How To Create An Android App With Android Studio

by **TheZachBales** on March 25, 2015

Table of Contents

Intro: How To Create An Android App With Android Studio

This tutorial will teach you the basics of how to build an Android app using the Android Studio development environment. As Android devices become increasingly more common, demand for new apps will only increase. Android Studio is an easy to use (and free) development environment to learn on. It's best if one has a working knowledge of the Java programming language for this tutorial because it is the language used by Android. There won't be much code used in this tutorial, so I will assume that you know enough Java to understand or are willing to look up what you don't know. This will take roughly 30-60 minutes, depending on how quickly you are able to download and install Android Studio. After using this tutorial to create your first Android app, you'll be well on your way to a fun new hobby or possibly even a promising career in mobile development.

Step 1: Install Android Studio

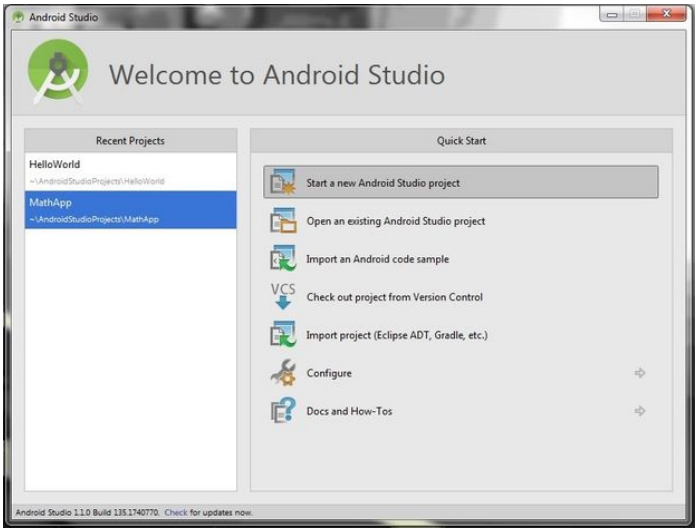
1. Go to http://developer.android.com/sdk/index.html to download Android Studio.
2. Use the installer to install Android Studio following its instructions.

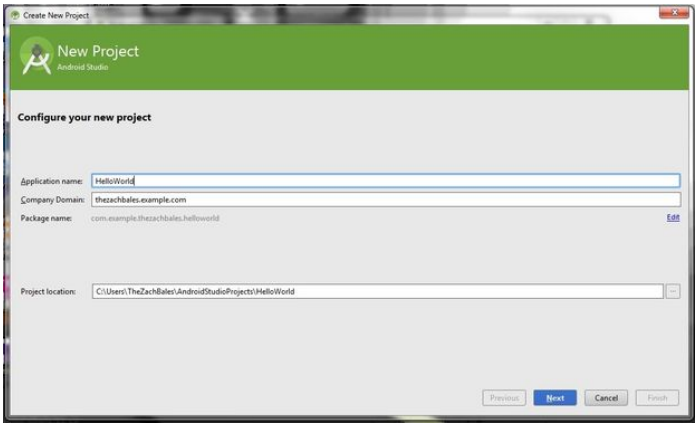
Image Notes
1. Download website.

