Laboratorio di Applicazioni Mobili
Bachelor in Computer Science &
Computer Science for Management

University of Bologna

Overview & Setup

Federico Montori
federico.montori2@unibo.it
(a few slides are courtesy of Luca Sciullo)

Table of Contents

- Mobile Applications
  - Native
  - Hybrid
  - Web apps
- Android Studio

Mobile Apps

“A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch”

Wikipedia

Mobile Apps: **Native**

- A native mobile app is intimately tied to the platform on which it is running.
- It has the ability to fully integrate with the capabilities of both the hardware and the OS on which it resides.
- It is completely analogous to most PC applications, which are downloaded to a desktop or laptop hard drive and completely executed within that machine.
- In order to accomplish this tight integration, the mobile app developer utilizes an SDK from the hardware manufacturer directly or via the mobile device's OS vendor.
- Combined with an IDE, a developer is able to code the mobile application logic and take advantage of any hardware or OS functionality that is available via the exposed APIs

Mobile Apps: **Hybrid**

- A hybrid mobile application is developed using both native libraries and web technologies in an attempt to get the best of both worlds.
- The interface between the separate components is an on-platform, embedded HTML rendering engine, which is either developed in-house or by acquiring one from a 3rd-party.
- The native portion of the app can be written as a top to bottom native app, which communicates to a web-based server backend. This has the same porting issues as a purely native app.
- 3rd-party cross-platform development tools exist that use native library containers to achieve near-native performance. Such tools bring the benefits of cross-platform development in both the native and web-based portions of a hybrid mobile app

Mobile Apps: **Web apps**

- Web-based mobile apps and **Progressive Web Apps** (PWA) are developed with the same tools used for mobile website development through the use of HTML, CSS style sheets and JavaScript.
- HTML5 provides the ability to create rich UI experiences with support for rich media, UI components, geolocation, and offline execution.
- Third-party suppliers of JavaScript toolkits can supply UI components that allow web-based apps to mimic native look and feel on the mobile device, such as Dojo or jQuery. However, their ability to provide precise native look and feel varies across toolkits.

Mobile Apps: **Comparison**

Hybrid

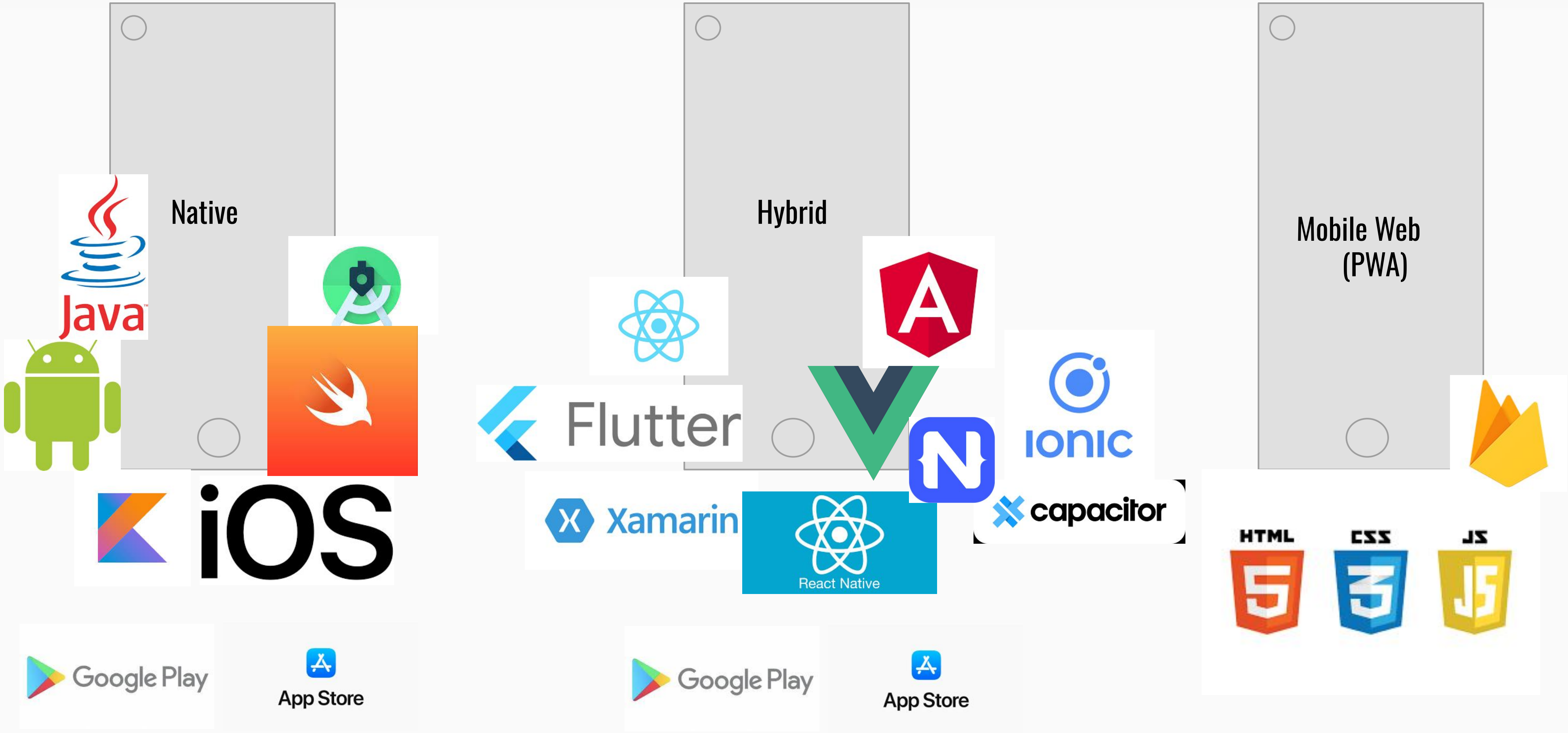
- Web technologies (HTML, CSS, Javascript)
- Run in a browser Web View
- Access device capabilities via plugins
- Wrapped in a native app shell
- Native app

