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
- 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

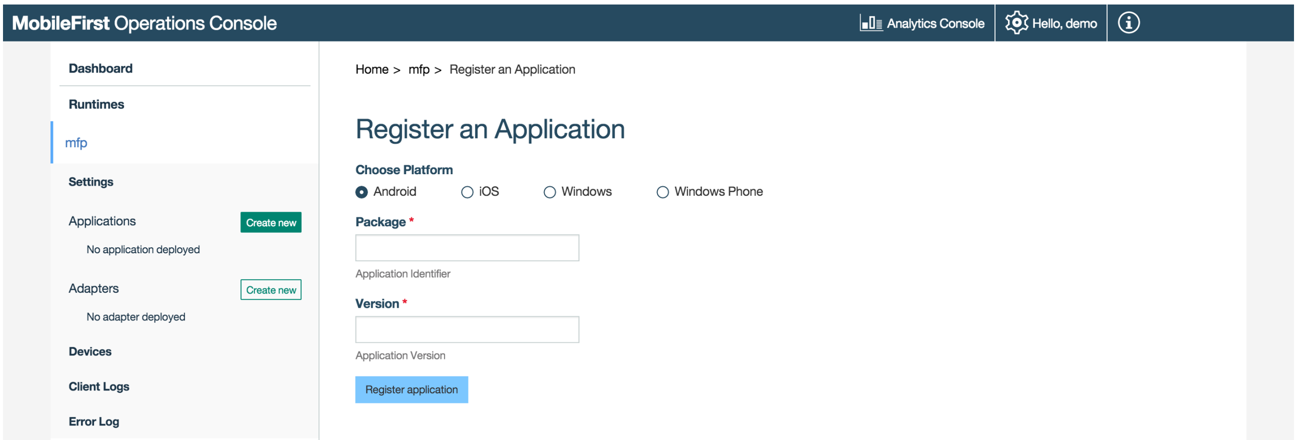
> If a remote server was already set-up, skip this step.

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.

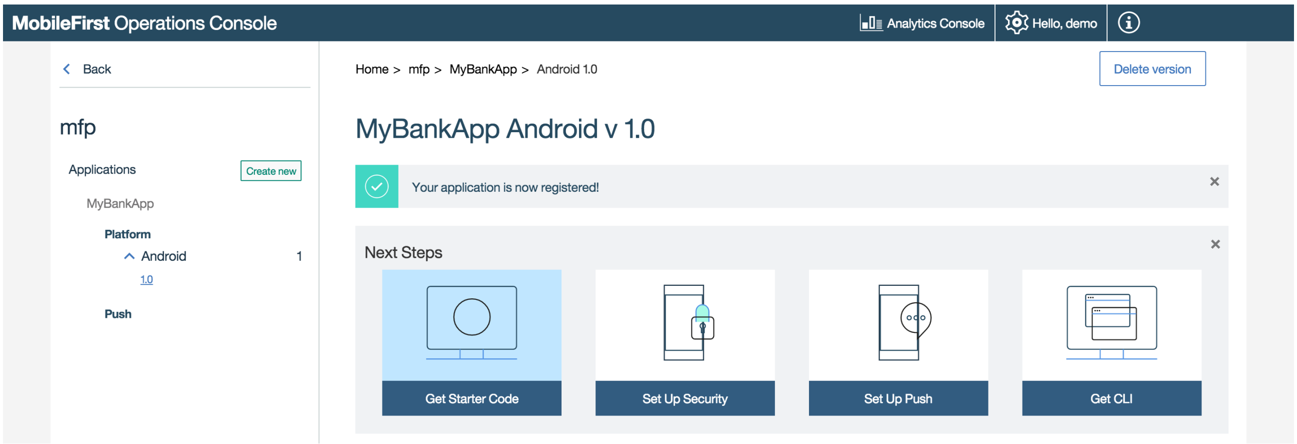
2. Creating an application

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.

