Creating your first native Android MobileFirst application

Overview

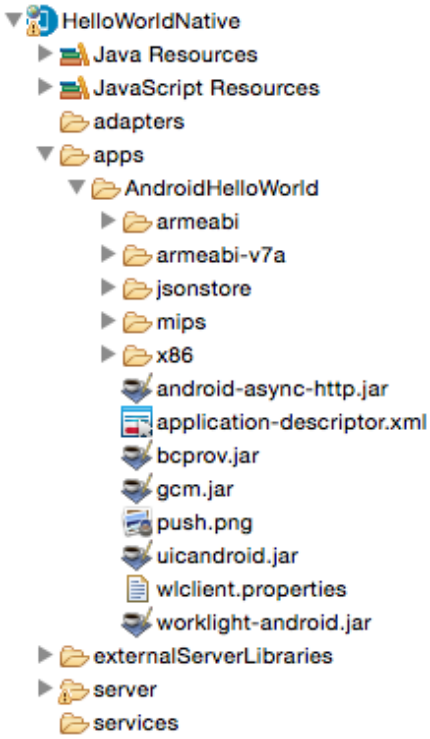
To serve a native Android application, MobileFirst Server must be aware of it. For this purpose, IBM MobileFirst Platform Foundation provides a Native API library, which contains a set of APIs and configuration files.

This tutorial explains how to generate the Android Native API and how to integrate it with a native Android application. These steps are necessary so that you can use it later on to achieve tasks such as connecting to MobileFirst Server, invoking adapter procedures, implementing authentication methods, and so on.

Prerequisite: Developers must be proficient with using Google's developer tools.

Creating a MobileFirst Native API

1. In MobileFirst Studio, create a MobileFirst project and add a MobileFirst Native API.
2. In the **New MobileFirst Native API** dialog, enter your application name and select **Android** for the **Environment** field.
3. Right-click the generated NativeAPI folder (located in `your-projects/apps/your-nativeapi-app-name`) and select **Run As > Deploy Native API**.
   `This action is required for MobileFirst Server to recognize the application when a request reaches the server.`
   The MobileFirst native API contains several components:
   - `wlclient.properties` contains the connectivity settings that a native Android application uses.
   - `worklight-android.jar` is the MobileFirst API library.
   - `gcm.jar` and `push.png` are for Google push notification services.
   - `JSONStore` folder is for optional JSONStore support in native applications.
   - `armabi`, `armabi-v7a`, `mips`, and `x86` folders are for optional Application Authenticity Protection in native applications.
   - As with any MobileFirst project, you create the server configuration by modifying the files that are in the **server\conf** folder.
   - You use the `application-descriptor.xml` file to define application metadata and to configure security settings that MobileFirst Server enforces.



wlclient.properties

The `wlclient.properties` file holds server configuration properties and is user-editable.

- `wlServerProtocol` – The communication protocol to MobileFirst Server, which can be either http or https.
- `wlServerHost` – The hostname of MobileFirst Server.
- `wlServerPort` – The port of MobileFirst Server.

- `wlServerContext` –The context root path of the application on the MobileFirst Server.
- `wlAppId` – The application ID as defined in the **application-descriptor.xml** file.
- `wlAppVersion` – The application version.
- `wlEnvironment` – The target environment of the native application.
- `wlUid` – The property used by MTWW to identify this as a MobileFirst application.
- `wlPlatformVersion` – The MobileFirst Studio version.
- `languagePreferences` – List of preferred locales.
- `GcmSenderId` – This property defines the GCM Sender ID to be used for push notifications. By default, this property is commented out.

Creating and configuring an Android native application

1. Create a native Android application or use an existing one.
2. Copy the `worklight-android.jar`, `uicandroid.jar`, `bcprov.jar`, and `android-async-http.jar` files from the WorklightAPI folder to the new native Android application, in the `/libs` directory.
3. Copy the file `wlclient.properties` from the MobileFirst native API folder to the new native Android application, in the `/assets` directory.
4. Add the following permissions to the `AndroidManifest.xml` file:

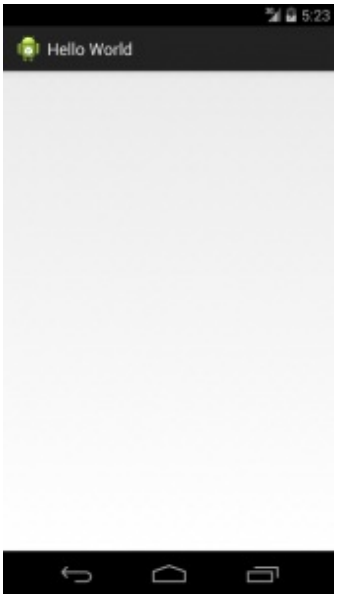
```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/
>
<uses-permission android:name="android.permission.GET_TASKS"/>
```

5. Add the MobileFirst UI activity:

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

> For more information, review the "Developing native applications for Android" user documentation topic

After the application is run in Eclipse, the final result is a native application that contains the MobileFirst API library. The provided Studio project contains a MobileFirst Studio project with the generated NativeAPI folder. The provided Native project contains an Android application already setup with the MobileFirst NativeAPI.



Sample application

Click to download (http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/HelloWorldNativeProject.zip) the Studio project.
Click to download (http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/AndroidHelloWorldNativeProject.zip) the Native project.