

Cordova end-to-end demonstration

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 **scripts** folder and run the command: `./start.sh` in Mac and Linux or `start.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 "Create 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 "Create new" button next to **Adapters** and download the **Java** adapter sample.

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

The screenshot shows the 'MobileFirst Operations Console' interface. The left sidebar contains navigation links: Dashboard, Runtimes, mfp, Settings, Applications, Adapters, Devices, Client Logs, and Error Log. The main content area is titled 'Create a new Adapter' and includes a breadcrumb 'Home > mfp > Create a new Adapter'. A message states: 'It seems like you don't have any adapters, lets get started'. A 'Deploy Adapter' button is in the top right. Below this is a section 'Follow these steps to set up an adapter' with a 'Hide guide' link. The steps are: 1. Setting up your environment, 2. Start with a sample adapter (with sub-tabs for Console, CLI, and Maven), 3. In your IDE of choice, edit the adapter code - REST end points and adapter descriptor, 4. Build and package, and 5. Upload adapter. A console log snippet is visible under step 2.

The screenshot shows the 'MobileFirst Operations Console' interface. The left sidebar is the same as the previous screenshot. The main content area is titled 'Get Starter Code' with a breadcrumb 'Home > mfp > Get Starter Code'. It has tabs for 'Application' and 'Adapter'. Under 'Adapter', there are sections for 'Java Adapter Samples' and 'Javascript Adapter Samples', each with a 'Select the sample that you would like to download' instruction. The Java section has buttons for 'Java' and 'Java-security check'. The Javascript section has buttons for 'Javascript-SQL', 'Javascript-SAP', 'Javascript-JMS', and 'Javascript-HTTP'. At the bottom, there is a 'Next Step' section with a diagram and a 'Go to Adapter Quickstart' button.

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

```
mpfdev adapter build
```

- When the build finishes, run the command:

```
mpfdev adapter deploy
```

If using a remote MobileFirst Server, run the command:

```
mpfdev adapter deploy Replace-with-remote-server-name
```

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](#))

