Android end-to-end demonstration

Overview

The purpose of this demonstration is to experience an end-to-end flow where an application is quickly created using the MobileFirst Operations Console and connectivity is verified with the MobileFirst Server.

Prerequisites:

- Configured Android Studio
- *Optional* Stand-alone MobileFirst Server (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))

1. Starting the MobileFirst Server

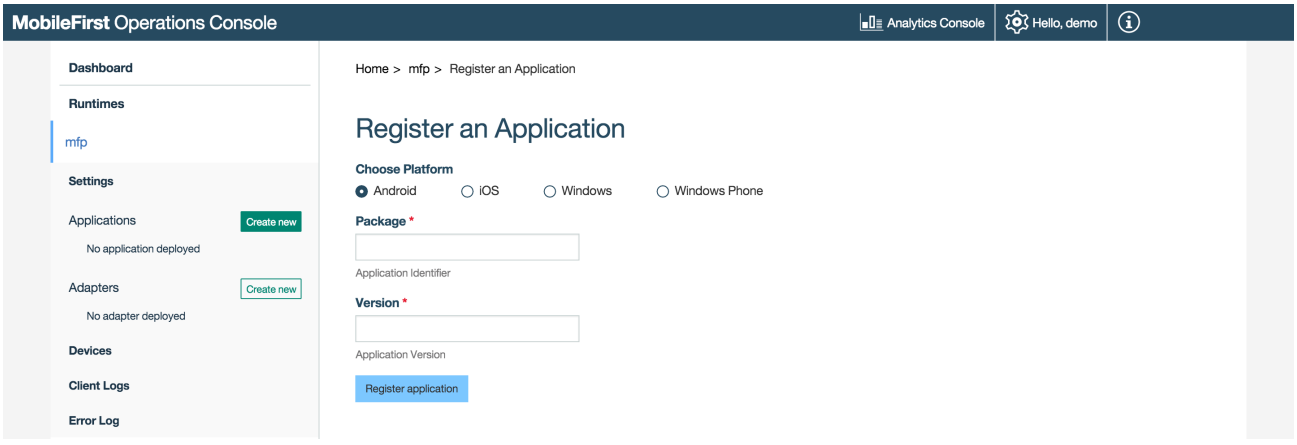
> If a remote server was already set-up, skip this step.

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.

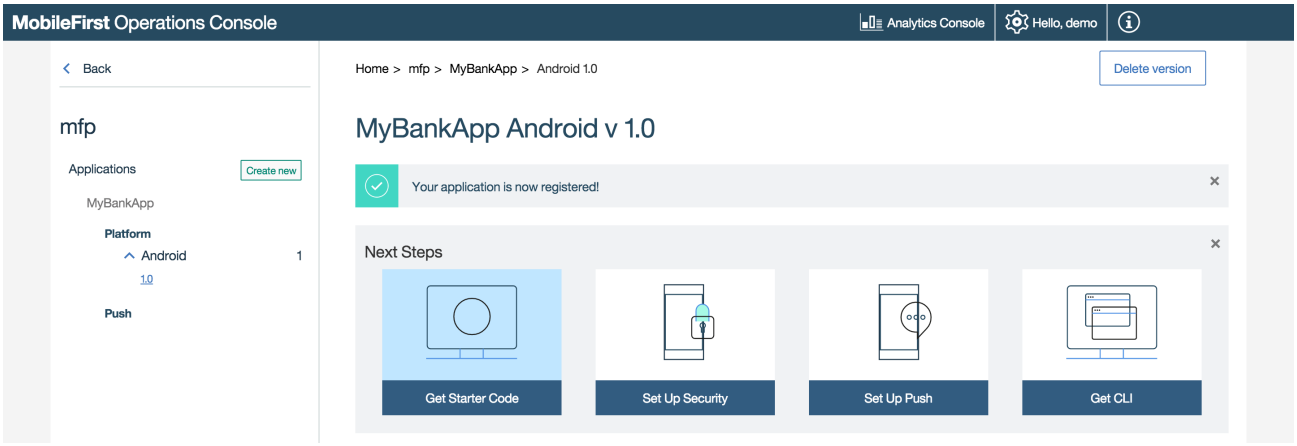
2. Creating an application

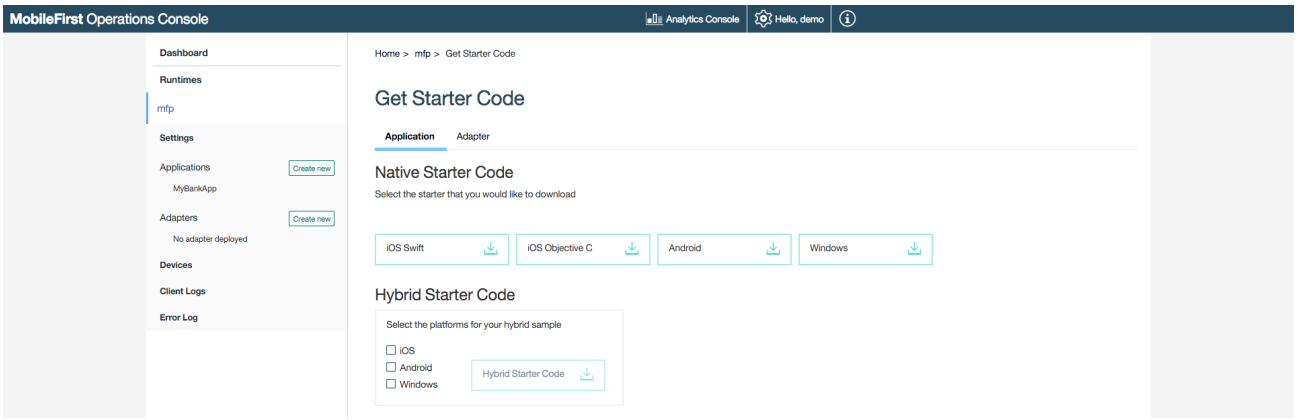
In a browser window, open the MobileFirst Operations Console by loading the URL: `http://your-server-host:server-port/mfpconsole`. If running locally, use: http://localhost:9080/mfpconsole (http://localhost:9080/mfpconsole). The username/password are *admin/admin*.

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 Android Starter Code.





3. Editing application logic

1. Open the Android Studio project.
2. Select the **app/java/com.mfp.sample/MainActivity.java** file and paste the following code snippet:

```
WLResourceRequest code snippet here
```

4. Creating an adapter

1. Click on the "Create new" button next to **Adapters** and download the **JavaScript-HTTP** adapter sample.

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

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

```
mfpdev adapter build
```

3. When the build finishes, run the command:

```
mfpdev adapter deploy
```

If using a remote MobileFirst Server, run the command:

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

5. Testing the application

1. In Android Studio, click on the **Run App** button.

Next steps

- Review the Client-side development tutorials (../../client-side-development/)
- Review the Server-side development tutorials (../../server-side-development/)
- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review All Tutorials (../../all-tutorials)