

How to update an app in "Java for Android Developers For Dummies, 2nd Edition" May, 2024

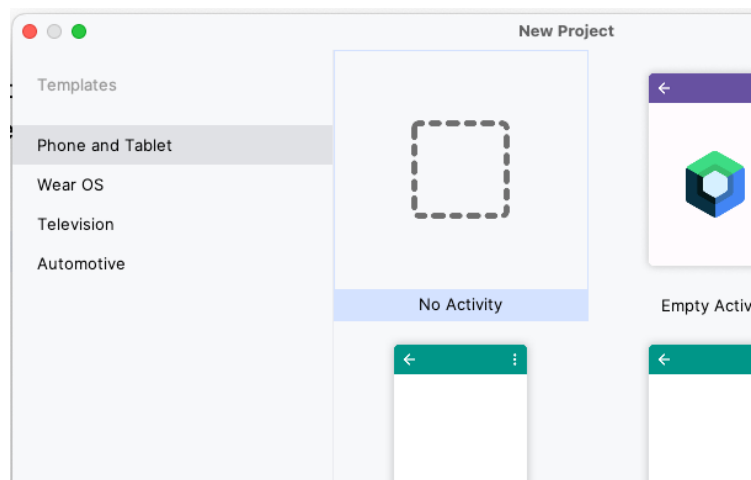
I've placed a file named [02_01_Java4Android_May2024.zip](#) on my book's web page. You can download that file, open it with the latest version of Android Studio, and run it.

These instructions tell you how to create that `02_01_Java4Android_May2024` project if you start with the old `02_01` project from my book.

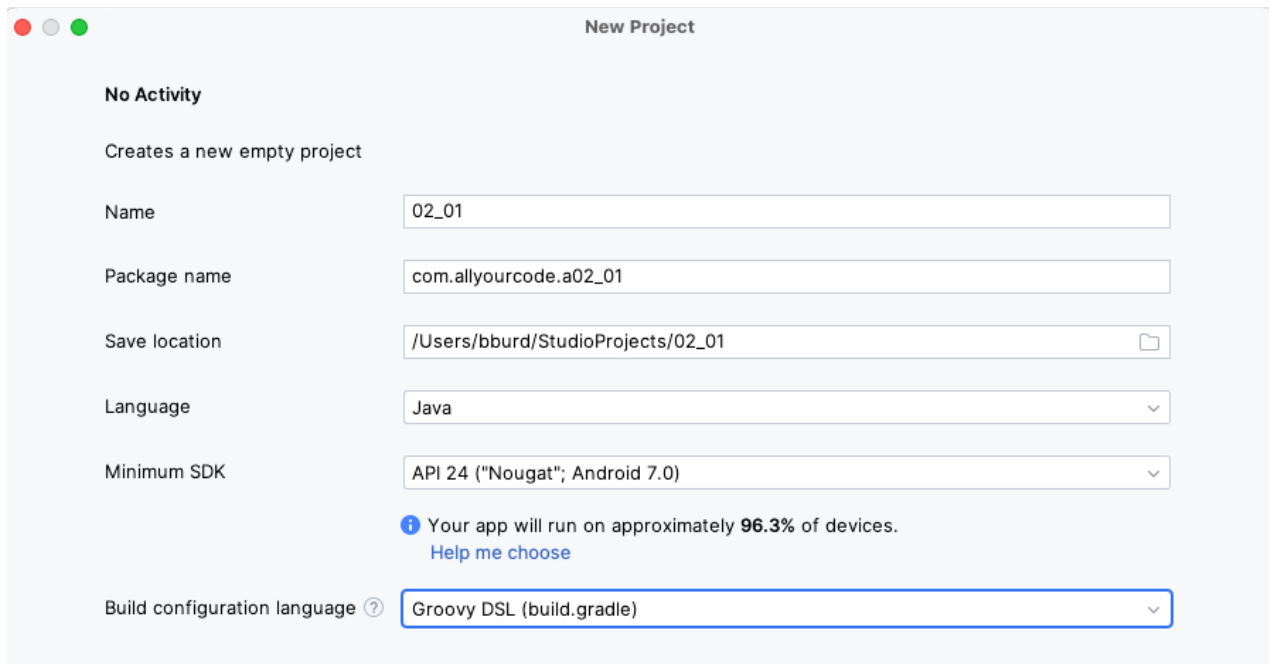
In these instructions, I'll refer to two copies of the `02_01` project:

- "Old project" is the project that you get when you download the original code from my book's web page.
(https://users.drew.edu/bburd/java4android/Java4Android_Projects.zip)
- "New project" is the project that you're creating in these instructions.

