Creating the first Android app

Contents

- Android Studio
- Running apps on virtual and physical devices
- Set up a project
- Creating "Hello World" app in Android Studio
- Basic app development workflow with Android Studio
- Exercises

These slides are partially based on the material that Google provides for the course
Android Developer Fundamentals

https://developer.android.com/courses/fundamentals-training/overview-v2

Installation Overview

- Mac, Windows, or Linux
- Download and install Android Studio from https://developer.android.com/studio/
- https://developer.android.com/studio/install.html
- Codelab here: https://codelabs.developers.google.com/codelabs/android-training-hello-world/

Installation Overview

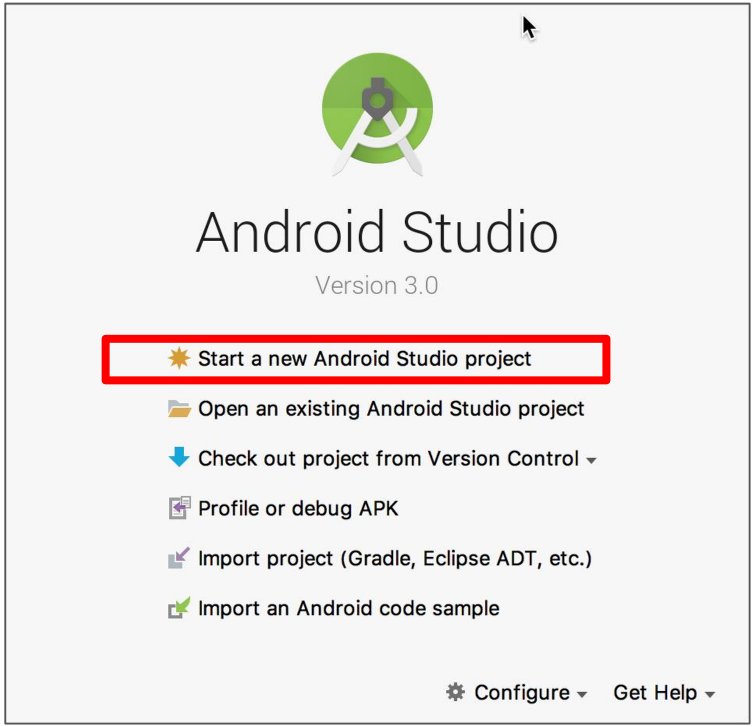
- Mac, Windows, or Linux
- Download and install Android Studio from https://developer.android.com/studio/
- https://developer.android.com/studio/install.html
- Codelab here: https://codelabs.developers.google.com/codelabs/android-training-hello-world/

Try to complete it!

What is Android Studio?

- Android integrated development environment (IDE)
- Project and Activity templates
- Layout editor
- Testing tools
- Gradle-based build
- Log console and debugger
- Emulators

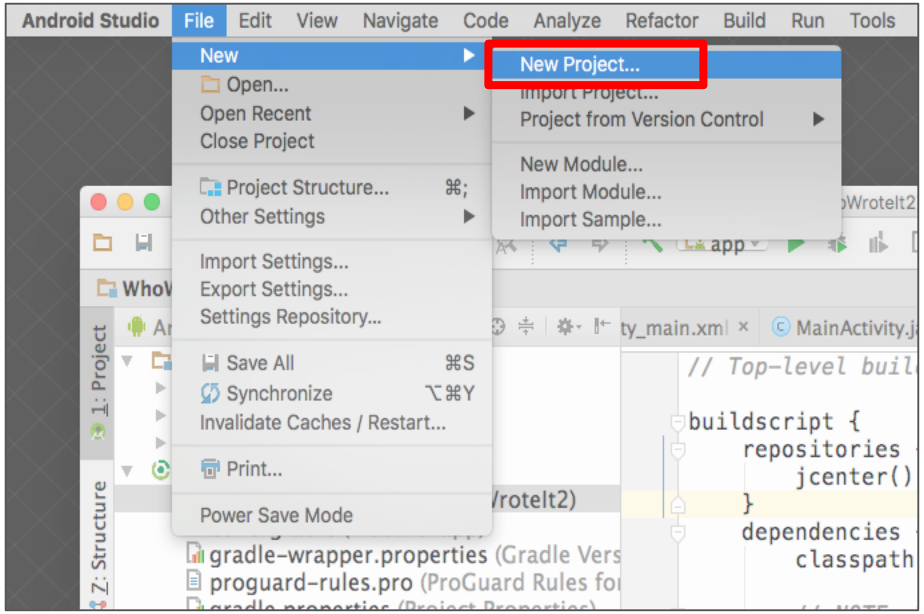
Start Android Studio


Android Studio

Version 3.0

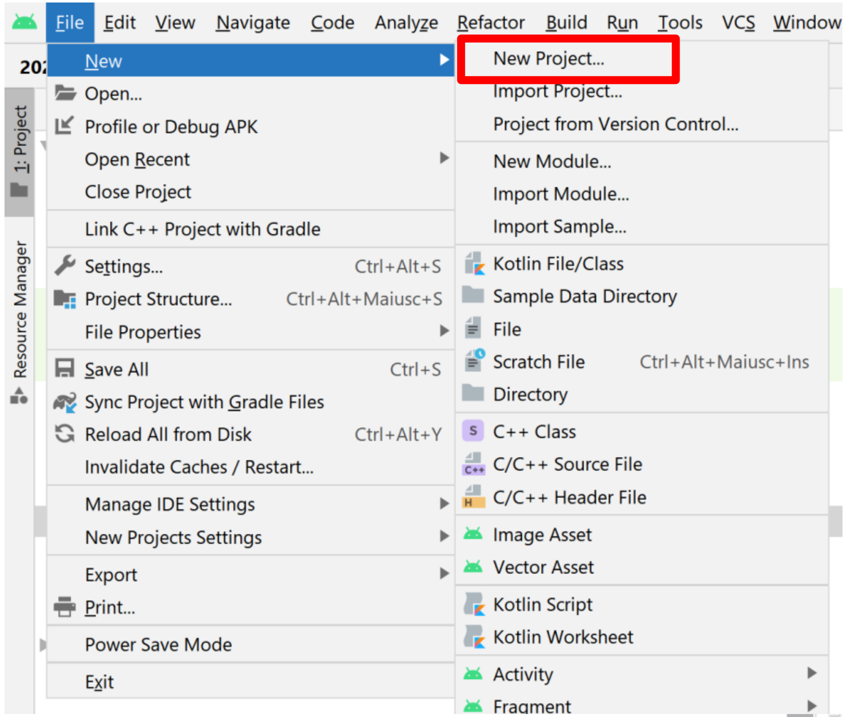
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

Create a project inside Android Studio



```
// Top-level buil
buildscript {
    repositories
        jcenter()
    }
    dependencies
        classpath
```

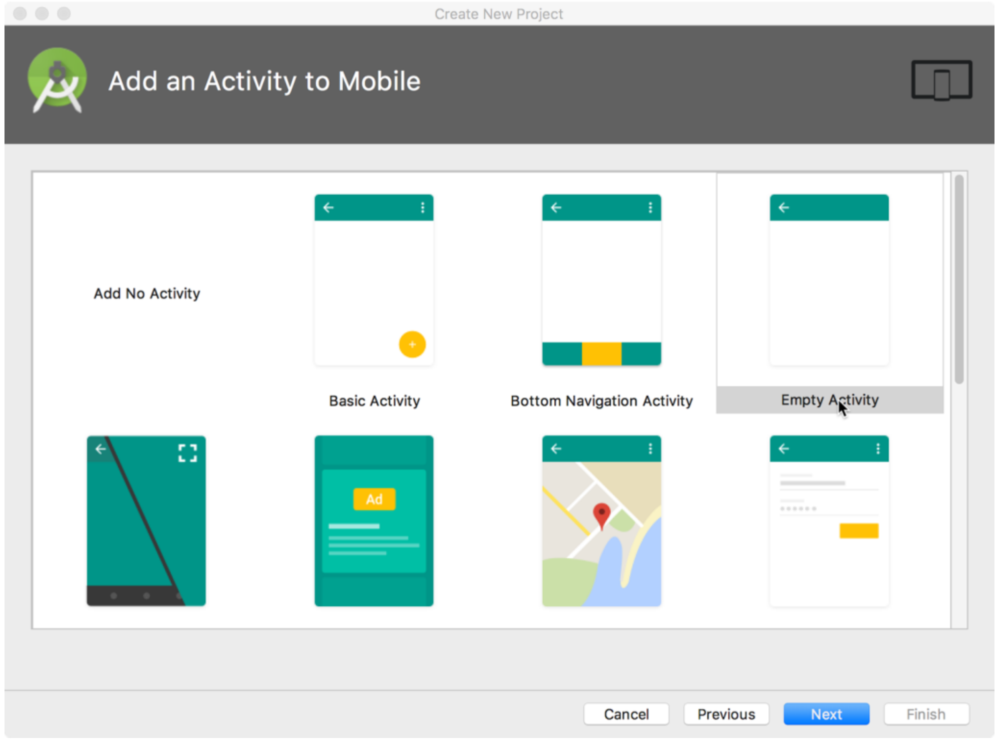
gradle-wrapper.properties (Gradle Vers
proguard-rules.pro (ProGuard Rules for