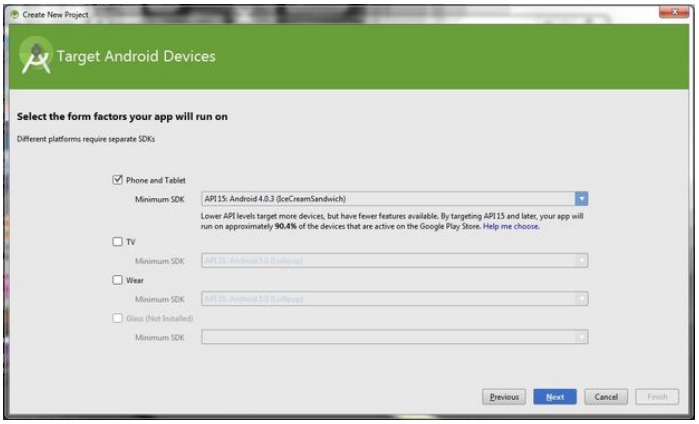
Step 2: Open a New Project

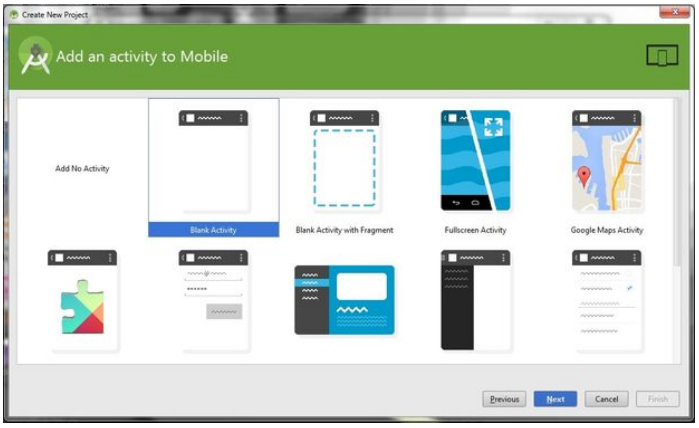
1. Open Android Studio.
2. Under the "Quick Start" menu, select "Start a new Android Studio project."
3. On the "Create New Project" window that opens, name your project "HelloWorld".
4. If you choose to, set the company name as desired*.
5. Note where the project file location is and change it if desired.
6. Click "Next."
7. Make sure on that "Phone and Tablet" is the only box that is checked.
8. If you are planning to test the app on your phone, make sure the minimum SDK is below your phone's operating system level.
9. Click "Next."
10. Select "Blank Activity."
11. Click "Next."
12. Leave all of the Activity name fields as they are.
13. Click "Finish."

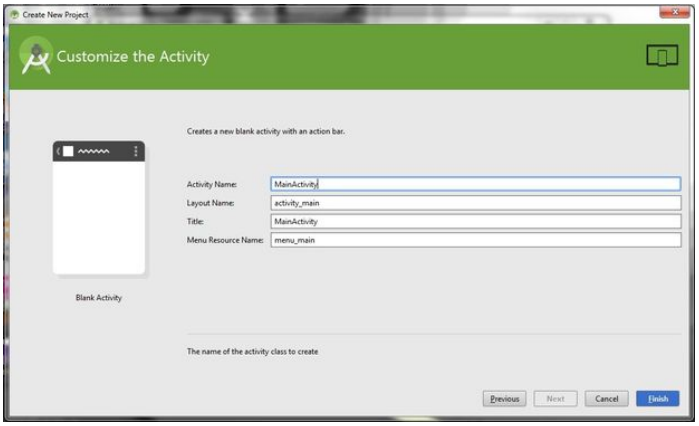
***Note:** It is typical naming convention in Android projects to set the company name as some form of "example.name.here.com".





