# Android Quick Start demonstration

## Overview

The purpose of this demonstration is to experience an end-to-end flow where the MobileFirst Platform Foundation SDK for Android is integrated into an Android project and used to retrieve data using a MobileFirst adapter.

To learn more about creating projects and applications, using adapters and lots more, visit the Native Android Development (../../android-tutorials/) landing page.

**Required installed:**

- MobileFirst Platform commandline tool (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads/))
- Android Studio

---

1. ### Create a MobileFirst project and adapter

   - **Create a new project and Android framework/server-side application entity**

     ```
     mfp create MyProject
     cd MyProject
     mfp add api MyAndroidFramework -e android
     ```

   - **Add a HTTP adapter to the project**

     ```
     mfp add adapter MyAdapter -t http
     ```

2. ### Deploy artifacts to the MobileFirst Server

   - **Start the MobileFirst Server and deploy the server-side application entity and adapter**

     ```
     mfp start
     # Wait until a browser window is opened, displaying the MobileFirst Console
     mfp deploy
     ```

3. ### Create an Android project

4. ### Add the MobileFirst Android SDK to the Android Studio project

   - From **project-folder-location > MyProject > apps > MyAndroidFramework**, copy the following files: `worklight-android.jar`, `uicandroid.jar`, `bcprov.jar` and `android-async-http.jar`
   - Open the **Project** view and navigate to the **app\libs** folder. Paste the copied files
   - Right-click on any of the added `.jar` files and select **Add as library** to add all libraries

- Create an **assets** folder under **src\main** and paste into it the `wlclient.properties` file
- Add the following permissions to the `AndroidManifest.xml` file:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.GET_TASKS" />
```

- Add the MobileFirst UI activity in the `AndroidManifest.xml` file:

```
<activity android:name="com.worklight.wlclient.ui.UIActivity" />
```

5. ## Implement MobileFirst adapter invocation

- **Main Activity class** Add imports:

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

Add the following to `onCreate`:

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

final WLClient client = WLClient.createInstance(this);

client.connect(new WLResponseListener() {

    @Override
    public void onSuccess(WLResponse wlResponse) {
        URI adapterPath = new URI("/adapters/MyAdapter/getFeed");
        WLResourceRequest request = new WLResourceRequest(adapterPath,WLResource
Request.GET);
        request.send(new MyInvokeListener());
    }

    @Override
    public void onFailure(WLFailResponse wlFailResponse) {
        Log.i("MFPMyProject","Failed connecting to the MobileFirst Server: " + wlFailRespons
e.getErrorMsg());
    }
});
```

- **`MyInvokeListener` class** Add a new `MyInvokeListener` class Add imports:

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

Paste the following:

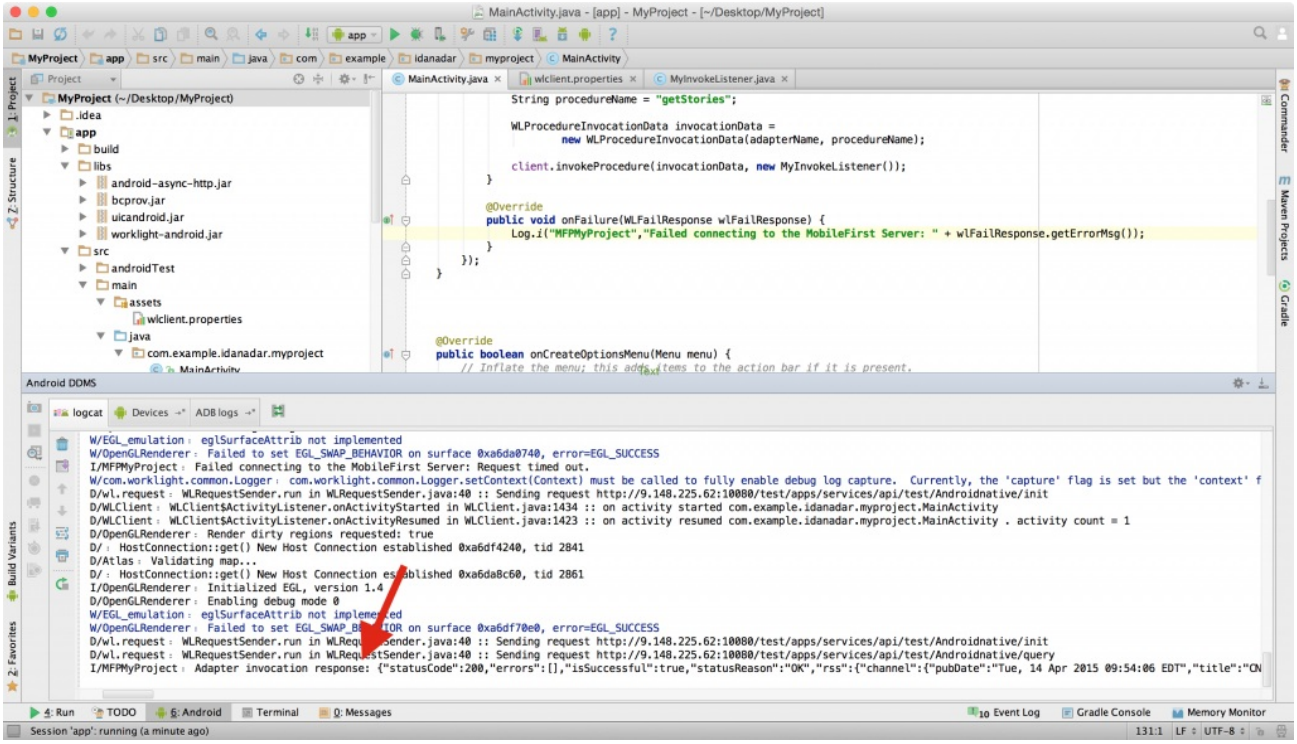
```
public class MyInvokeListener implements WLResponseListener {

    @Override
    public void onSuccess(WLResponse wlResponse) {
        Log.i("MFPMyProject","Adapter invocation response: " + wlResponse.getResponseJ
SON());
    }

    @Override
    public void onFailure(WLFailResponse wlFailResponse) {
        Log.i("MFPMyProject", "Adapter invocation response: " + wlFailResponse.getErrorMs
g());
    }
}
```

6. **Final configurations**

   - Supply the machine's IP address for the `host` property in `wlclient.properties`
   - Create an AVD

7. **Click Run**

   Review the LogCat view for the data retrieved by the adapter request.