# General information when developing for Android

## Overview

This tutorial presents general development information for the Android environment in the following topics:

- Project structure
- Designing for Android
- Environment-specific images
- Accessing native capabilities by using Apache Cordova
- Optimizing applications

## Project structure

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

- A Java main Android Activity which 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 which 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, which 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 you use a source control management system (such as Rational Team Concert, Git, Subversion and so on), see the topic about integrating with source control system, in the user documentation.

## Designing for Android

### Guidelines

When you develop applications, it is useful to always consult the Google Design (http://developer.android.com/design/index.html) and Develop (http://developer.android.com/develop/index.html) websites.

### 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 what 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 by using skins (../../advanced-client-side-development/supporting-multiple-form-factors-using-skins) tutorial.

### Look and Feel

By altering the `targetSdkVersion` value in the `AndroidManifest.xml` file, it is possible to control the look and 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 previously, 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).
  The MobileFirst builder then copies these images to the native folder of the generated project.

## Accessing native capabilities by using Apache Cordova

You can use the Apache Cordova framework in Android applications to access the native elements of a device, such as contacts, geolocation services, media services, or the accelerometer.

For more information about Apache Cordova development, see the Apache Cordova overview (../../adding-native-functionality/apache-cordova-overview) tutorial.

## Optimizing applications

When developing a mobile application, you can use minification and concatenation to reduce the size and number of files that are used within the application. This feature is available for the following environments: Android, iOS, Windows 8, Windows Phone 8, BlackBerry 10, Mobile Web and Desktop Browser.

> For more information about minification and concatenation, see the topic about optimizing MobileFirst applications, in the user documentation.