PWA

It is just a regular website that runs in a browser with some enhancements and gives app-like experience to users by using modern web capabilities.

- Installation on a mobile home screen
- Offline usage
- Camera, push notifications
- Background synchronization

Mobile Apps: **Comparison**

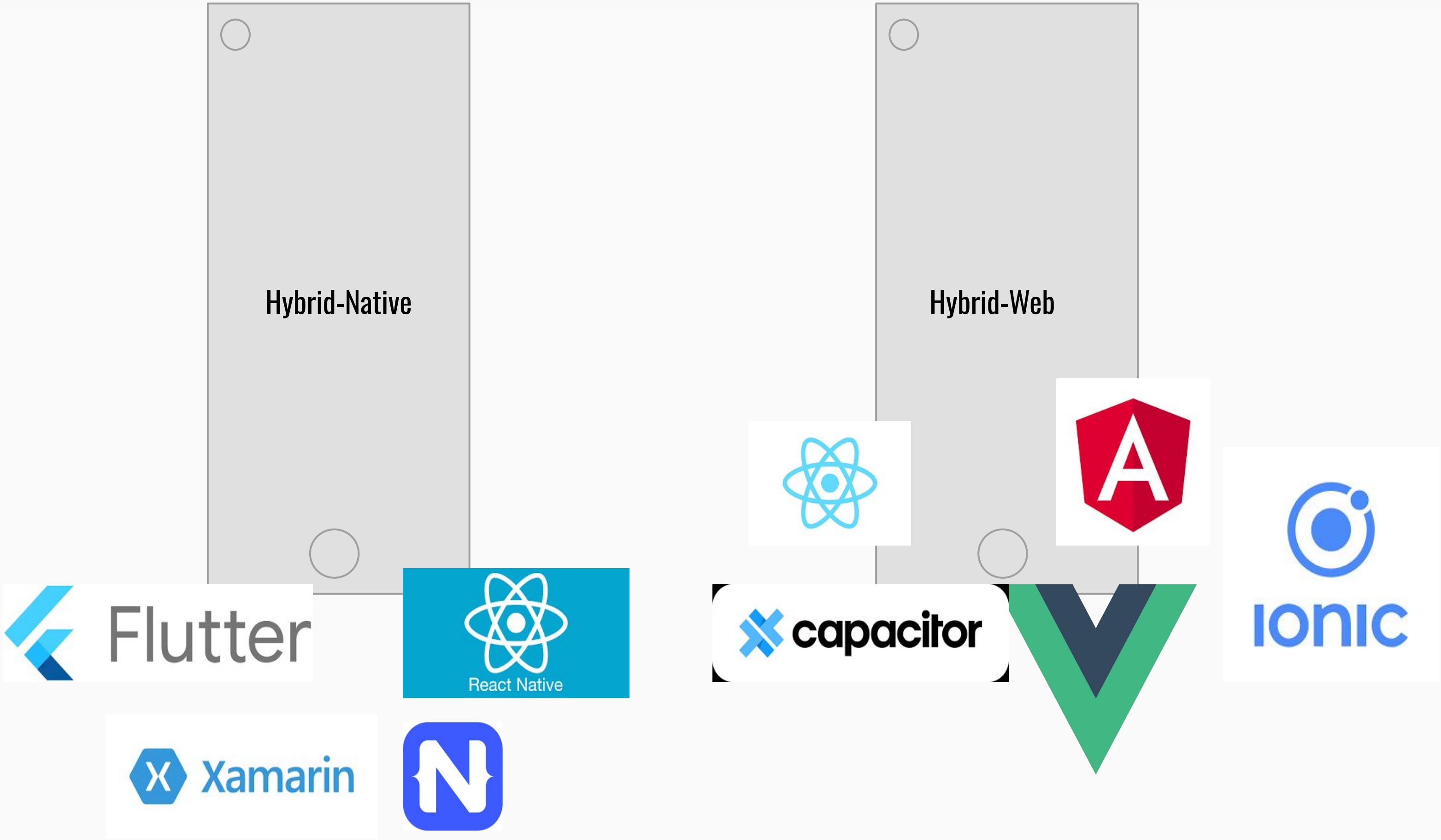


Native

Hybrid

Mobile Web (PWA)

Mobile Apps: **Comparison**



Hybrid-Native

Hybrid-Web

Mobile Apps: **Comparison**

Owned by Ionic

- Basically Web code running into a WebView
- Capacitor is an evolution of Cordova/PhoneGap
- Works very well for the Web too
- Good if you want to reuse code from your Web Application (Angular, React, Vue, …)

Owned by Facebook

- You need to code in React
- Renders elements in native elements
- EXPO
- Few basic components, (too) many plugins
- Good if you want to reuse (some) React code but be closer to the native render

Owned by Google

- Code in Dart, render in native
- Declarative UI (similar to Swift/Components)
- A lot of widgets OOTB
- Not very good for the web, it renders everything into a canvas.
- Good if you want to start from scratch.

Mobile Apps: **Comparison**

Hybrid-Native: Native UI, Shared Code

JavaScript code runs and orchestrates native UI controls under the hood, so that your app UI is running (almost but not completely) natively. It means that the underlying native UI component that you would like to use or customize must be supported by the framework in order to use that component or change it.

Hybrid-Web: Web UI, Shared Components

The UI is built with HTML/CSS/JS, with native functionality being accessed through portable APIs (or “plugins”) that abstract the underlying platform dependencies.Instead of your entire UI depending on the native platform, only certain native device features, like the Camera, are platform-dependent.

Mobile Apps: **Comparison**

Native, Hybrid or PWA ?

Mobile Apps: **Comparison**

Native, Hybrid or PWA ?

Of course, it depends…

Mobile Apps: **Comparison**

Native

- Direct access to all the mobile device’s features
- Highest performance possible
- Direct access to UI components of specific platforms

Hybrid

- Easy access to mobile device’s features
- UX is nearly identical to native apps
- High performances
- Single codebase, and lower development costs

Web

- No fee nor approval procedures to publish the application
- No installation required
- Lowest development costs
- Single codebase

Mobile Apps: **Comparison**

Native

- Highest Development & Maintenance costs
- Multiple codebases
- High Development Time
- Limited customization

Hybrid

- High Development & Maintenance costs
- The efficiency and the quality of the application strongly depends on the technology used
- Worse performances

Web

- Limited set of functionalities
- Worse UX
- The efficiency and the quality of the application strongly depends on the technology used
- Worst performances