Pick activity template

Choose templates for **common activities**, such as maps or navigation drawers

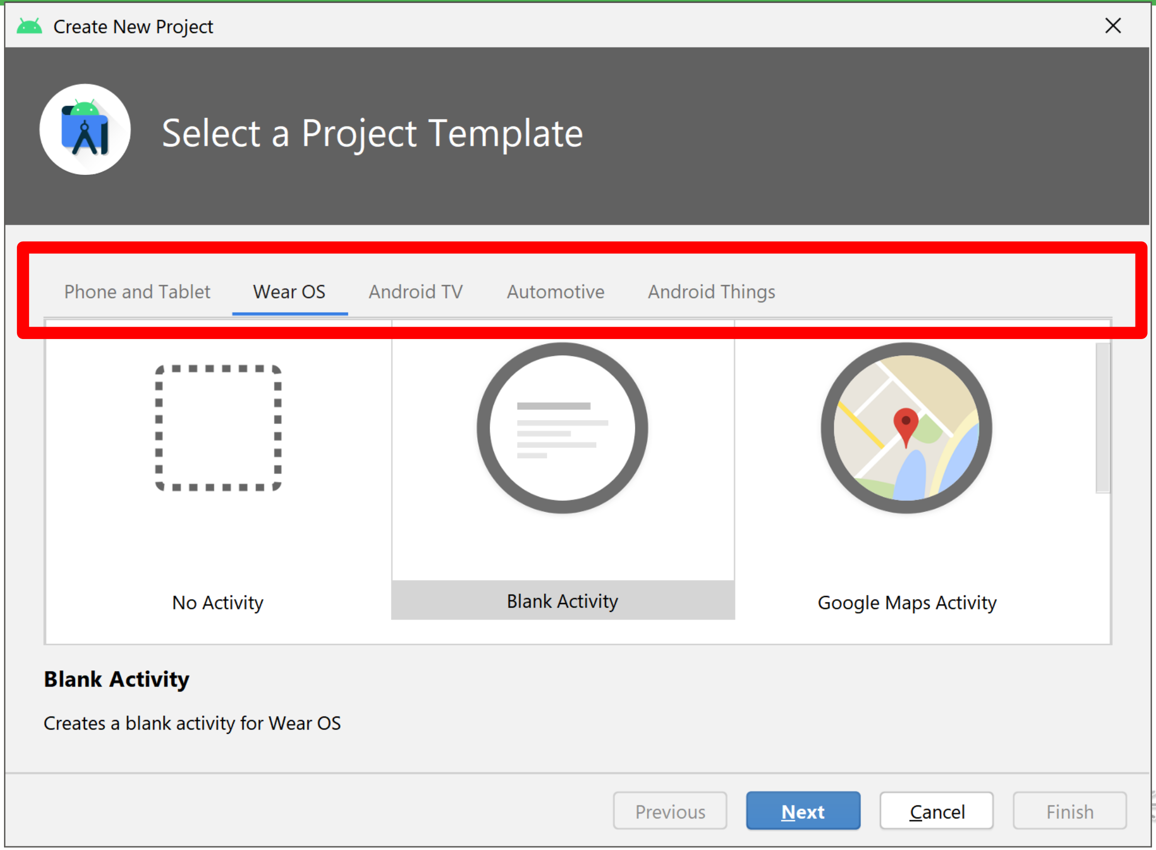
Pick **Empty Activity** or **Basic Activity** for simple and custom activities



Pick activity template

Choose templates for **common activities**, such as maps or navigation drawers

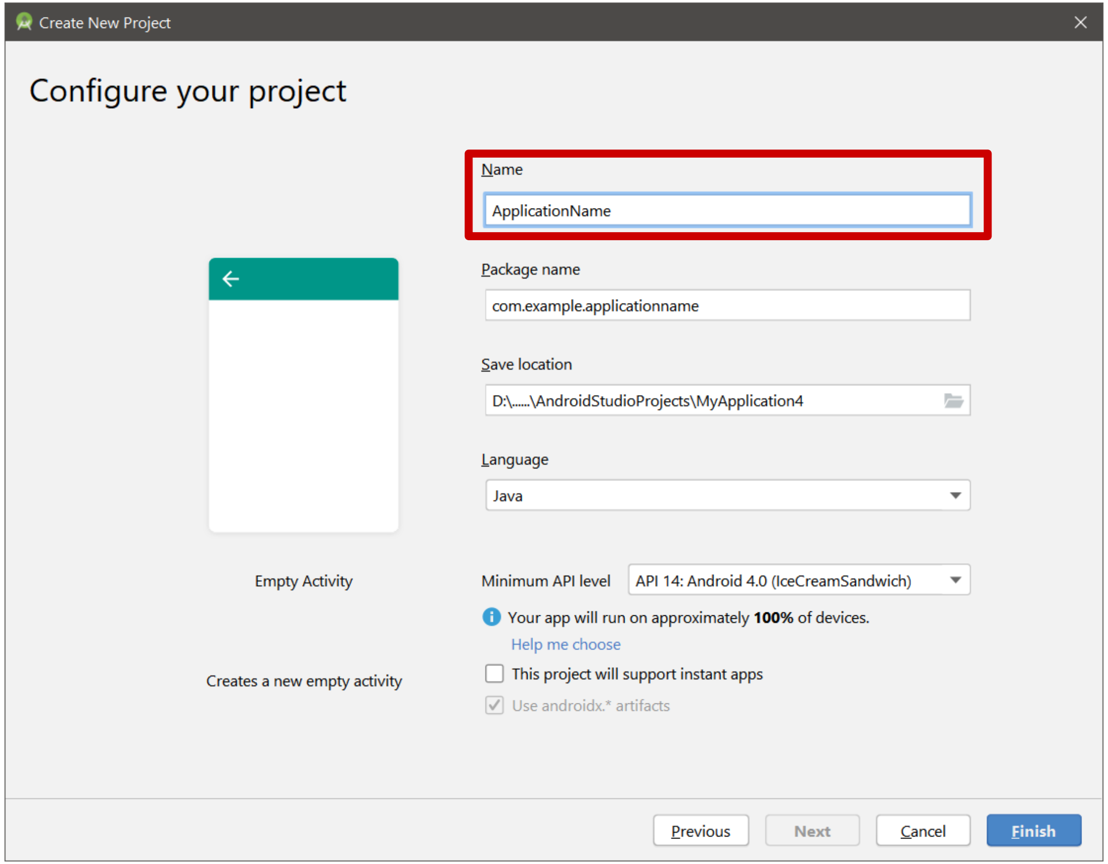
Pick **Empty Activity** or **Basic Activity** for simple and custom activities



Blank Activity

Creates a blank activity for Wear OS

Name your app



Create New Project

Configure your project

Name

ApplicationName

Package name

com.example.applicationname

Save location

D:\......\AndroidStudioProjects\MyApplication4

Language

Java

Minimum API level API 14: Android 4.0 (IceCreamSandwich)

Your app will run on approximately **100%** of devices.

