

How to Develop an App in Android Studio

Nicola Dunne
December 10, 2015

Contents

1. Introduction	1
2. Audience	1
3. Getting Ready	1
3.1 Tools	1
3.2 Check your computer for Java Development Kit (Version 7 or Higher).....	1
3.3 Download and Install Java Development Kit (Version 7 or Higher).....	2
3.4 Download and Install Android Studio	2
3.5 Connect your Android device to Android Studio	3
3.6 Install a Windows driver to detect your Android device	3
3.7 Check that your Android device is Connected to Android Studio.....	3
3.8 Setup the Android Emulator	3
3.9 Tools Checklist	4
4. Getting Started	4
4.1 App Description	4
4.2 Android Studio Dashboard	4
4.3 XML Syntax.....	6
4.4 Drawable Folder.....	6
5. Writing XML	6
5.1 Views and ViewGroups	6
5.2 Add Text	7
5.3 Add an Image.....	8
5.4 Style the Text	8
6. Finishing Up.....	9
6.1 Debugging the App.....	9
6.2 Testing the App.....	9

List of Illustrations

Figure 1: Next actions for Java Development Kit (Version 7 or Higher).....	5
Figure 2: Next actions for Android Device (Version 4.2 or later).....	5
Figure 3: The Android Studio dashboard design screen.	8
Figure 4: The Android Studio dashboard Text screen.....	8

1. Introduction

Android is a mobile operating system developed by Google for smartphones and tablets. An Android application, or ‘app’, is a computer program that runs on these devices. People use apps for many reasons like communicating with friends, learning new things, tracking exercise, and playing games. Useful, stylish apps are in demand and app development is a valuable skill. This document outlines the tools you need to create an app, describes the creation of the app, and explains what to do with your app when it is finished.

2. Audience

This guide is for readers who want to build their own app and have a beginner's interest in Computer Technology. Readers should have access to a Windows 10 Operating System and be familiar with tasks such as downloading files, installing software, and opening, closing, and saving files.

3. Getting Ready

This section lists the tools you need to create your Android app, and explains where to find them, how to download them, and how to install them on your computer. By the end of this section, you will be ready to develop your app.

3.1 Tools

- Java Development Kit (Version 7 or higher)
- Android Studio
- Android Device (Version 4.2 or later) **OR** Android Emulator
- USB cable
- Internet access

3.2 Check your computer for Java Development Kit (Version 7 or Higher)

Important! DO NOT download Android Studio until Java Development Kit (Version 7 or higher) is installed on your computer.

1. In the lower left corner of your computer screen, type Run in the search bar.
2. Under Best Match, double-click on Run (Desktop App) to open a dialog box.
3. In the dialog box, type cmd in the search bar and click OK to open the Terminal.
4. In the Terminal window, type java –version and press Enter. The Terminal window will display the version of Java on your computer or report that you do not have Java on your computer.

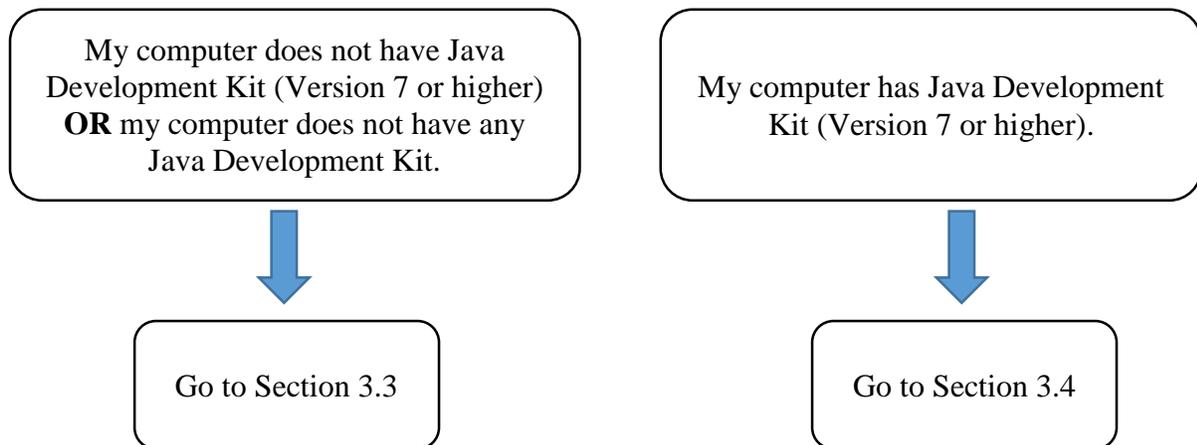


Figure 1: Next actions for Java Development Kit (Version 7 or Higher).