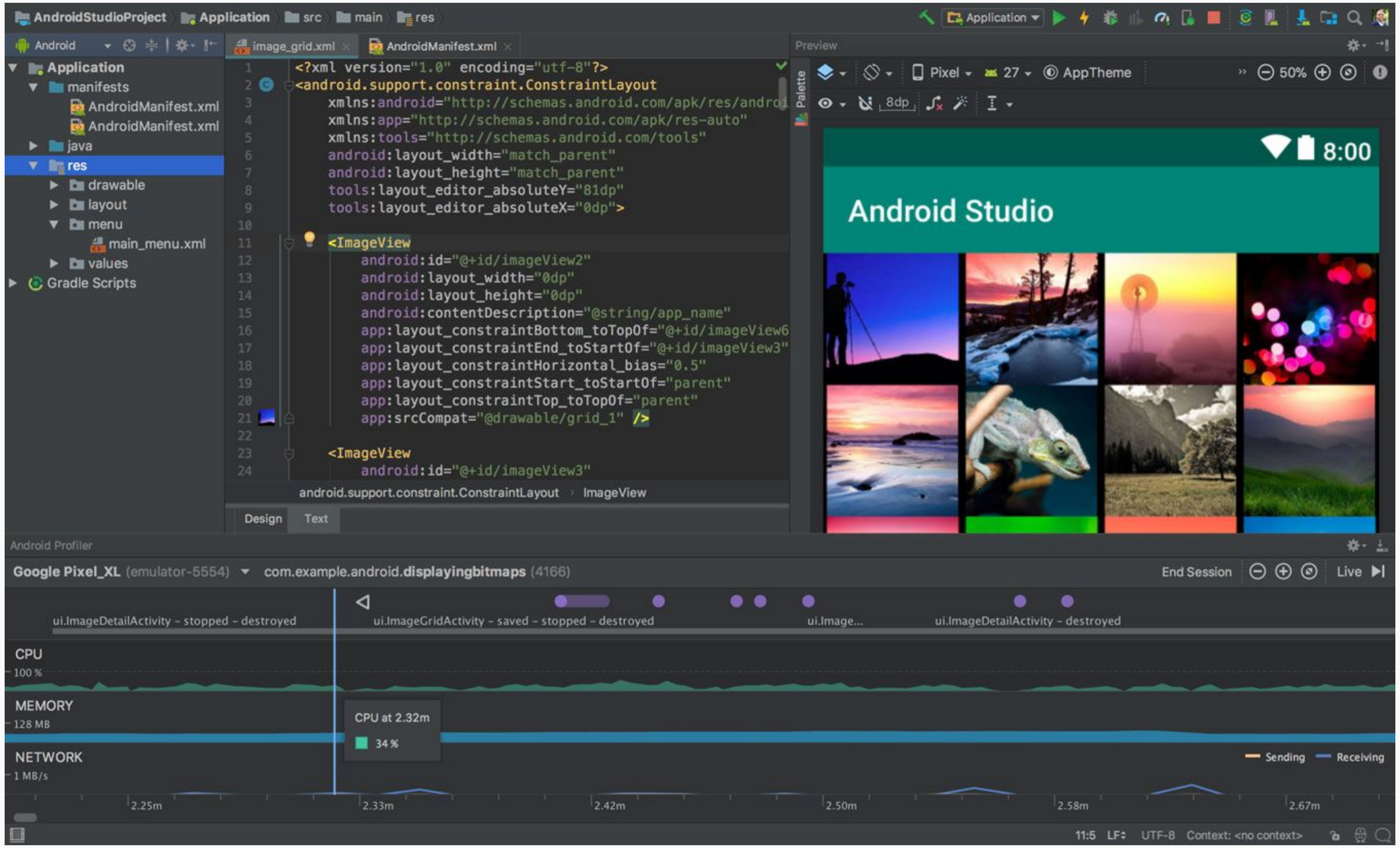
Android Studio

The official development platform for Android Apps is **Android Studio** by JetBrains.

Historically Android development was in Eclipse with and android plugin.

- Eclipse SDK has been DEPRECATED since the end of 2015, which means newer versions of Android are no longer supported.
  - https://android-developers.googleblog.com/2015/06/an-update-on-eclipse-android-developer.html