Help me choose

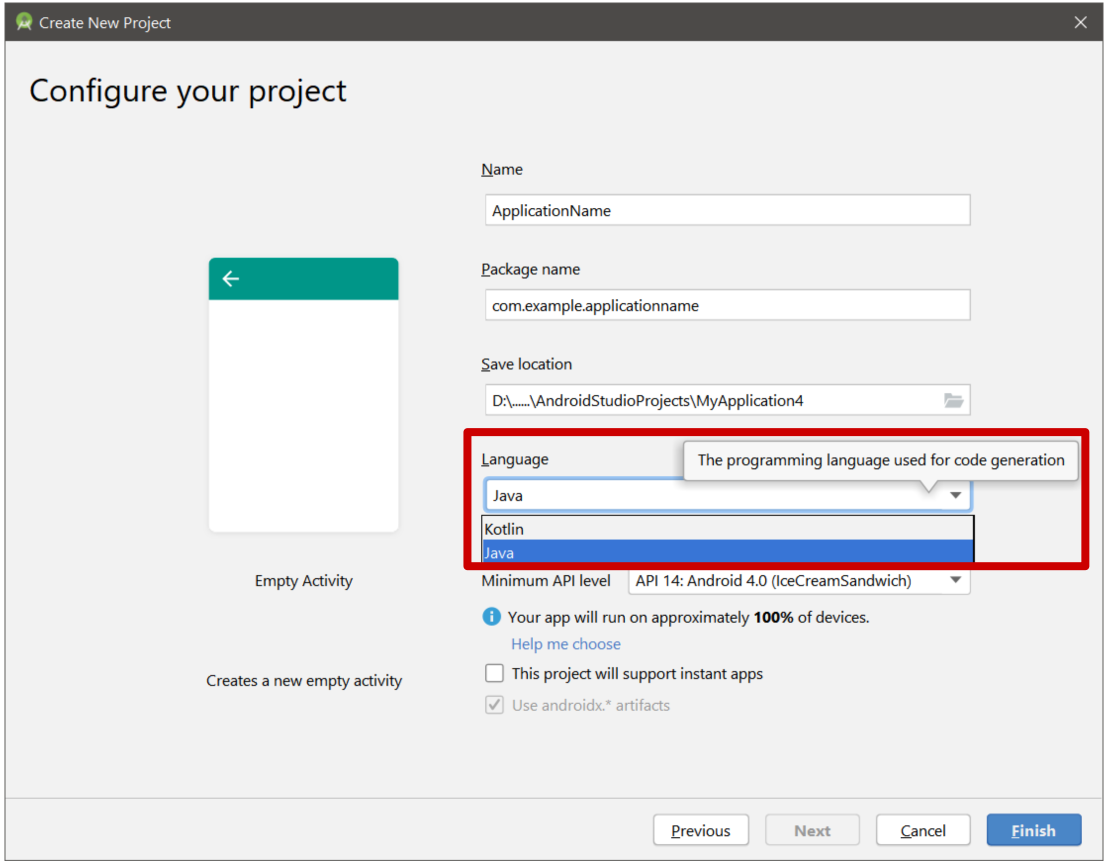
☐ This project will support instant apps

☑ Use androidx.* artifacts

Empty Activity

Creates a new empty activity

Language to code your app



Create New Project

Configure your project

Name: ApplicationName

Package name: com.example.applicationname

Save location: D:\......\AndroidStudioProjects\MyApplication4

Language: Java

The programming language used for code generation

Kotlin
Java

Minimum API level: API 14: Android 4.0 (IceCreamSandwich)

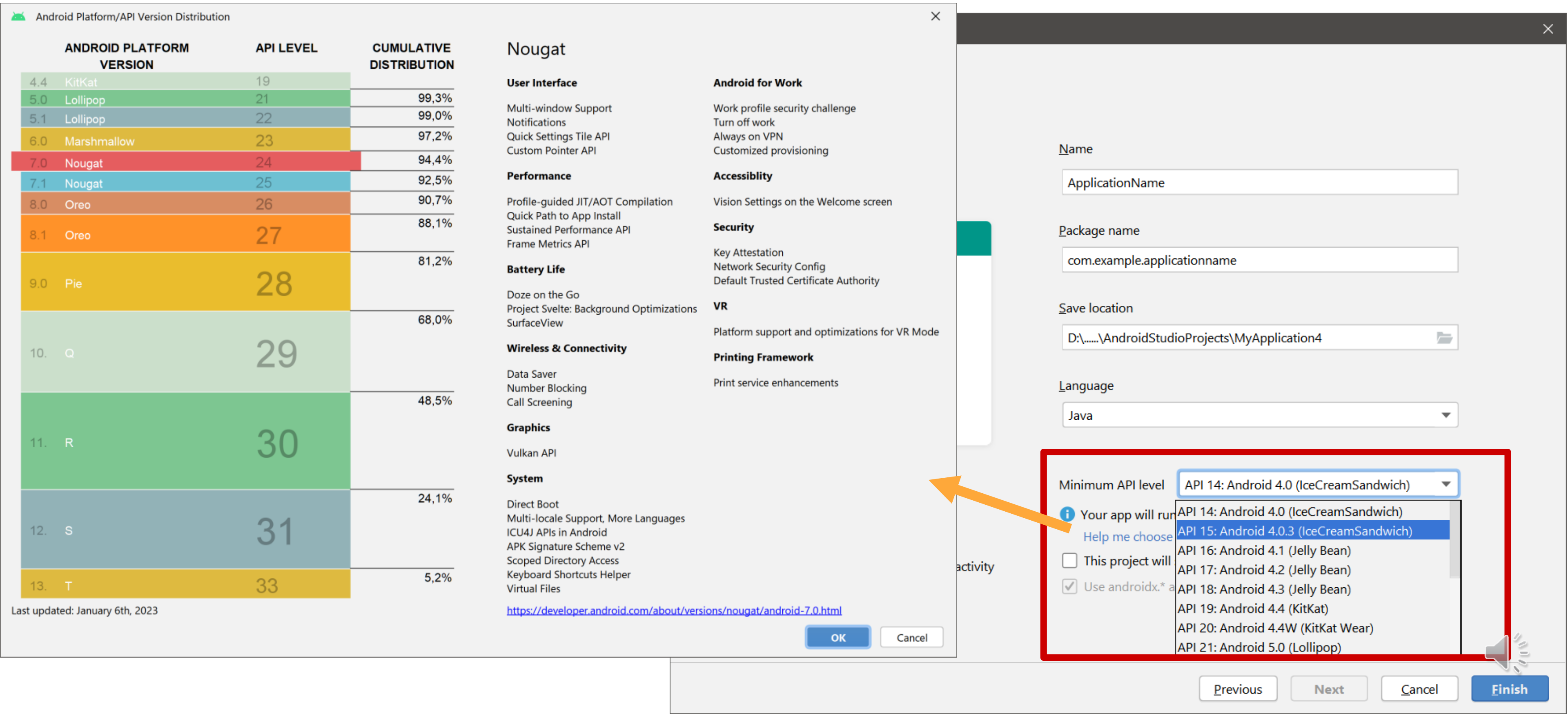
Your app will run on approximately **100%** of devices.

Help me choose

This project will support instant apps

Use androidx.* artifacts

Min API level for your app



Android Platform/API Version Distribution

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.4	KitKat	19	
5.0	Lollipop	21	99,3%
5.1	Lollipop	22	99,0%
6.0	Marshmallow	23	97,2%
7.0	Nougat	24	94,4%
7.1	Nougat	25	92,5%
8.0	Oreo	26	90,7%
8.1	Oreo	27	88,1%
9.0	Pie	28	81,2%
10.	Q	29	68,0%
11.	R	30	48,5%
12.	S	31	24,1%
13.	T	33	5,2%

Last updated: January 6th, 2023

Nougat

User Interface

Multi-window Support
Notifications
Quick Settings Tile API
Custom Pointer API

Performance

Profile-guided JIT/AOT Compilation
Quick Path to App Install
Sustained Performance API
Frame Metrics API

Battery Life

Doze on the Go
Project Svelte: Background Optimizations
SurfaceView

Wireless & Connectivity

Data Saver
Number Blocking
Call Screening

Graphics

Vulkan API

System

Direct Boot
Multi-locale Support, More Languages
ICU4J APIs in Android
APK Signature Scheme v2
Scoped Directory Access
Keyboard Shortcuts Helper
Virtual Files

Android for Work

Work profile security challenge
Turn off work
Always on VPN
Customized provisioning

Accessiblity

Vision Settings on the Welcome screen

Security

Key Attestation
Network Security Config
Default Trusted Certificate Authority

VR

Platform support and optimizations for VR Mode

Printing Framework

Print service enhancements

https://developer.android.com/about/versions/nougat/android-7.0.html

Name: ApplicationName

Package name: com.example.applicationname

Save location: D:\......\AndroidStudioProjects\MyApplication4

Language: Java

Minimum API level: API 14: Android 4.0 (IceCreamSandwich)