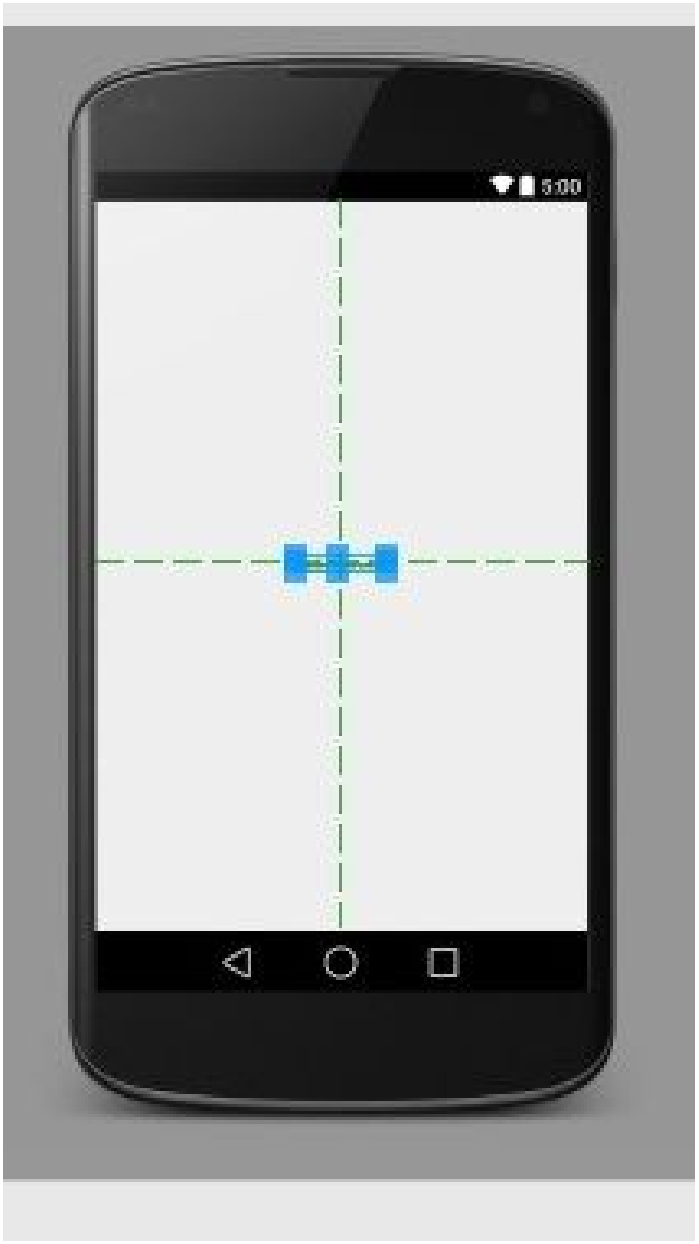






Step 3: Edit the Welcome Message in the Main Activity

1. Navigate to the activity_main.xml tab if it is not already open.
2. Make sure that the Design tab is open on the activity_main.xml display.
3. Click and drag the "Hello, world!" from the upper left corner of the phone display to the center of the screen.
4. In the project file system on the left side of the window, open the values folder.
5. In the values folder, double-click the strings.xml file.
6. In this file, find the line "Hello world!".
7. After the "Hello world!" message, add "Welcome to my app!"
8. Navigate back to the activity_main.xml tab.
9. Make sure that your centered text now reads "Hello world! Welcome to my app!"

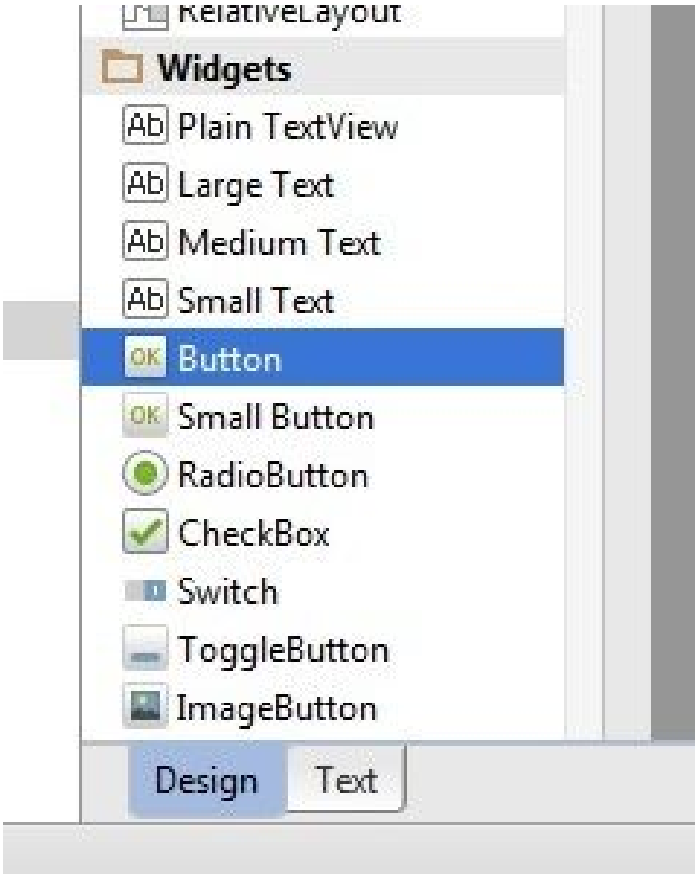



```
Edit translations for all locales in the translations editor.
<resources>
    <string name="app_name">HelloWorld</string>

    <string name="hello_world">Hello world! Welcome to my app!</string>
    <string name="action_settings">Settings</string>
</resources>
```