3.3 Download and Install Java Development Kit (Version 7 or Higher)

1. Open a search engine and go to www.oracle.com//technetwork/java/javase/downloads
2. On the left of the screen, double-click on the Java icon.
3. Navigate to the Java SE Development Kit 8u65.
4. Depending on your operating system, select the link for either Windows x86 or Windows x64.
5. When the download is complete, a Java Installation Wizard will open. Follow the onscreen instructions.
6. Verify that Java Development Kit (Version 7 or higher) is installed on your computer by repeating steps 1-4 of Section 3.2.

3.4 Download and Install Android Studio

1. Go to developer.android.com.
2. Click Download Android Studio for Windows.
3. When your download is complete, launch the .exe file.
4. To install Android Studio, follow the instructions of the on-screen Set-Up Wizard.

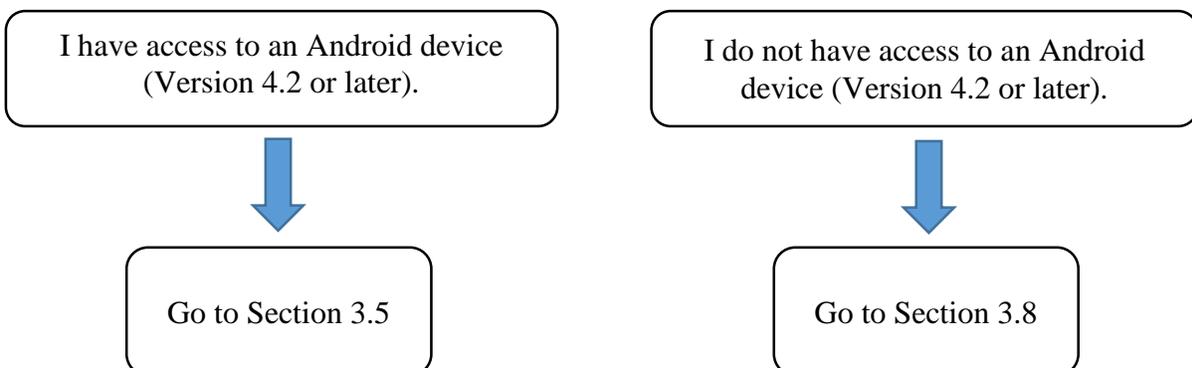


Figure 2: Next actions for Android Device (Version 4.2 or later).

3.5 Connect your Android device to Android Studio

1. On your Android device, go to Settings and scroll down to About Phone.
2. In About Phone, tap 'Build Number' 7 times to display a message that reads 'You are now a developer.'
3. Return to About Phone and select Developer Options.
4. Scroll down to USB Debugging and select enable.
5. Using a USB cable, connect your Android device to your computer.

3.6 Install a Windows driver to detect your Android device

1. Go to developer.android.com/tools/extras/oem-usb.html.
2. Scroll down to find your Android device and click on the corresponding Windows driver to download a zip folder.
3. Unzip the folder, go to This PC and select Manage.
4. In the left pane, select Device Manager and find Other Devices.
5. Expand Other Devices to see your Android device or the Android Debug Interface (ADB).
6. Right-click on your Android device or the ADB and select Update Driver Software to launch a Set-Up wizard.
7. To update your driver software, follow the on-screen instructions.

3.7 Check that your Android device is Connected to Android Studio

1. Launch Android Studio.
2. On the ribbon at the top of the screen, click the green play button and wait 10 seconds. The Choose a Running Device dialog box will open. Your device should be listed in the dialog box.
3. Select the button next to your device. Ensure that Launch Emulator is not selected.
4. Select phone and click OK. After five seconds, the 'Hello World' message will display on your Android device.