Your app will run

Help me choose

This project will

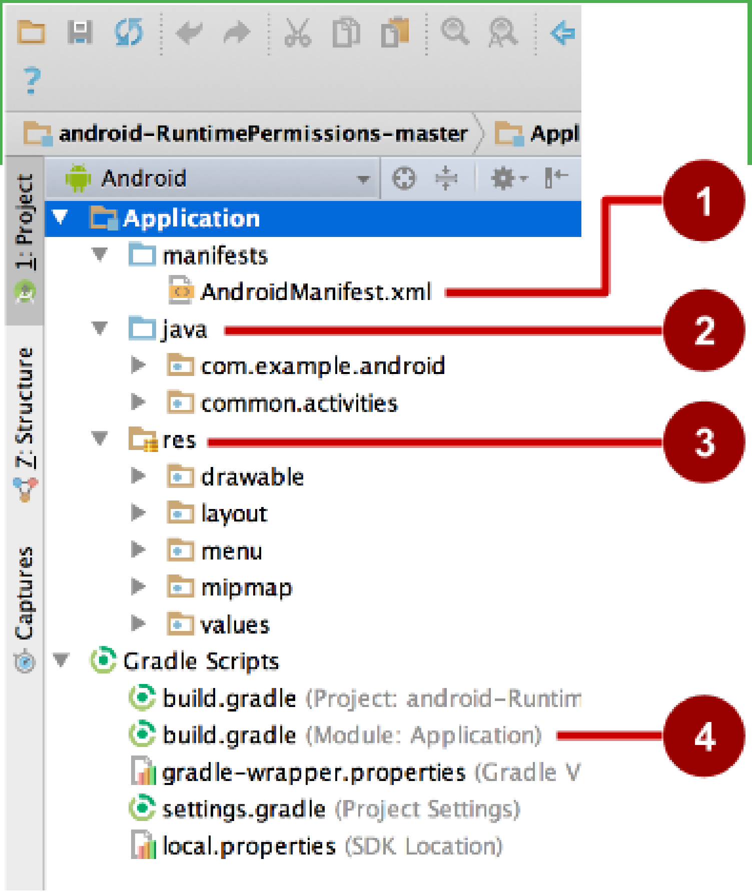
Use androidx.* a

activity

API 14: Android 4.0 (IceCreamSandwich)
API 15: Android 4.0.3 (IceCreamSandwich)
API 16: Android 4.1 (Jelly Bean)
API 17: Android 4.2 (Jelly Bean)
API 18: Android 4.3 (Jelly Bean)
API 19: Android 4.4 (KitKat)
API 20: Android 4.4W (KitKat Wear)
API 21: Android 5.0 (Lollipop)
Previous
Next
Cancel
Finish

Project folders

1. **manifests**—Android Manifest file - description of app read by the Android runtime
2. **java**—Java source code packages
3. **res**—Resources (XML) - layout, strings, images, dimensions, colors...
4. **build.gradle**—Gradle build files

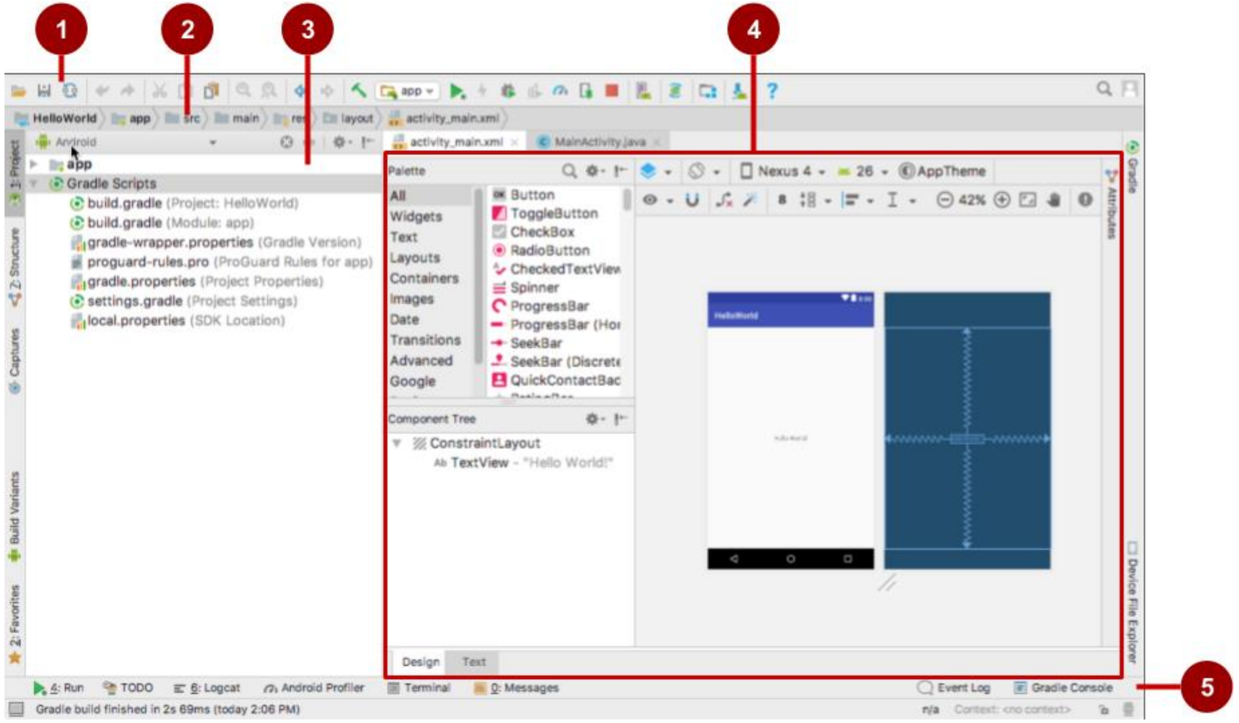


android-RuntimePermissions-master
Appl
1: Project
7: Structure
Captures
Android
Application
manifests
AndroidManifest.xml
java
com.example.android
common.activities
res
drawable
layout
menu
mipmap
values
Gradle Scripts
build.gradle (Project: android-Runtim
build.gradle (Module: Application)
gradle-wrapper.properties (Gradle V
settings.gradle (Project Settings)
local.properties (SDK Location)
1
2
3
4

Gradle build system

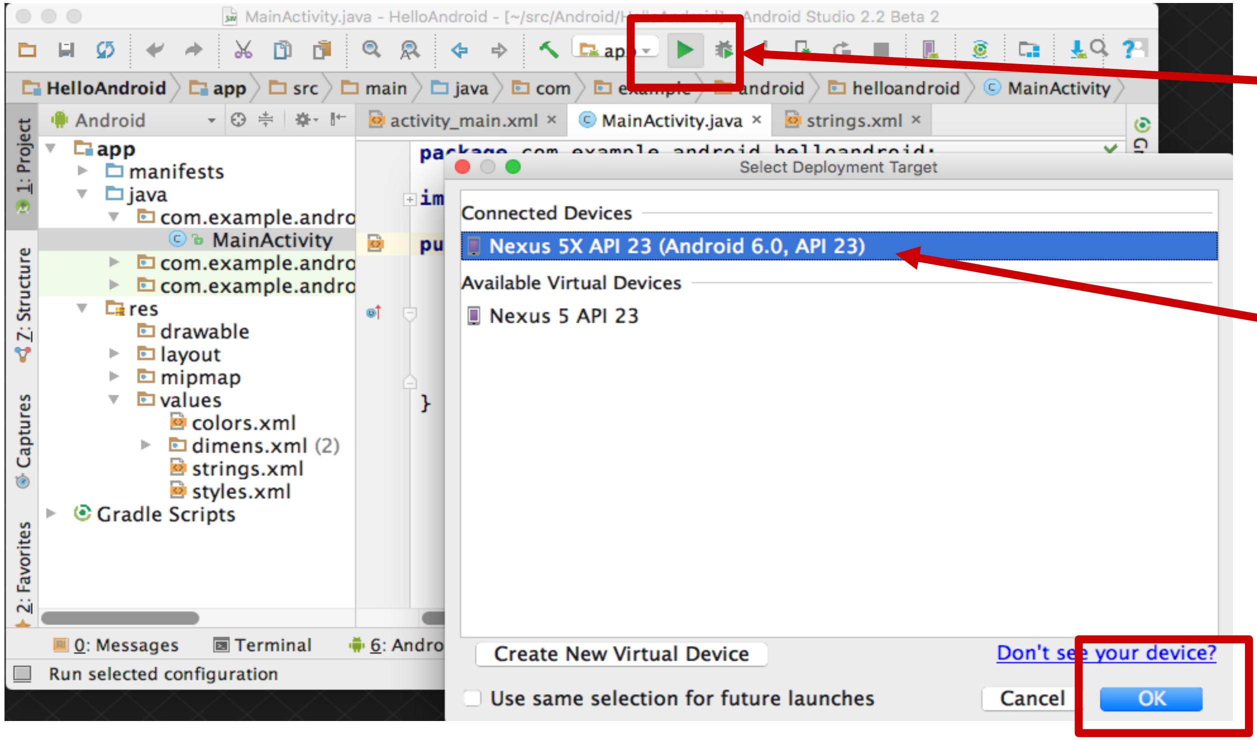
- Modern build subsystem in Android Studio
  - Generates the APK file (Android Application Package)
- Typically not necessary to know low-level Gradle details
- Learn more about gradle at https://gradle.org/
- Learn more on how to configure apps build
  - https://developer.android.com/studio/build