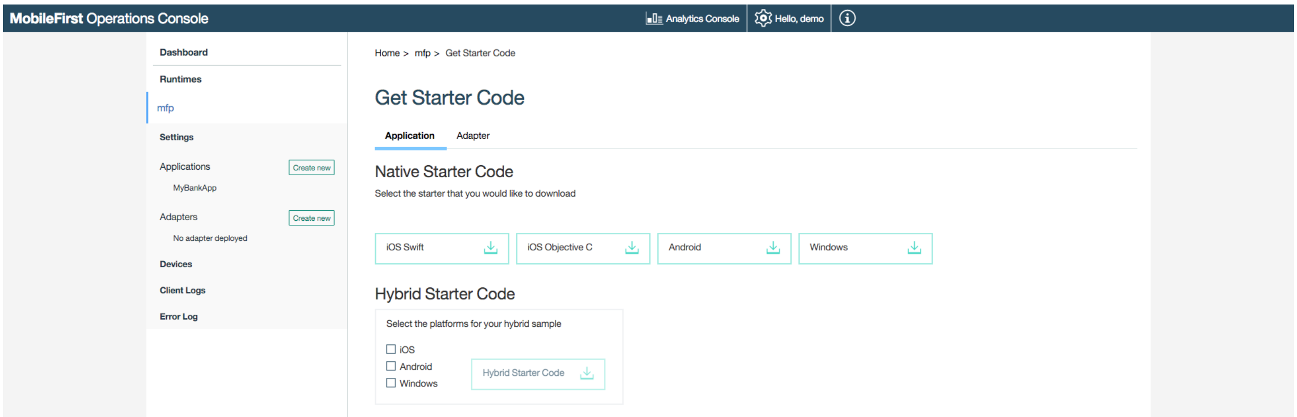


2. Click on the **Get Starter Code** tile and select to download the Android Starter Code.



MyBankApp Android v 1.0

Your application is now registered!



Get Starter Code

Native Starter Code

Select the starter that you would like to download

Hybrid Starter Code

Select the platforms for your hybrid sample

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 snippets:

- Imports:

```
import com.worklight.wlclient.api.*;
import java.net.URI;
import android.util.Log;
```

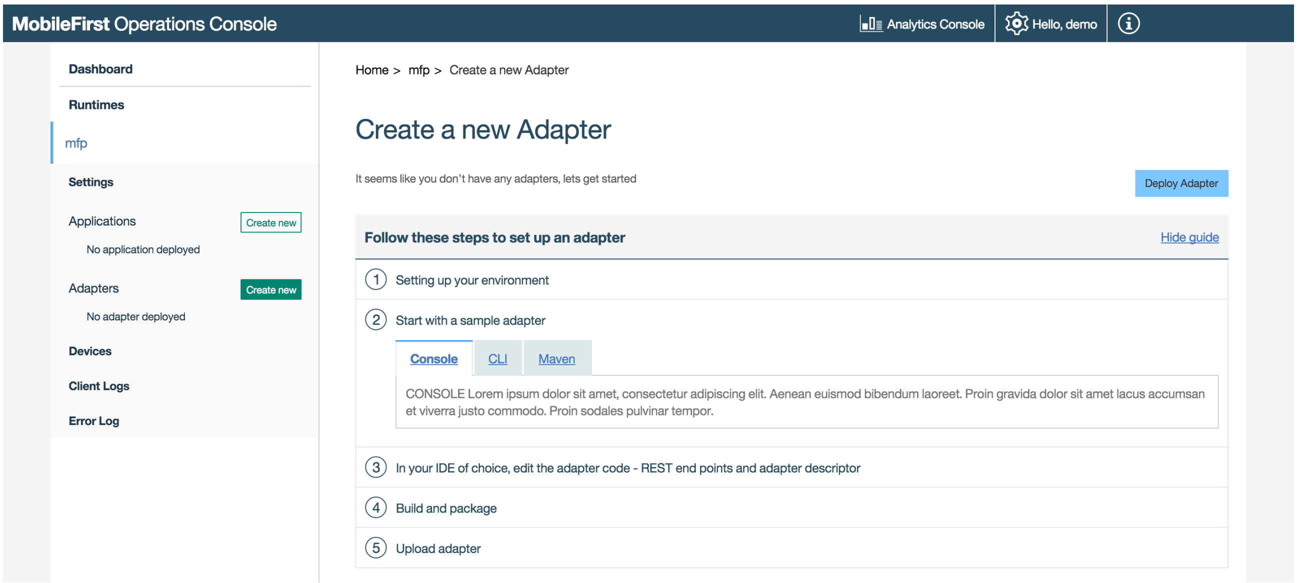
- In `protected void onCreate()`:

```
WLClient client = WLClient.createInstance(this);
URI adapterPath = null;
try {
    adapterPath = new URI("/adapters/javaAdapter/users/world");
} catch (URISyntaxException e) {
    e.printStackTrace();
}

WLResourceRequest request = new WLResourceRequest(adapterPath, WLResourceRequest.GET);
request.send(new WLResponseListener() {
    @Override
     public void onSuccess(WLResponse wlResponse) {
        Log.i("MobileFirst Quick Start", "Success: " + wlResponse.getResponseText());
    }

    @Override
    public void onFailure(WLFailResponse wlFailResponse) {
        Log.i("MobileFirst Quick Start", "Failure: " + wlFailResponse.getErrorMsg());
    }
});
```

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.