3.8 Setup the Android Emulator

1. Launch Android Studio.
2. To start the Android Emulator, click on the green run button on the ribbon at the top of the screen.
3. On the dialog box that opens, select Launch Emulator.
4. The dropdown menu next to Launch Emulator displays one device called Nexus 5 API21x 86. Select this device and click OK.
5. After 10 seconds, a black phone will appear on the screen of your Android device. Wait for the Android Loading screen to finish, then click and drag the home button upwards in

a swiping motion to open the Android Emulator. The Android Studio app automatically appears on the Android Emulator.

3.9 Tools Checklist

Before you proceed to the next section, check that your tools are ready for use.

Tool	Action	Check
Java Development Kit (Version 7 or Higher)	Installed	
Android Studio	Installed	
Android Device AND Windows Driver	Connected AND Detecting Android Device	
OR	OR	
Android Emulator	Displays Android Studio App	

4. Getting Started

This section includes overviews of the app we will build, the Android Studio dashboard, sample XML code, and the drawable folder.

4.1 App Description

The app will display information about local services in Exeter, New Hampshire. Specifically, the app will consist of a background image and the following text:

- Exeter Police Department 603-772-1212
- Exeter Town Office 603-778-0591
- Exeter Public Library 603-772-3101

4.2 Android Studio Dashboard

The dashboard opens after you create a new project in Android Studio. The dashboard has two screens – Design and Text, and you can toggle between both screens as you create your app. The Design screen displays the current version of your app in real-time. The Text screen displays the files that contain your code you write.

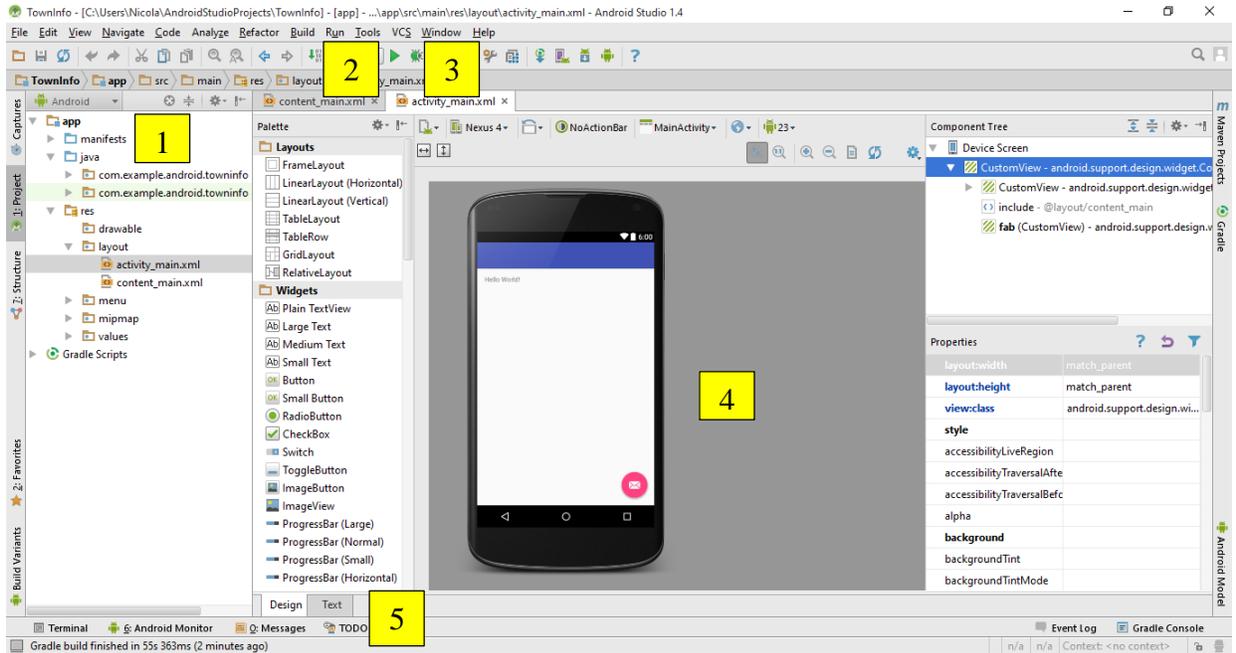


Figure 3: The Android Studio dashboard design screen.