Android Studio interface



1. Toolbar
2. Navigation bar
3. Project pane
4. Editor
5. Tabs for other panes

Run your app



1. Run

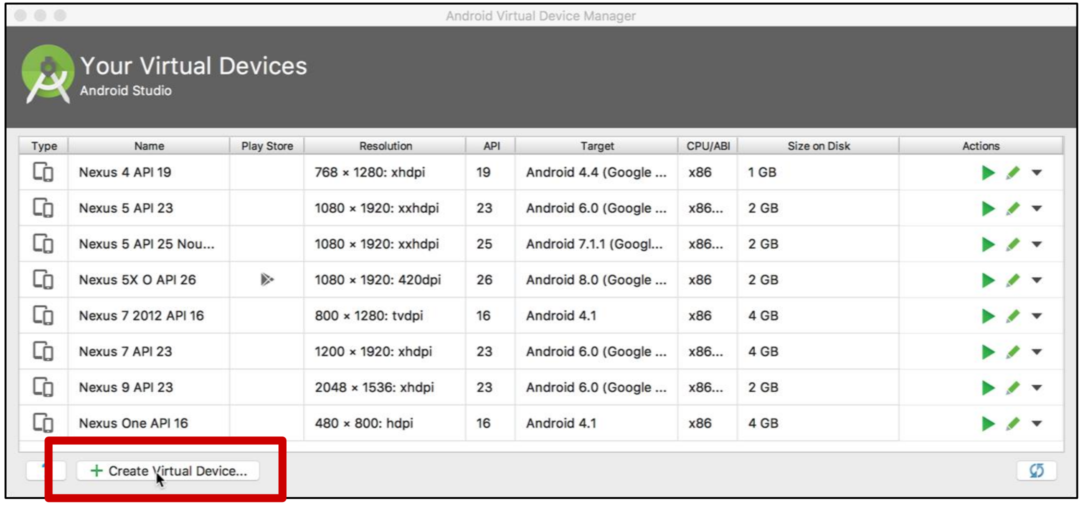
2. Select virtual or physical device

3. OK

Create a virtual device

Use emulators to test app on different versions of Android and form factors.

Tools > Android > AVD Manager or:

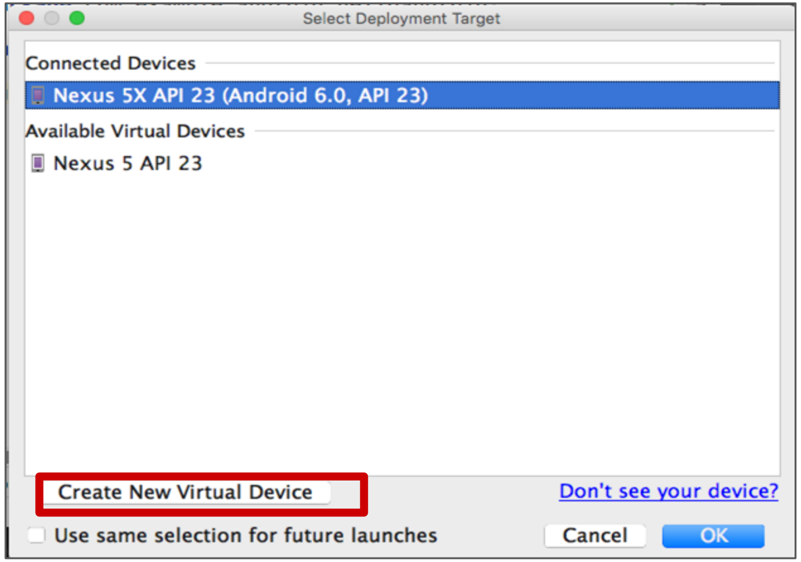


Android Virtual Device Manager

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 4 API 19		768 × 1280: xhdpi	19	Android 4.4 (Google ...	x86	1 GB	
	Nexus 5 API 23		1080 × 1920: xxhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus 5 API 25 Nou...		1080 × 1920: xxhdpi	25	Android 7.1.1 (Googl...	x86...	2 GB	
	Nexus 5X O API 26		1080 × 1920: 420dpi	26	Android 8.0 (Google ...	x86	2 GB	
	Nexus 7 2012 API 16		800 × 1280: tvdpi	16	Android 4.1	x86	4 GB	
	Nexus 7 API 23		1200 × 1920: xhdpi	23	Android 6.0 (Google ...	x86...	4 GB	
	Nexus 9 API 23		2048 × 1536: xhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus One API 16		480 × 800: hdpi	16	Android 4.1	x86	4 GB	

+ Create Virtual Device...



Select Deployment Target

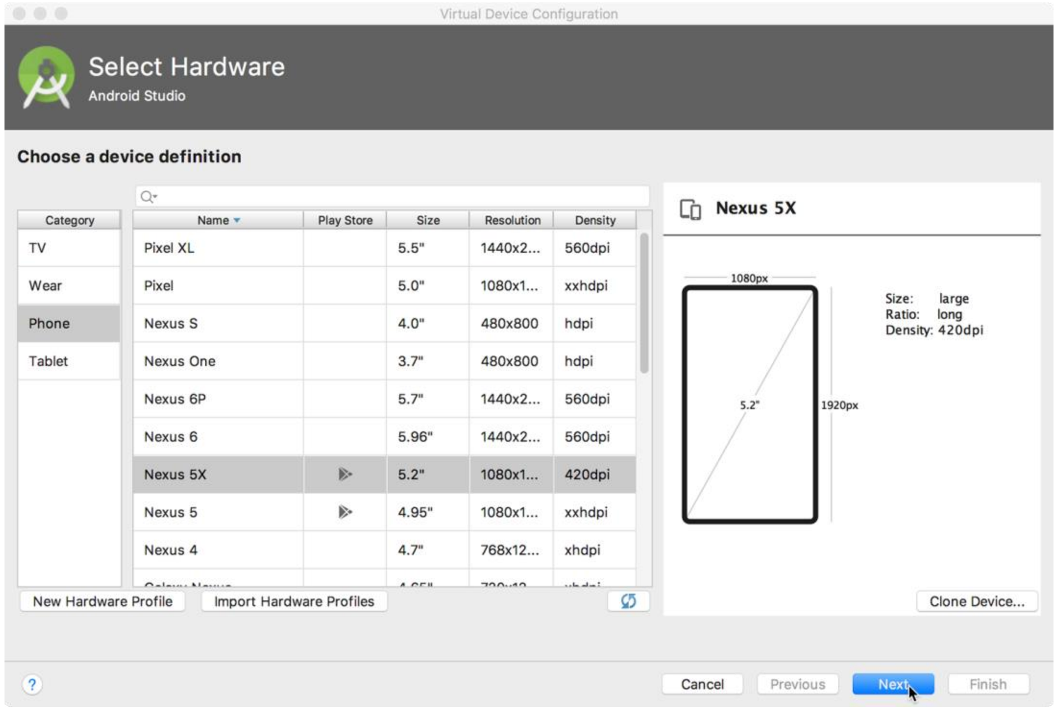
Connected Devices
Nexus 5X API 23 (Android 6.0, API 23)

Available Virtual Devices
Nexus 5 API 23

Create New Virtual Device
Don't see your device?
Use same selection for future launches
Cancel
OK