Android Studio



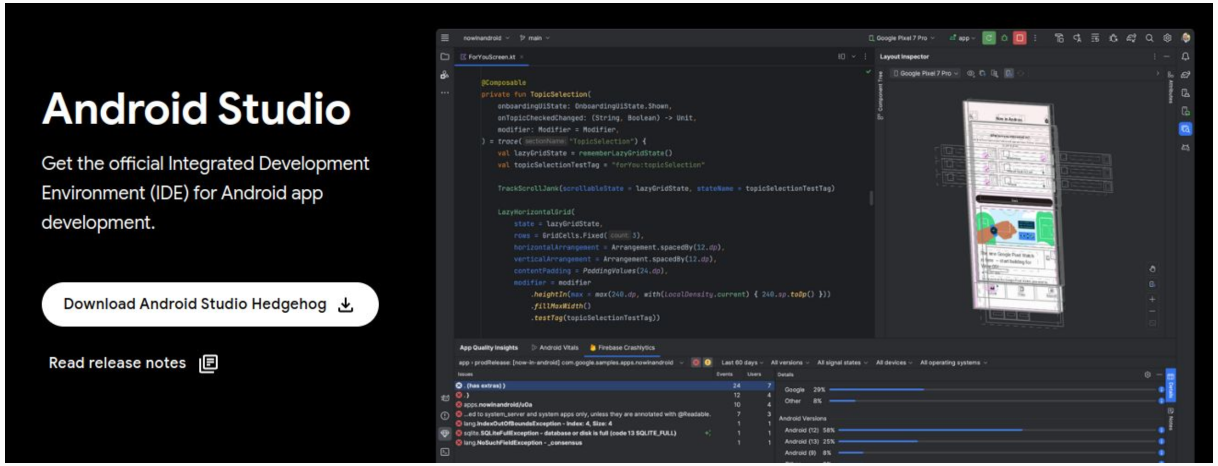
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="81dp"
    tools:layout_editor_absoluteX="0dp">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:contentDescription="@string/app_name"
        app:layout_constraintBottom_toTopOf="@+id/imageView6
        app:layout_constraintEnd_toStartOf="@+id/imageView3"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/grid_1" />

    <ImageView
        android:id="@+id/imageView3"
```

Android Studio

- Go to https://developer.android.com/studio
- Download Android Studio and the SDK
- Install it and you’re done!
    - Version Hedgehog at the time of writing



Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.

Download Android Studio Hedgehog

Read release notes

```
@Composable
private fun TopicSelection(
    onboardingUiState: OnboardingUiState.Shown,
    onTopicCheckedChanged: (String, Boolean) -> Unit,
    modifier: Modifier = Modifier,
) = trace( sectionName: "TopicSelection") {
    val lazyGridState = rememberLazyGridState()
    val topicSelectionTestTag = "forYou:topicSelection"

    TrackScrollJank(scrollableState = lazyGridState, stateName = topicSelectionTestTag)

