

# 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 Conso
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:

```

1 |
2 | <uses-permission android:name="android.permission.INTERNET"/>
3 | <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
4 | <uses-permission android:name="android.permission.GET_TASKS" />

```

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

```

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

```

## 5. Implement MobileFirst adapter invocation

- **Main Activity class** Add imports:

```

1 | import com.worklight.wlclient.api.*;
2 | import android.util.Log;

```

Add the following to `onCreate`:

```

1 | super.onCreate(savedInstanceState);
2 | setContentView(R.layout.activity_main);
3 |
4 | final WLClient client = WLClient.createInstance(this);
5 |
6 | client.connect(new WLResponseListener() {
7 |
8 |     @Override
9 |     public void onSuccess(WLResponse wlResponse) {
10 |         URI adapterPath = new URI("/adapters/MyAdapter/getFeed");
11 |         WLResourceRequest request = new WLResourceRequest(adapterPath,WLResourceF
12 |             request.send(new MyInvokeListener());
13 |     }
14 |
15 |     @Override
16 |     public void onFailure(WLFailResponse wIFailResponse) {
17 |         Log.i("MFPMYProject","Failed connecting to the MobileFirst Server: " + wIFailResponse
18 |     }
19 | });

```

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

```

1 | import com.worklight.wlclient.api.*;
2 | import android.util.Log;

```

Paste the following:

```

1 | public class MyInvokeListener implements WLResponseListener {
2 |
3 |     @Override
4 |     public void onSuccess(WLResponse wlResponse) {
5 |         Log.i("MFPMMyProject", "Adapter invocation response: " + wlResponse.getResponseJSON());
6 |     }
7 |
8 |     @Override
9 |     public void onFailure(WLFailResponse wlFailResponse) {
10 |         Log.i("MFPMMyProject", "Adapter invocation response: " + wlFailResponse.getErrMsg());
11 |     }
12 | }

```

## 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.