Configure virtual device

1. Choose hardware
2. Select Android version
3. Finalize


Virtual Device Configuration

Select Hardware
Android Studio

Choose a device definition

Category
TV
Wear
Phone
Tablet

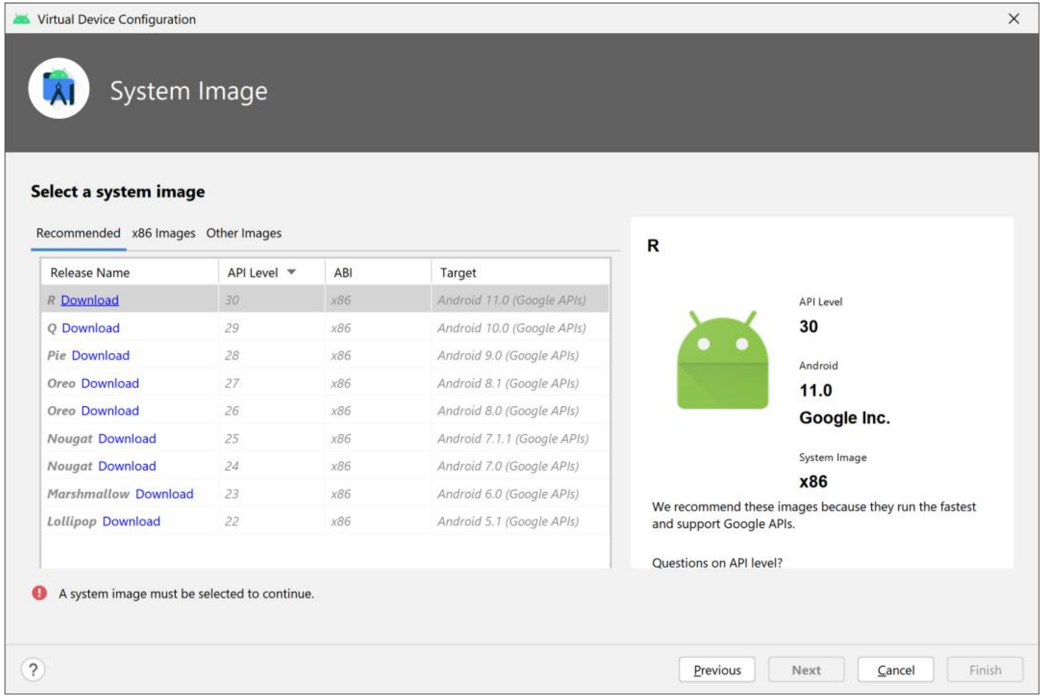
Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2...	560dpi
Pixel		5.0"	1080x1...	xxhdpi
Nexus S		4.0"	480x800	hdpi
Nexus One		3.7"	480x800	hdpi
Nexus 6P		5.7"	1440x2...	560dpi
Nexus 6		5.96"	1440x2...	560dpi
Nexus 5X	▶	5.2"	1080x1...	420dpi
Nexus 5	▶	4.95"	1080x1...	xxhdpi
Nexus 4		4.7"	768x12...	xhdpi

Nexus 5X

Size: large
Ratio: long
Density: 420dpi

Configure virtual device

1. Choose hardware
2. Select Android version
3. Finalize



Virtual Device Configuration

System Image

Select a system image

Recommended | x86 Images | Other Images

Release Name	API Level	ABI	Target
R Download	30	x86	Android 11.0 (Google APIs)
Q Download	29	x86	Android 10.0 (Google APIs)
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Lollipop Download	22	x86	Android 5.1 (Google APIs)

R

API Level
30

Android
11.0
Google Inc.

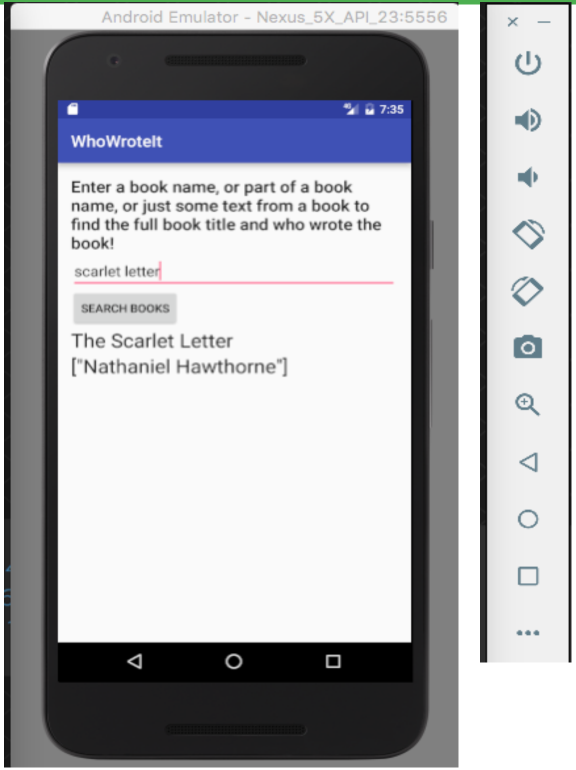
System Image
x86

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?

A system image must be selected to continue.

Run on a virtual device



Android Emulator - Nexus_5X_API_23:5556

7:35

WhoWroteIt

Enter a book name, or part of a book name, or just some text from a book to find the full book title and who wrote the book!

scarlet letter

SEARCH BOOKS

The Scarlet Letter
["Nathaniel Hawthorne"]

Run on a physical device

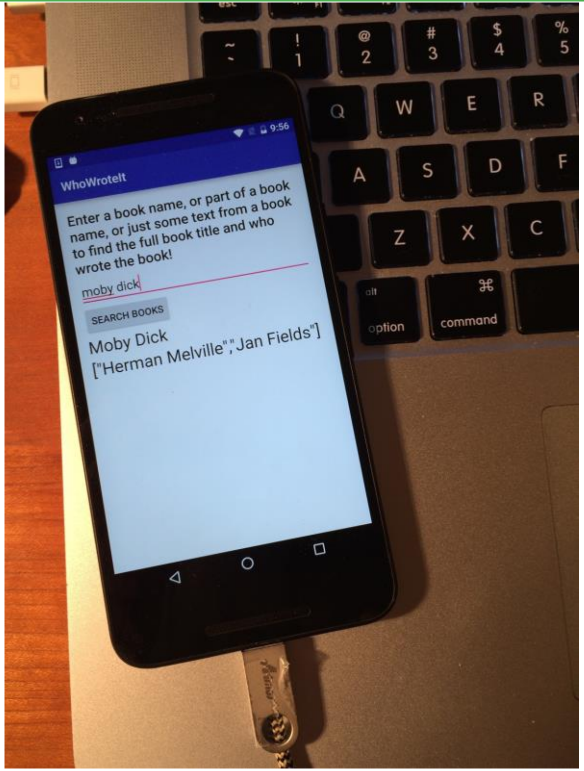
1. Turn on Developer Options:
   a. **Settings > About phone**
   b. Tap **Build number** seven times
2. Turn on USB Debugging
   a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- https://developer.android.com/studio/run/device

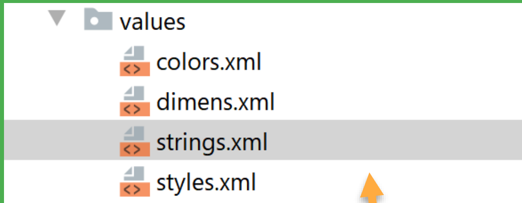
Windows drivers:

- https://developer.android.com/studio/run/oem-usb



WhoWroteIt
Enter a book name, or part of a book name, or just some text from a book to find the full book title and who wrote the book!
moby dick
SEARCH BOOKS
Moby Dick
["Herman Melville","Jan Fields"]

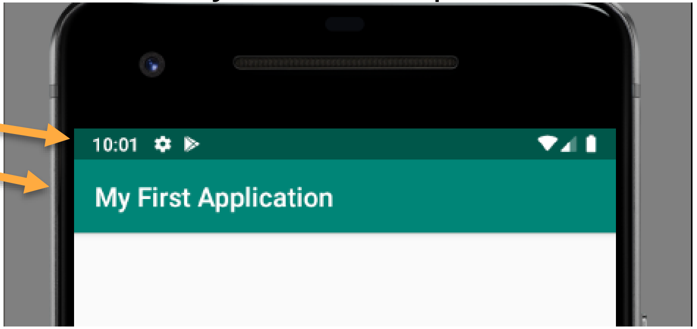
Exercise One


values
colors.xml
dimens.xml
strings.xml
styles.xml

- Develop the Hello World => https://codelabs.developers.google.com/codelabs/android-training-hello-world/
- Run the Hello World app (i.e. an empty app) on the Emulator
- Run the Hello World app on your real device (if you have one)
- Change the Name of the App
  - e.g., from 'My Application' to 'My First Application' (see xml files in "res" folder)
- Change the color of these two components