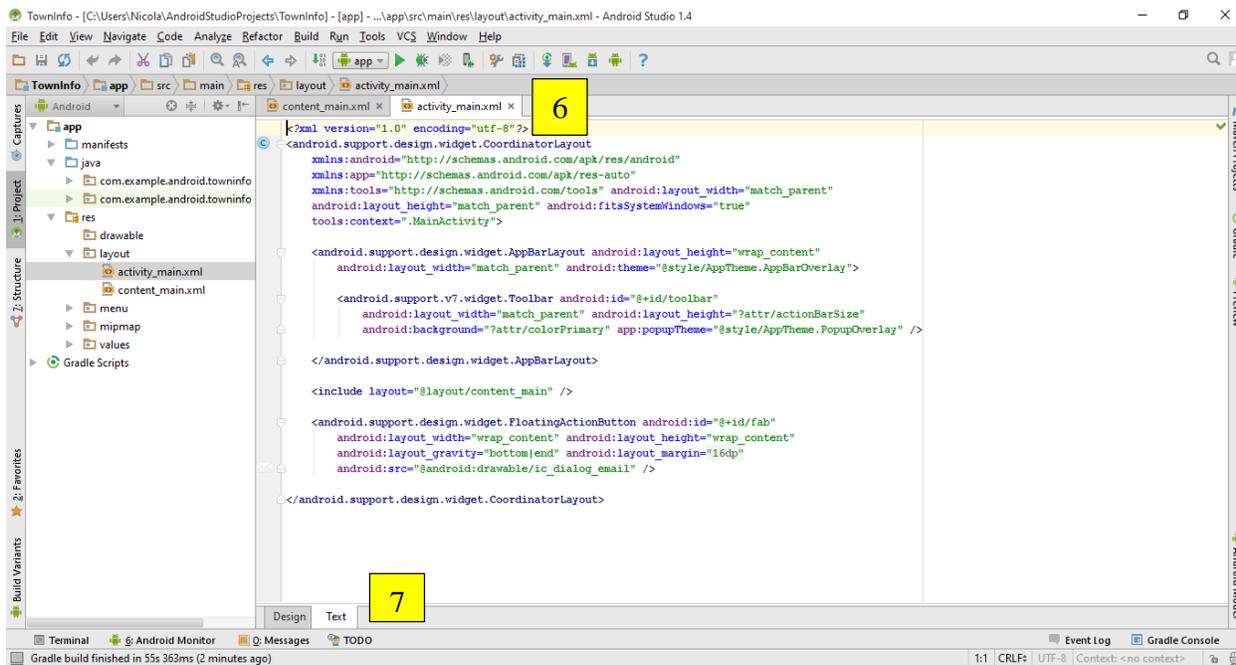


Figure 4: The Android Studio dashboard Text screen.

To build the Exeter Town app, use the following Android Studio features:

Number	Feature	Use
1.	Folders	Expandable folders containing automatically generated files

2.	Run icon	Click to run Android Emulator
3.	Debug icon	Click to debug the app
4.	Android Emulator	View design changes to the app
5.	Design icon	Toggle to the Design screen
6.	XML file	Location of XML code
7.	Text icon	Toggle to the Text screen

4.3 XML Syntax

To create the Exeter Town app, use the programming language XML. Like any language, XML has its own syntax or rules that define valid and invalid XML. This means that if the syntax is incorrect, the XML code is invalid and will not run on your Android device or Android Emulator. To build the Exeter Town app, the XML syntax will consist of an opening tag, a view name, attributes, and a closing tag.