    LazyHorizontalGrid(
        state = lazyGridState,
        rows = GridCells.Fixed( count: 3),
        horizontalArrangement = Arrangement.spacedBy(12.dp),
        verticalArrangement = Arrangement.spacedBy(12.dp),
        contentPadding = PaddingValues(24.dp),
        modifier = modifier
            .heightIn(max = max(240.dp, with(LocalDensity.current) { 240.sp.toDp() }))
            .fillMaxWidth()
            .testTag(topicSelectionTestTag))
```

Android Studio

How to develop Android applications?

- Linux / Mac OS / Windows? Doesn’t matter
- A real device is not mandatory although suggested...

- Code your application in the IDE
  - Test it with the emulator
  - Deploy it on a real device (if you can)

Small Glossary…

SDK (Software development kit)

- A set of tools that help you in creating software
  - Compiler, tester, debugger, libraries

IDE (Integrated Development Environment)

- Graphical environment in which all the tools are accessible.

API (Application Program Interface)

- A set of calls that the underlying world exposes to the developer for interaction.
  - It does not correspond to “libraries”.

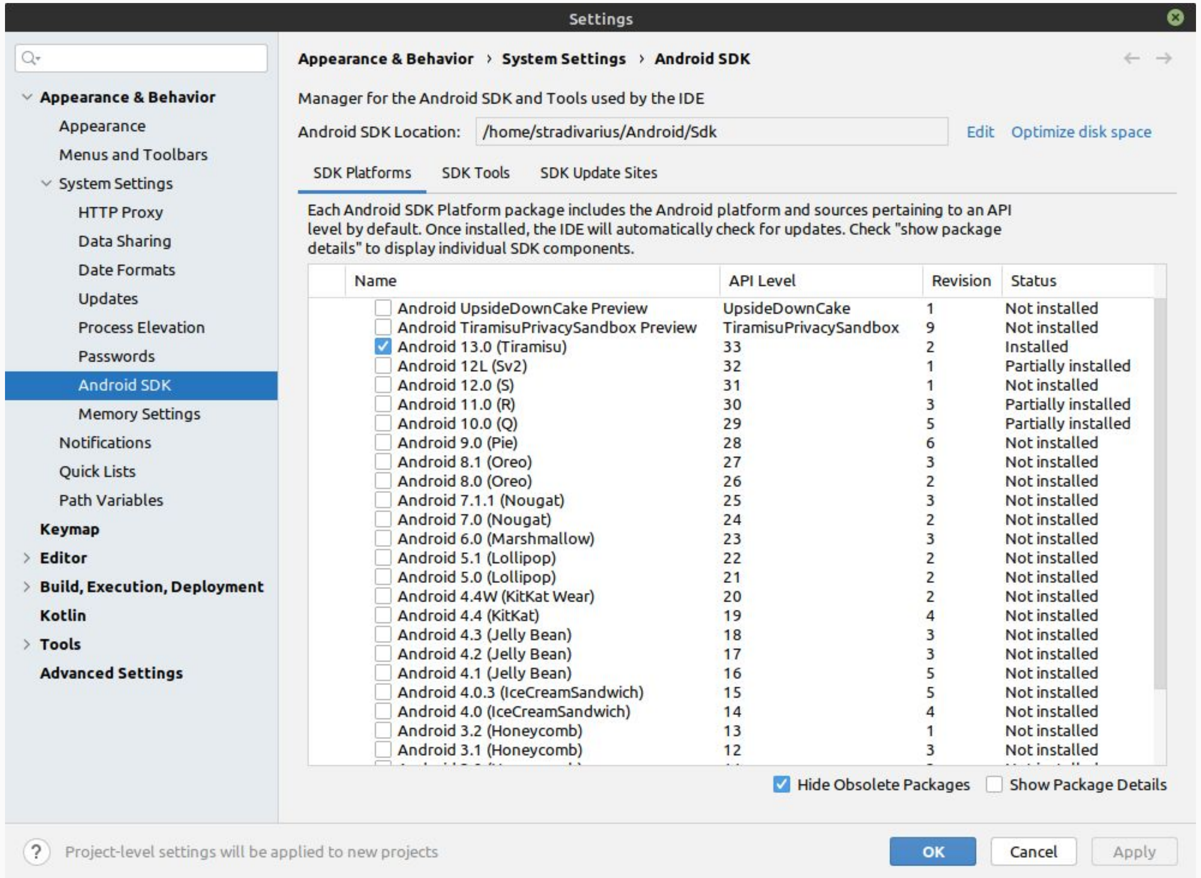
SDK Setup

Tools > SDK Manager

Android tool Used to get APIs and add-ons

You can also start it from Android Studio

It'll affect the compileSdkVersion (details later)



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: /home/stradivarius/Android/Sdk Edit Optimize disk space