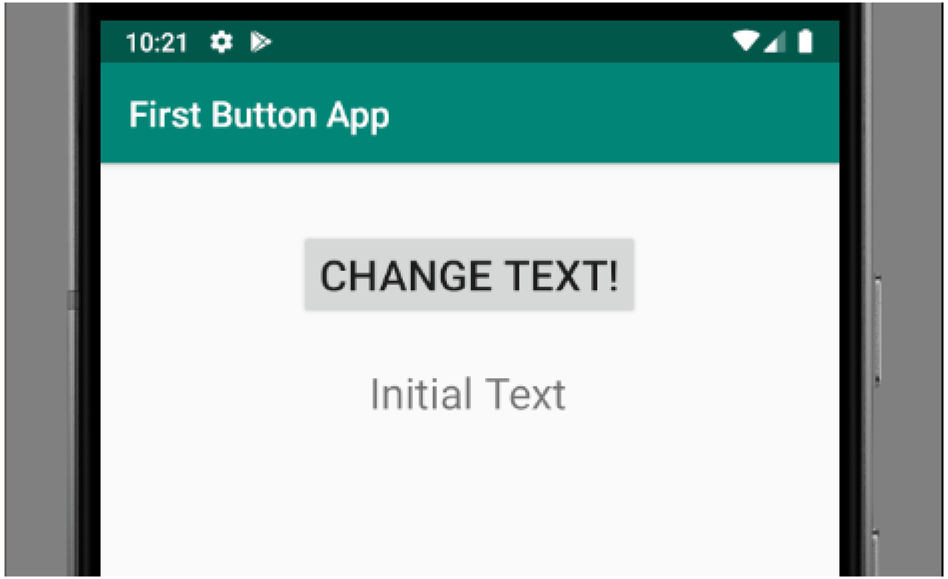
hint: colors.xml, analyse the color picker functionality

Check that everything works



Exercise Two

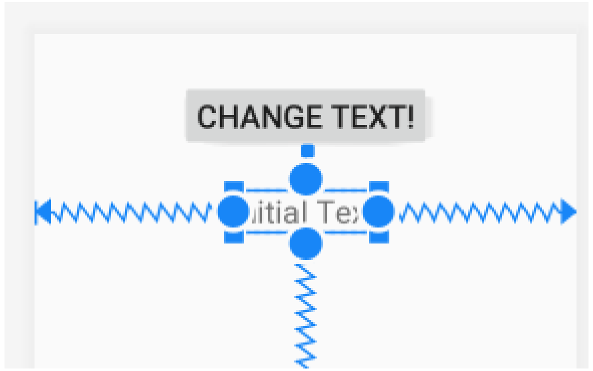
- We want to interact with the app using a button.... Create a new app
- Then create a simple activity like this:

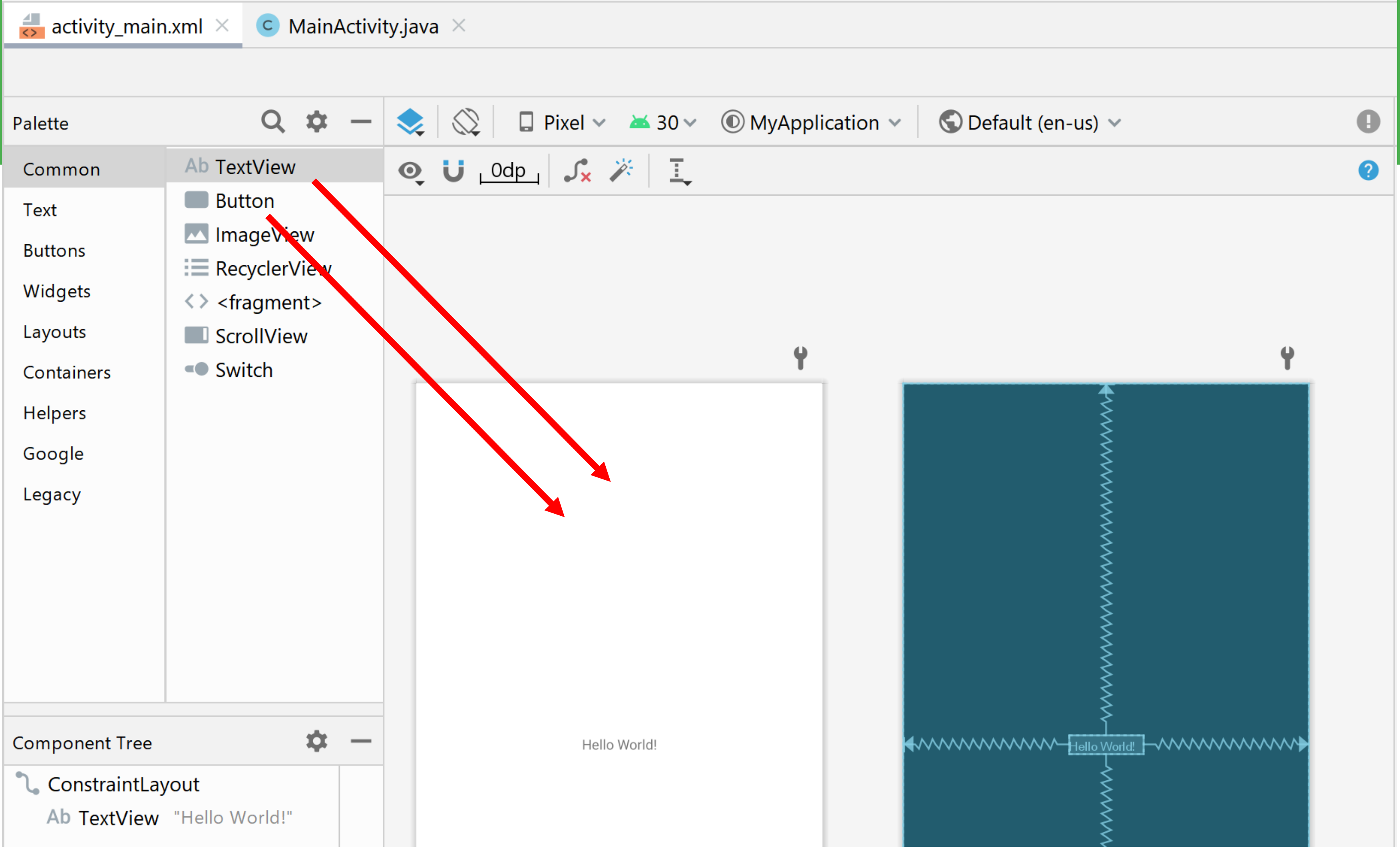


- A title
- A button
- A textView

Hint: open the layout file e.g. activity_main.xml as shown in the next slide

Use the default Costraint Layout





activity_main.xml
MainActivity.java
Palette
Pixel
30
MyApplication
Default (en-us)
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
0dp
Hello World!
Hello World!
Component Tree
ConstraintLayout
Ab TextView "Hello World!"

Exercise Two

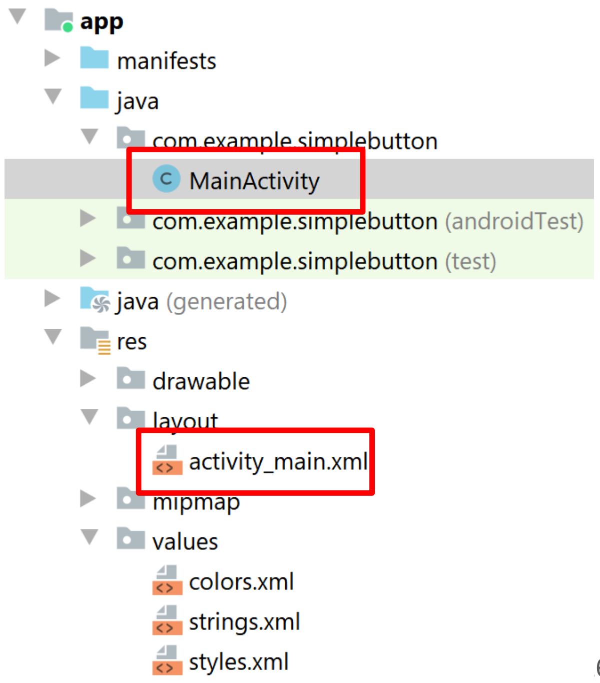
Required Behavior:

- when the **User clicks** on the **button** labelled «*Change Text!*» the **textView** must change its text from «*Initial Text*» to «*Updated Text!!*»

We have to:

- Write a method that implements such behavior (in java/MainActivity.java)
- Associate such method to the button