1. Launch an up-to-date version of Android Studio.
2. In Android Studio, create a new project with No Activity.



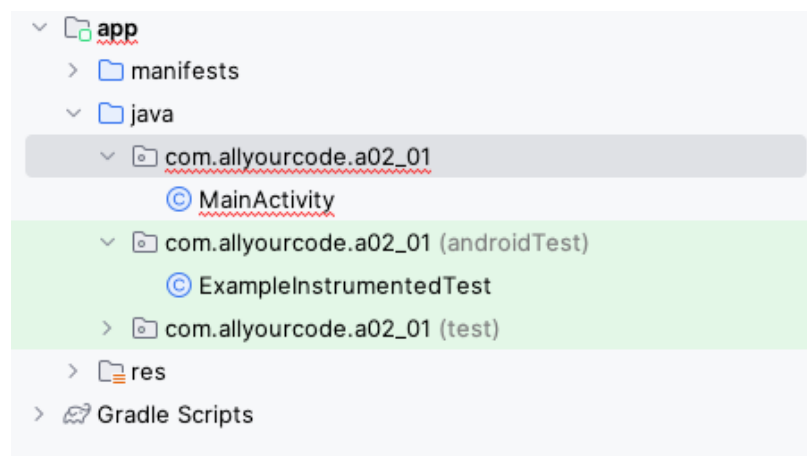
3. Give your new project the same package name (`com.allyourcode.a02_01`) as the package name in the old project.



In the New Project dialog box's Language field, select Java (not Kotlin). In the Build Configuration Language field, select Groovy DSL (build.gradle), not Kotlin DSL (build.gradle.ktx).

4. Copy the

`app/src/main/java/com/allyourcode/a02_01/MainActivity.java` file from the old project into the `app/java/com.allyourcode.a02_01` branch of the new project.



The `MainActivity.java` file that you pasted into your new project has some errors. We'll fix those as we go along.

5. In the `MainActivity.java` file, replace the `import android.support.v7` line with this line:

```
import androidx.appcompat.app.AppCompatActivity;
```

```
3 //import android.support.v7.app.AppCompatActivity;  
4 import androidx.appcompat.app.AppCompatActivity;  
5 import android.os.Bundle;
```

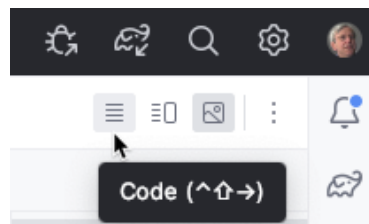
6. Copy the `app/src/main/res/layout` folder from the old project to the new project's `app/src/main/res` folder.

Your new project now has an `app/src/main/res/layout/activity_main.xml` file.

7. In the tree of folders and files along the left side of the Android Studio window, double-click the `activity_main.xml` file.

This opens up a visual depiction of the app's layout.

8. Look for a little icon with horizontal lines in the upper right corner of the Android Studio window. Click this icon to see the `activity_main.xml` file's code.



9. In the `activity_main.xml` file, change both occurrences of `android.support.constraint.ConstraintLayout` to `androidx.constraintlayout.widget.ConstraintLayout`.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/ar
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.allyourcode.a02_01.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="@+id/ac1
        app:layout_constraintLeft_toLeftOf="@+id/activi1
        app:layout_constraintRight_toRightOf="@+id/activi
        app:layout_constraintTop_toTopOf="@+id/activity.

</androidx.constraintlayout.widget.ConstraintLayout>

```

10. Replace the `app/src/main/AndroidManifest.xml` with the old project's `app/src/main/AndroidManifest.xml` file.

Tip: If you look at the Android View in the tree of folders and files along the left side of the Android Studio window, you'll see the tree's `app/manifests/AndroidManifest.xml` branch. To update this file, you may have to double-click the file, delete the file's text in the Editor, and paste the old file's text into the Editor.

11. In the same `AndroidManifest.xml` file, add `android:exported="true"` to the activity start tag.

```

android:theme="@style/AppTheme"
<activity android:name=".MainActivity" android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>

```

12. Copy the `app/src/main/res/values/styles.xml` file from the old project to the `app/src/main/res/values` folder in the new project.

13. Look at the old project's `app/src/main/res/values/colors.xml` file. Copy the color elements from that file and add them to the new project's `app/src/main/res/values/colors.xml` file.

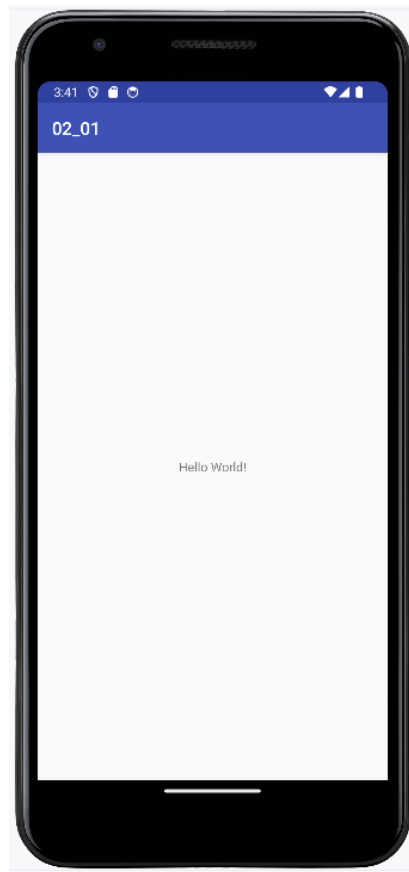
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#3F51B5</color>
  <color name="colorPrimaryDark">#303F9F</color>
  <color name="colorAccent">#FF4081</color>

  <color name="purple_200">#FFBB86FC</color>
  <color name="purple_500">#9C27B0</color>

```

Now you're finished doing the update!

14. Run the new project.



All that work to run a Hello World program! These instructions should work (more or less) for many of the projects in the book's 2nd edition. If you run into any problems, feel free to send email to me. The address is java4android@allmycode.com.