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.

This tutorial covers the following topics:

- Creating a MobileFirst native API
- The wlclient.properties file
- Creating and configuring an Android native application
- Tutorials to follow next

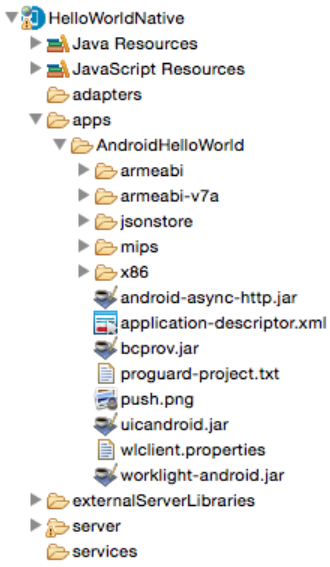
Creating a MobileFirst native API

CLI

1. In the terminal with the CLI (../../advanced-client-side-development/using-cli-create-build-manage-project-artifacts/) installed, create a new MobileFirst project using: `$ mfp create HelloWorldNative`.
2. Go to the newly created project directory: `$ cd HelloWorldNative/`.
3. Add a new Android native API using `$ mfp add api AndroidHelloWorld -e android`.
4. Build and deploy the application using `$ mfp bd`. *This action is required for MobileFirst Server to recognize the application if it attempts to connect.*

Studio

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**.
   Note: *This action is required for MobileFirst Server to recognize the application when a request reaches the server.*



The MobileFirst native API contains the following components:

- The `wlclient.properties` file contains the connectivity settings that a native Android application uses.
- The `worklight-android.jar` file is the MobileFirst API library.
- The `push.png` file is used for displaying an image for an incoming push notification.
- The `proguard-project.txt` file contains configuration settings for using Android ProGuard with Android Native API.
- The `JSONStore` folder is for optional JSONStore support in native applications.
- The `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.

The wlclient.properties file

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 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, the text workbench, to identify this as a MobileFirst application.
- `wlPlatformVersion` – The MobileFirst SDK version.
- `languagePreferences` – The list of preferred locales.
- `GcmSenderId` – This property defines the Google Cloud Message (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 generated NativeAPI folder to the new native Android application, in the `/libs` directory.
3. Copy the `wlclient.properties` file 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"/><br />
   <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
   ```

5. Add the MobileFirst UI activity:

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

> For more information, see the topic about developing native applications for Android, in the user documentation.

Tutorials to follow next

Now that your application contains the Native API library, you can follow the tutorials in the Native Android Development (../../android-tutorials/) section to learn more about authentication and security, server-side development, advanced client-side development, notifications and more.