Project Structure

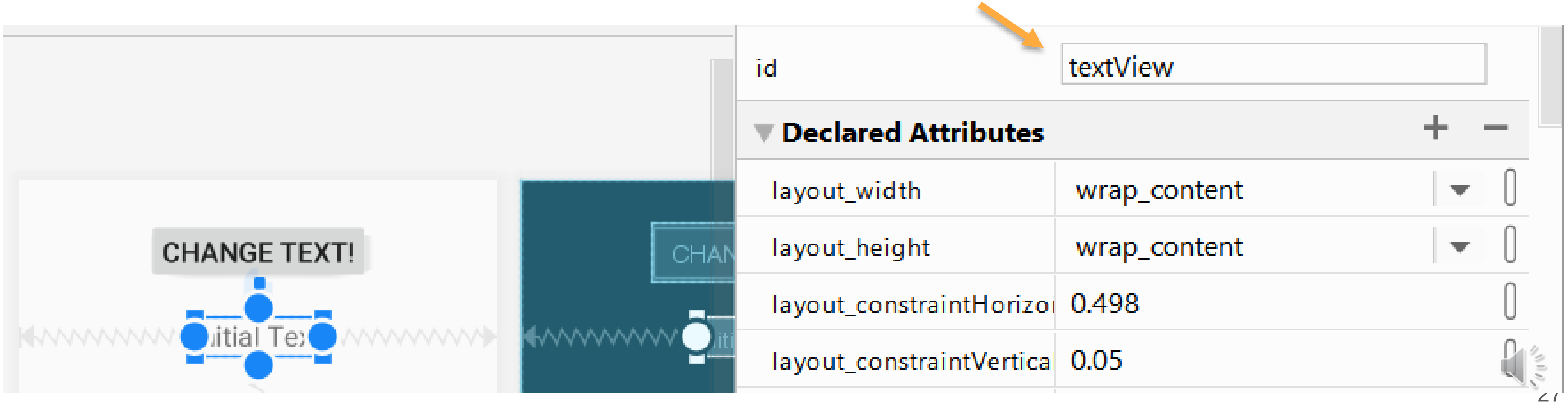


Exercise Two

Hint: open the layout file e.g. activity_main.xml

In detail:

- Insert the widgets in the main activity (Button and TextView)
- Provide a meaningful ID to our textView



id	textView

Declared Attributes

layout_width	wrap_content
layout_height	wrap_content
layout_constraintHorizo	0.498
layout_constraintVertica	0.05

Exercise Two

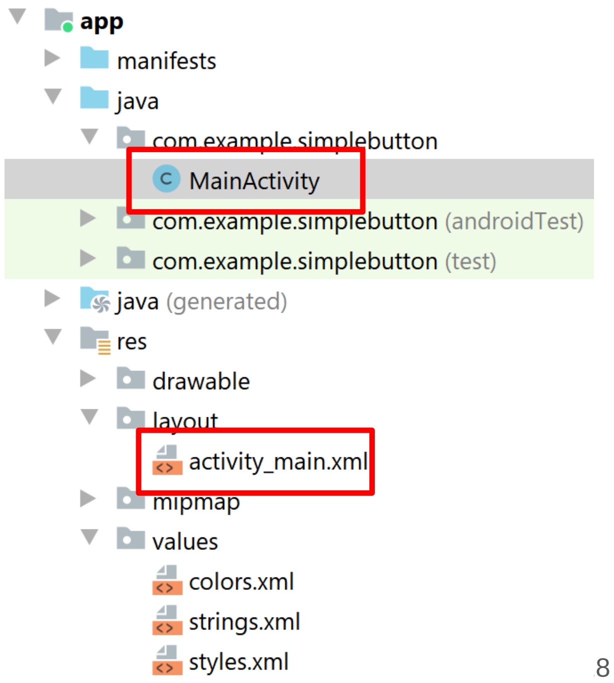
In detail:

Write the code implementing the behavior in 'MainActivity.java'

Idea:

- when the user clicks on the button
- find the textView element (using the ID)
- assign the new text to it

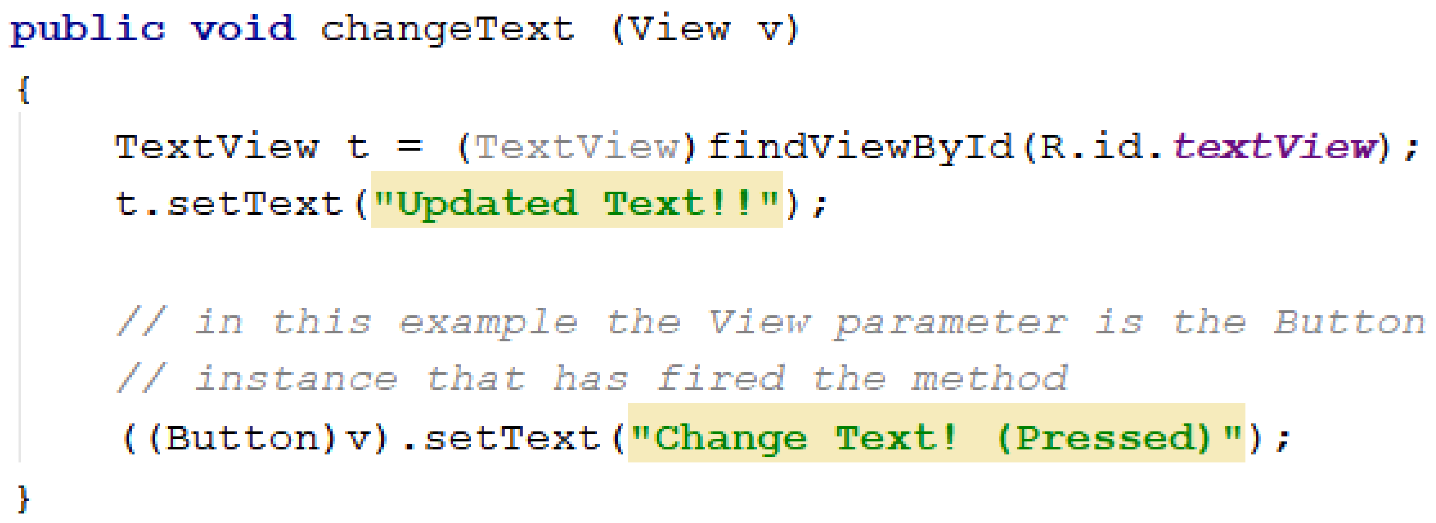
Project Structure



Exercise Two

In detail:

Write the code implementing the behavior in ‘MainActivity.java’



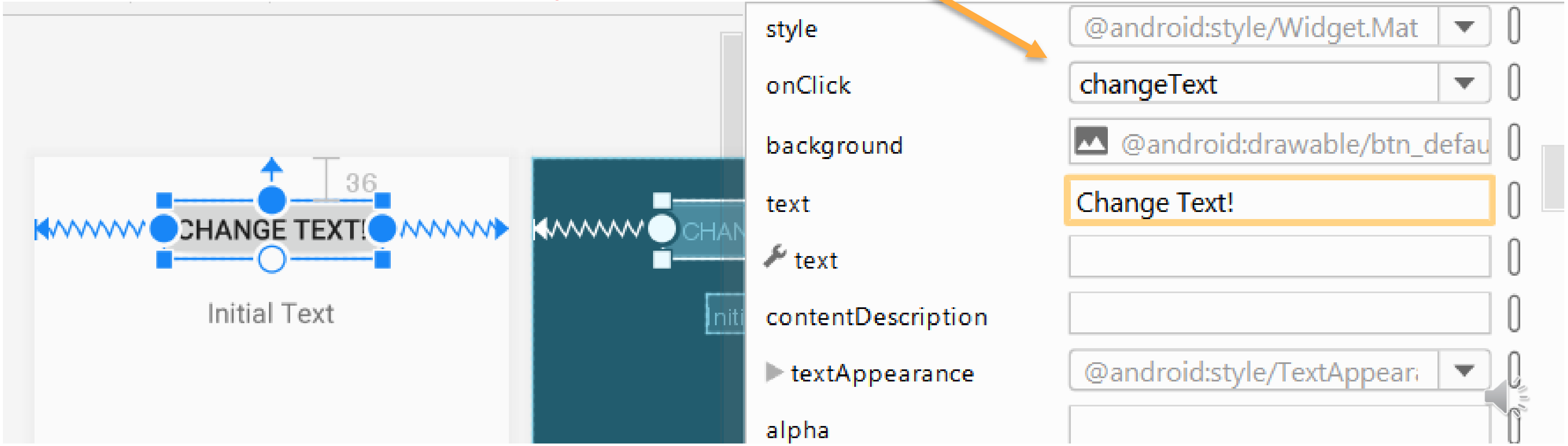
```
public void changeText (View v)
{
    TextView t = (TextView)findViewById(R.id.textView);
    t.setText("Updated Text!!");

    // in this example the View parameter is the Button
    // instance that has fired the method
    ((Button)v).setText("Change Text!  (Pressed)");
}
```

Exercise Two

In detail:

Call **changeText** method in response to **onClick** events on the button

in activity_main.xml



style	@android:style/Widget.Mat
onClick	changeText
background	@android:drawable/btn_defau
text	Change Text!
text	
contentDescription	
textAppearance	@android:style/TextAppear
alpha	