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
☐ Android UpsideDownCake Preview	UpsideDownCake	1	Not installed
☐ Android TiramisuPrivacySandbox Preview	TiramisuPrivacySandbox	9	Not installed
☑ Android 13.0 (Tiramisu)	33	2	Installed
☐ Android 12L (Sv2)	32	1	Partially installed
☐ Android 12.0 (S)	31	1	Not installed
☐ Android 11.0 (R)	30	3	Partially installed
☐ Android 10.0 (Q)	29	5	Partially installed
☐ Android 9.0 (Pie)	28	6	Not installed
☐ Android 8.1 (Oreo)	27	3	Not installed
☐ Android 8.0 (Oreo)	26	2	Not installed
☐ Android 7.1.1 (Nougat)	25	3	Not installed
☐ Android 7.0 (Nougat)	24	2	Not installed
☐ Android 6.0 (Marshmallow)	23	3	Not installed
☐ Android 5.1 (Lollipop)	22	2	Not installed
☐ Android 5.0 (Lollipop)	21	2	Not installed
☐ Android 4.4W (KitKat Wear)	20	2	Not installed
☐ Android 4.4 (KitKat)	19	4	Not installed
☐ Android 4.3 (Jelly Bean)	18	3	Not installed
☐ Android 4.2 (Jelly Bean)	17	3	Not installed
☐ Android 4.1 (Jelly Bean)	16	5	Not installed
☐ Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
☐ Android 4.0 (IceCreamSandwich)	14	4	Not installed
☐ Android 3.2 (Honeycomb)	13	1	Not installed
☐ Android 3.1 (Honeycomb)	12	3	Not installed

☑ Hide Obsolete Packages ☐ Show Package Details

Project-level settings will be applied to new projects

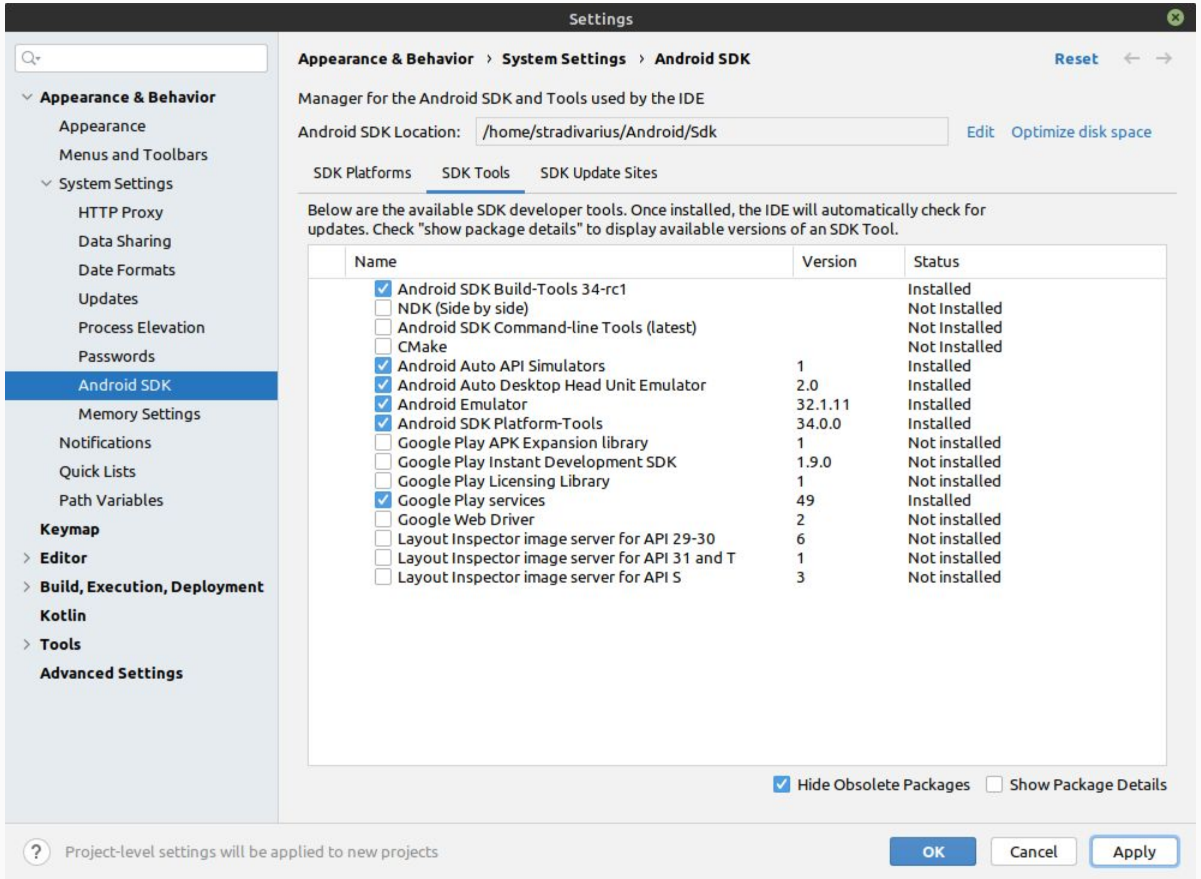
SDK Setup

Tools > SDK Manager

Android tool Used to get APIs and add-ons