Step 4: Add a Button to the Main Activity

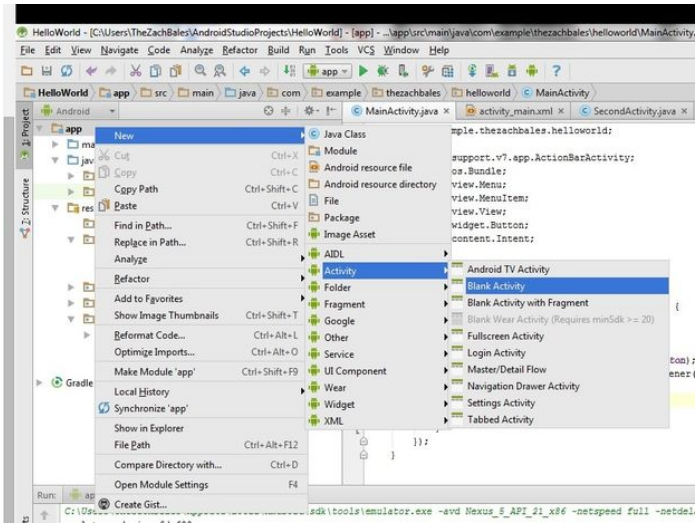
1. Navigate to the Design tab of the activity_main.xml display.
2. In the Palette menu to the left of the phone display, find Button (under the heading Widgets).
3. Click and drag Button to be centered underneath your welcome message.
4. Make sure your button is still selected.
5. In the Properties menu (on the right side of the window), scroll down to find the field for "text."
6. Change the text from "New Button" to "Next Page."

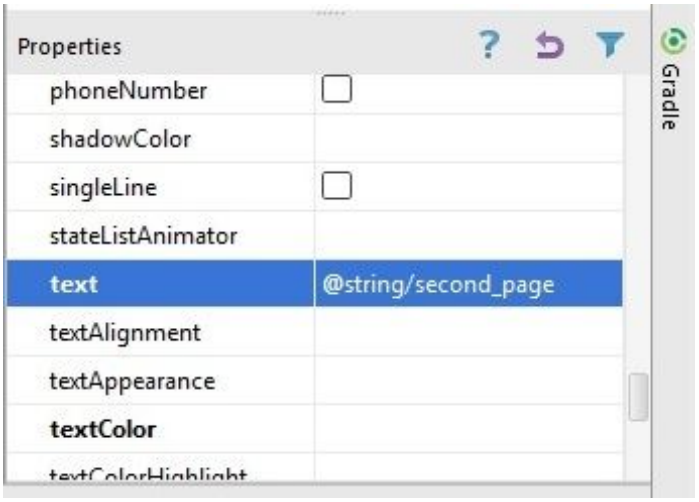


exus_5_API_21_x86 -netspeed full -netdelay

Step 5: Create a Second Activity

1. At the top of the project's file system tree, right click on "app."
2. Navigate through to New > Activity > Blank Activity.
3. Change the name of this activity to "SecondActivity".
4. Click "Finish."
5. Make sure you are in the Design view of activity_second.xml.
6. Drag the text box in the upper left of the phone display down to the center as you did on the Main Activity.
7. With the text box still selected, find the "id" field in the Properties menu on the right, and set it to "text2".
8. Open strings.xml again.
9. Add a new line under "Hello world! Welcome to my app!" that reads "Welcome to the second page!".
10. Navigate back to activity_second.xml.
11. Select the text box again.
12. In the Properties pane, set the "text" field to "@string/second_page".
13. Make sure that the text box now reads "Welcome to the second page!" and is in the center of the screen in the phone display.





