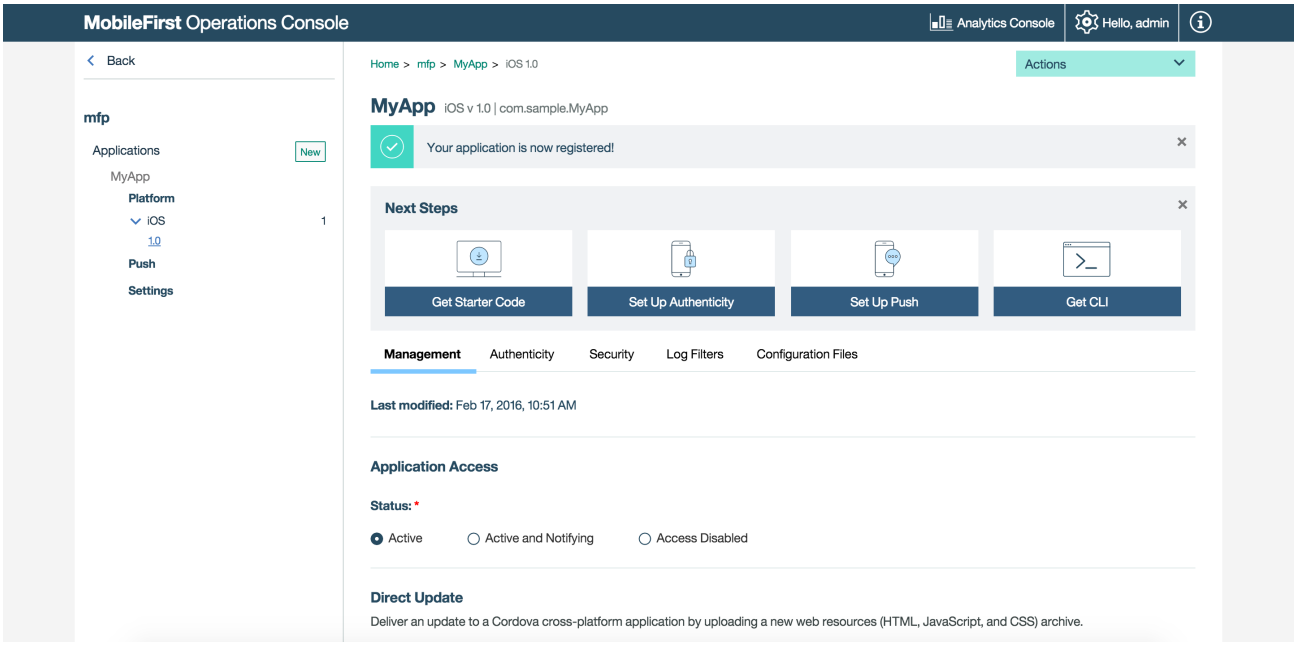
2. Creating an application

In a browser window, open the MobileFirst Operations Console by loading the URL: `http://your-server-host:server-port/mfpconsole`. If running locally, use: http://localhost:9080/mfpconsole (http://localhost:9080/mfpconsole). The username/password are *admin/admin*.

1. Click on the "New" button next to **Applications** and select the desired *platform*, *identifier* and *version* values.



2. Click on the **Get Starter Code** tile and select to download the Cordova Starter Code.



3. Editing application logic

1. Open the Cordova project in your code editor of choice.
2. Select the **www/js/index.js** file and paste the following code snippet, replacing the existing `wlCommonInit()` function:

```
function wlCommonInit() {
    var resourceRequest = new WLResourceRequest(
        "/adapters/javaAdapter/users/world",
        WLResourceRequest.GET
    );

    resourceRequest.send().then(
        function(response) {
            // Will display "Hello world" in an alert dialog.
            alert("Success: " + response.responseText);
        },
        function(response) {
            alert ("Failure: " + response.errorMsg);
        }
    );
}
```

