General information when developing for Android

Overview

This tutorial presents general development information for the Android environment: as project structure, access to native capabilities through Apache Cordova, environment-specific resources and design resources.

Project structure

An Android app that is developed using IBM MobileFirst Platform comprises the following components:

- A Java main Android Activity that hosts an instance of a WebView component. The main activity uses a provided template, but can also be a pure Android native activity
- A set of Java and JavaScript™ libraries that provide access to various device features and capabilities
- Web application code that is provided by the developer. Such code is written in HTML, CSS, and JavaScript and runs in the WebView instance
- All application components, including the web code from the developer, that are packaged into a single Android project

The IBM MobileFirst Platform supports Android devices that run Android OS version 2.3.3 and 4.x.
If a source control management system (such as Rational Team Concert, Git, Subversion and so on), see the user documentation topic: "Integrating with source control system".

Designing for Android

Guidelines
It is useful to always consult the Google Design and Develop websites when developing applications:

- Design: http://developer.android.com/design/index.html (http://developer.android.com/design/index.html)
- Develop: http://developer.android.com/develop/index.html (http://developer.android.com/develop/index.html)

Resolutions
Various Android devices have different screen resolutions, for example:

- HTC Hero – 320 x 480
- Nexus One – 480 x 800
- Samsung Galaxy S4 – 1080 x 1920
- Google Nexus 4 – 1280 x 768

When displaying graphical elements on wider screens, Android automatically scales images and fonts to the appropriate ratio.
For example, on a 480-pixel wide screen, a 100-pixel wide image is scaled by 480/320 (= 1.5) to 150 pixels in width, and the text that is displayed with an 18-pixel font is displayed with a 24-pixel font.

Screen size support
By default, the `AndroidManifest.xml` file that is created by MobileFirst Studio defines that the application supports only normal screens:

```
<supports-screens
    android:smallScreens="false"
    android:normalScreens="true"
    android:largeScreens="false"
    android:resizeable="false"
    android:anyDensity="false"
    />
```

- To support tablets and newer Android devices, it must be declared that the application supports other types of screens
- Make sure to add appropriately sized resources (icons, images, styling) for those screen types and sizes
- For more information about supporting multiple screens, see the *Supporting multiple form factors using MobileFirst skins* tutorial
- For more information about supporting multiple screens, see http://developer.android.com (http://developer.android.com)

Look & Feel
By altering the `targetSdkVersion` value in the `AndroidManifest.xml` file, it is possible to control the look & feel of the native UI elements that are displayed in the application interface (dialogs, buttons, spinners, display resolution, and so on).

By default, a MobileFirst application is generated with `MinSdkVersion=10` (Android 2.3.3) and `targetSdkVersion=19`.
This means that a device uses the UI elements that fit best to its installed Android OS version. For example, in Android 2.x and Android 4.x, the look and feel of the Options Menu feature differs.

Environment-specific images

As seen in the previous slide, different Android devices come with different resolutions, so they require icons of different sizes.

If the Options Menu feature is to be implemented in an application, all the icons must be placed in the appropriate drawable folder, which is in `your-project-name\apps\your-app-name\android\nativeResources\res\drawable-*`.

- Designates a screen density (`LDPI, MDPI, HDPI, XDPI, XXDPI`).
  MobileFirst builder then copies these images to the generated project native folder.

Apache Cordova

The Apache Cordova framework can be used in Android applications to access the native elements of a device, such as contacts, geo-location services, media services, or the accelerometer.

For more information about Apache Cordova development, see the *Apache Cordova overview tutorial*.