Step 6: Write the Button's "onClick" Method

1. Select the MainActivity.java tab along the top of the work environment.

2. Add the following lines of code at the end of the onCreate method:

```
Button button = (Button) findViewById(R.id.button);
button.setOnClickListener(new View.onClickListener() {
@Override
public void onClick(View v) {
goToSecondActivity();
}
});
```

3. Add the following method to the bottom of the MainActivity class:

```
private void goToSecondActivity() {
Intent intent = new Intent(this, SecondActivity.class);
startActivity(intent);
}
```

4. Click the + next to import at the third line of MainActivity.java to expand the import statements.

5. Add the following to the end of the import statements if they are not already there:

```
import android.content.Intent;
import android.view.View;
```

import android.widget.TextView;

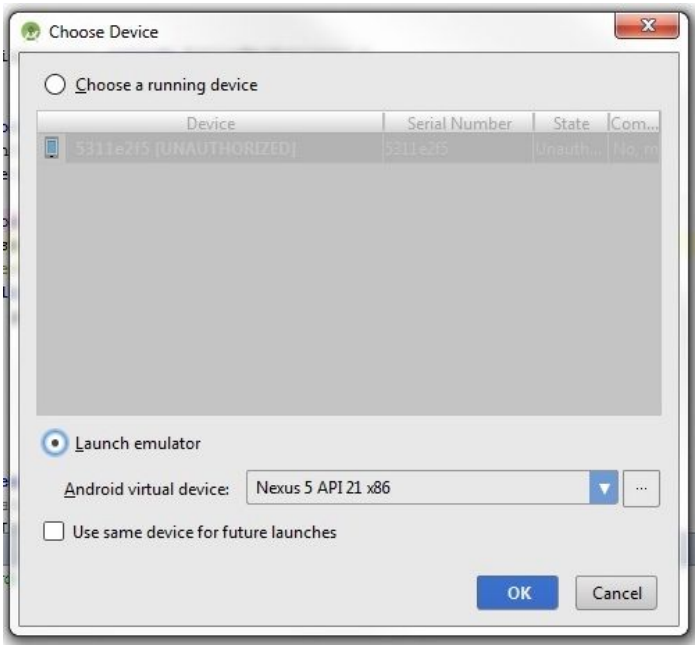
```
Button button = (Button) findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        goToSecondActivity();
    }
});
```

```
private void goToSecondActivity()
{
    Intent intent = new Intent(this, SecondActivity.class);
    startActivity(intent);
}
```

```
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.content.Intent;
```

Step 7: Test the Application

1. Click the green play symbol from the toolbar at the top of the Android Studio window.
2. When the "Choose Device" dialog apperas (this may take a few moments), select the "Lauch emulator" option.
3. Click OK.
4. When the emulator opens (this too could take awhile), the app will automatically launch the app upon the virtual phone being unlocked.
5. Make sure that all of your text displays correctly and that the button takes you to the next page.



Step 8: Up, Up, and Away!

Congrats! You've now completed your first Android application with some basic functionality. Your finished app should have a page greeting the user and a button that takes the user to a second page.

From here you have the cursory knowledge you need to go on to learn all there is to know about Android application development.

Related Instructables

Your first Android Application by Computothought

How To Setup Eclipse for Android App Development by im292fresh

How to: Android Screen Shots by TheSmartLemon

GamePad using Android mobile sensors and Arduino by ashraf nabil

Use android cellphone as webcam free by ViperSniper

How To Use Miracast With an Android Device by Chinavasion

Comments