Home > mfp > Create a new Adapter

Create a new Adapter

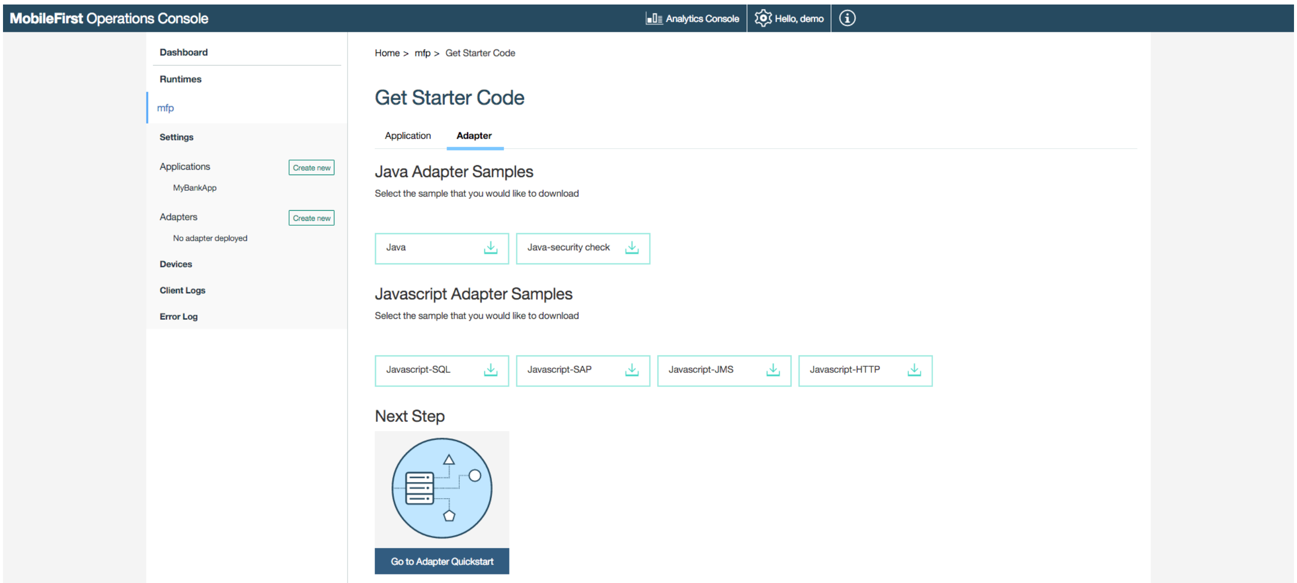
It seems like you don't have any adapters, lets get started

Follow these steps to set up an adapter

1. Setting up your environment
2. Start with a sample adapter

CONSOLE Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor.

3. In your IDE of choice, edit the adapter code - REST end points and adapter descriptor
4. Build and package
5. Upload adapter



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:

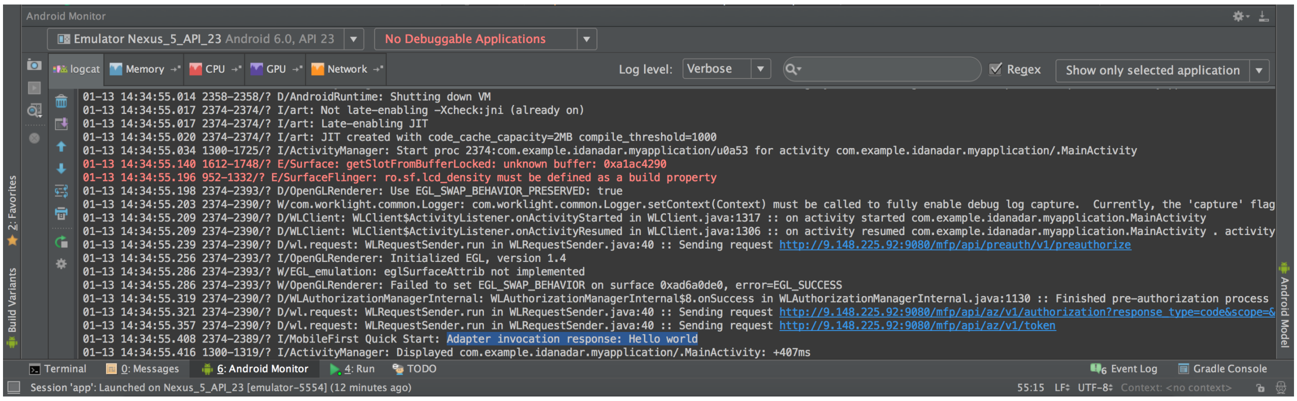
```
mfpdev adapter deploy
```

   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.
   The adapter response is then printed in Android Studio's **LogCat**.



```
01-13 14:34:55.014 2358-2358/? D/AndroidRuntime: Shutting down VM
01-13 14:34:55.017 2374-2374/? I/art: Not late-enabling -Xcheck:jni (already on)
01-13 14:34:55.017 2374-2374/? I/art: Late-enabling JIT
01-13 14:34:55.020 2374-2374/? I/art: JIT created with code_cache_capacity=2MB compile_threshold=1000
01-13 14:34:55.034 1300-1725/? I/ActivityManager: Start proc 2374:com.example.idanadar.myapplication/u0a53 for activity com.example.idanadar.myapplication/.MainActivity
01-13 14:34:55.140 1612-1748/? E/Surface: getSlotFromBufferLocked: unknown buffer: 0xa1ac4290
01-13 14:34:55.196 952-1332/? E/SurfaceFlinger: ro.sf.lcd_density must be defined as a build property
01-13 14:34:55.198 2374-2393/? D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true
01-13 14:34:55.203 2374-2390/? W/com.worklight.common.Logger: com.worklight.common.Logger.setContext(Context) must be called to fully enable debug log capture.  Currently, the 'capture' flag
01-13 14:34:55.209 2374-2390/? D/WLClient: WLClient$ActivityListener.onActivityStarted in WLClient.java:1317 :: on activity started com.example.idanadar.myapplication.MainActivity
01-13 14:34:55.209 2374-2390/? D/WLClient: WLClient$ActivityListener.onActivityResumed in WLClient.java:1306 :: on activity resumed com.example.idanadar.myapplication.MainActivity . activity
01-13 14:34:55.239 2374-2390/? D/wl.request: WLRequestSender.run in WLRequestSender.java:40 :: Sending request http://9.148.225.92:9080/mfp/api/preauth/v1/preauthorize
01-13 14:34:55.256 2374-2393/? I/OpenGLRenderer: Initialized EGL, version 1.4
01-13 14:34:55.286 2374-2393/? W/EGL_emulation: eglSurfaceAttrib not implemented
01-13 14:34:55.286 2374-2393/? W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xad6a0de0, error=EGL_SUCCESS
01-13 14:34:55.319 2374-2390/? D/WLAuthorizationManagerInternal: WLAuthorizationManagerInternal$8.onSuccess in WLAuthorizationManagerInternal.java:1130 :: Finished pre-authorization process
01-13 14:34:55.321 2374-2390/? D/wl.request: WLRequestSender.run in WLRequestSender.java:40 :: Sending request http://9.148.225.92:9080/mfp/api/az/v1/authorization?response_type=code&scope=&
01-13 14:34:55.357 2374-2390/? D/wl.request: WLRequestSender.run in WLRequestSender.java:40 :: Sending request http://9.148.225.92:9080/mfp/api/az/v1/token
01-13 14:34:55.408 2374-2389/? I/MobileFirst Quick Start: Adapter invocation response: Hello world
01-13 14:34:55.416 1300-1319/? I/ActivityManager: Displayed com.example.idanadar.myapplication/.MainActivity: +407ms
```

Session 'app': Launched on Nexus_5_API_23 [emulator-5554] (12 minutes ago)

Next steps

Learn more on using adapters in applications, and how to integrate additional services such as Push Notifications, using the MobileFirst security framework and more:

- Review the Server-side development tutorials (../../server-side-development/)

- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)