4. Creating an adapter

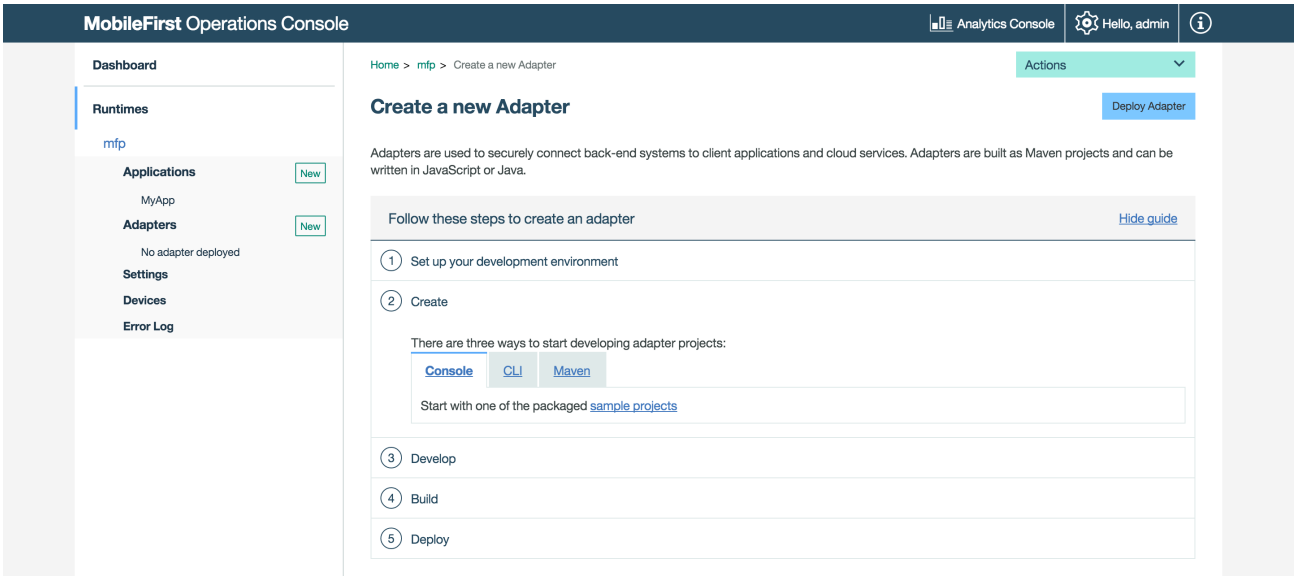
1. Click on the "New" button next to **Adapters**
    - Select the **Actions → Download sample** option. Download the **Java** adapter sample.

> If Maven and MobileFirst CLI are not installed, follow the on-screen **Setting up your environment** instructions to install.

- From a **Command-line** window, navigate to the adapter's Maven project root folder and run the command:

```
mfpdev adapter build
```

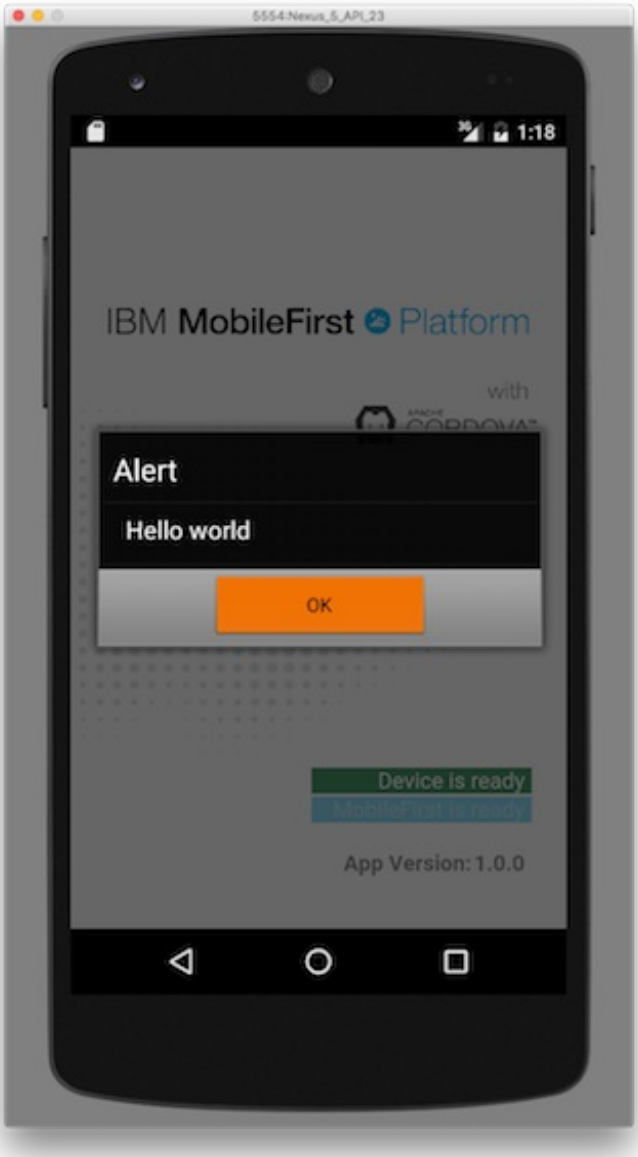
- When the build finishes, deploy it from the MobileFirst Operations Console using the **Actions → Deploy adapter** action. The adapter can be found in the **[adapter]/target** folder.
- Alternatively, download this prepared .adapter artifact and deploy it from the MobileFirst Operations Console using the **Actions → Deploy adapter** action.



5. Testing the application

1. In the Cordova project, select the **config.xml** file and edit the `<mfp:server ... url=" "/>` value with the IP address of the MobileFirst Server.
2. From a **Command-line** window, navigate to the Cordova project root folder.
3. Run the command: `cordova platform add ios/android/windows` to add a platform.
4. Run the command: `cordova run`.

If a device is connected, the application will be installed and launched in the device,
Otherwise the Simulator or Emulator will be used.



Next steps

- Review the Client-side development tutorials (../../using-the-mfpf-sdk/)
- Review the Server-side development tutorials (../../adapters/)
- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)