<pre> 1 <TextView 2 android:text="Hello World." android:background="@android:color/darker_gray" android:layout_width="150dp" 3 android:layout_height="75dp" /> 4 </pre>	<ol style="list-style-type: none"> 1. Opening Tag 2. View Name 3. Attributes 4. Closing Tag
---	---

Figure 5: Sample XML syntax

4.4 Drawable Folder

Android Studio uses a folder called drawable to store images for apps. To add an image to an app, you must first save the image to drawable. To save an image to drawable, use the file name @drawable/image_name. For example, @drawable/Exeter_town.

5. Writing XML

In this section, you will build a static app in Android Studio using Extensible Markup Language (XML). We will use Views to add text and an image to the app, change the layout and positioning of the app elements, and decide on styles to complete the design.

5.1 Views and ViewGroups

A view is an invisible rectangle on an app that contains content. Text, images, and buttons are common views and the layout of an app consists of these elements. The two views that you will use to create the app are TextView and ImageView.

To add more than one View to an app, use a ViewGroup. The Linear Layout ViewGroup displays Views in a vertical or horizontal line on our device screens.

5.2 Add Text

1. Create a new project in Android Studio.
2. In the Activity.xml file, use an opening tag, a TextView, an attribute, and a closing tag to add the first line of text to the app.

```
<TextView
    android:text="Exeter Police Department. 603-772-
    1212"
    android:background="@android:color/darker_gray"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

3. Enclose the previous code in LinearLayout ViewGroup.
4. Inside the LinearLayout ViewGroup, add two more TextViews. These two TextViews should contain the remaining information about Exeter Town's services.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    Android: layout_width="match_parent"
    Android: layout_height="match_parent"
    Android: orientation="vertical" >

    <TextView
        android:text="Exeter Police Department. 603-772-1212"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="16dp"
        android:paddingTop="16dp" />

    <TextView
        android:text="Exeter Town Office. 603-778-0591"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="16dp"
        android:paddingTop="16dp" />

    <TextView
        android:text="Exeter Public Library. 603-772-3101"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="16dp"
        android:paddingTop="16dp" />
```

```
</LinearLayout>
```

5.3 Add an Image

1. To add an image to the app, use the ImageView and the drawable file name. Add the ImageView directly underneath the opening tag of the LinearLayout ViewGroup.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    Android: layout_width="match_parent"
    Android: layout_height="match_parent"
    Android: orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="centerCrop"
        android:src="@drawable/ExeterTown" />

    <TextView
        android:text="Exeter Police Department. 603-772-1212"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="16dp"
        android:paddingTop="16dp" />

    .....

</LinearLayout>
```

5.4 Style the Text

To change style of the text, add the fontFamily, fontColor, and fontSize attributes to each TextView.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    Android: layout_width="match_parent"
    Android: layout_height="match_parent"
    Android: orientation="vertical" >

    <ImageView
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:scaleType="centerCrop"
android:src="@drawable/ExeterTown" />
```

```
<TextView
    android:text="Exeter Police Department. 603-772-1212"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:fontFamily="sans-serif-light"
    android:textColor="@android:color/white"
    android:textSize="36sp" />
.....
</LinearLayout>
```

6. Finishing Up

6.1 Debugging the App

When an app is finished, it is important to check the code for syntax errors in a process known as ‘debugging’.

1. Open your project in Android Studio.
2. On the toolbar at the top of the screen, click the debug icon.
3. In the Choose Device window, click your Android device or the Android Emulator.
4. To begin debugging your app, click OK.

When the debugging is complete, Android Studio will highlight any code errors in red and suggest solutions. When there are no errors in the code, your app will be ready for testing.

6.2 Testing the App

Testing is an important part of app development because it helps to improve the quality of apps, increases the quality of the user’s experience, and reduces the time required to fix defective apps.

Three tools that you can use to test the Exeter Town app are:

- Android Testing Support Library
- Monkey
- monkeyrunner

Glossary

A	Android Emulator	Virtual simulator for Android devices in Android Studio
D	Drawable	Folder in Android Studio that stores images for Android apps.
I	ImageView	Container that holds an image in an app.
T	TextView	Container that holds text inside an app.
V	View, ViewGroup	Container that holds one item inside an app, container that holds multiple Views inside an app.
X	XML	Programming language used to describe data.