You can also start it from Android Studio

It’ll affect the compileSdkVersion (details later)



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: /home/stradivarius/Android/Sdk Edit Optimize disk space

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
☑ Android SDK Build-Tools 34-rc1		Installed
□ NDK (Side by side)		Not Installed
□ Android SDK Command-line Tools (latest)		Not Installed
□ CMake		Not Installed
☑ Android Auto API Simulators	1	Installed
☑ Android Auto Desktop Head Unit Emulator	2.0	Installed
☑ Android Emulator	32.1.11	Installed
☑ Android SDK Platform-Tools	34.0.0	Installed
□ Google Play APK Expansion library	1	Not installed
□ Google Play Instant Development SDK	1.9.0	Not installed
□ Google Play Licensing Library	1	Not installed
☑ Google Play services	49	Installed
□ Google Web Driver	2	Not installed
□ Layout Inspector image server for API 29-30	6	Not installed
□ Layout Inspector image server for API 31 and T	1	Not installed
□ Layout Inspector image server for API S	3	Not installed

☑ Hide Obsolete Packages □ Show Package Details

Project-level settings will be applied to new projects

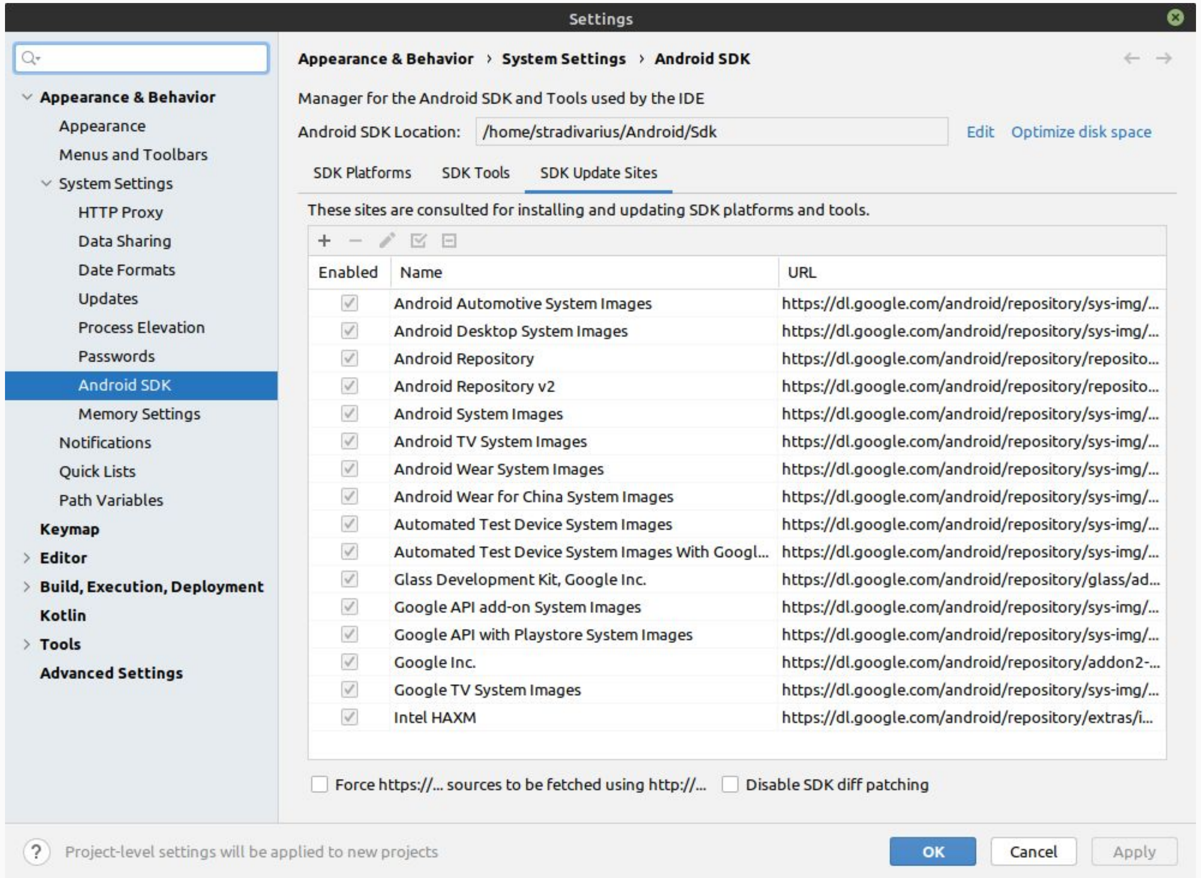
SDK Setup

Tools > SDK Manager

Android tool Used to get APIs and add-ons

You can also start it from Android Studio

It'll affect the compileSdkVersion (details later)



Settings

- Appearance & Behavior
  - Appearance
  - Menus and Toolbars
  - System Settings
    - HTTP Proxy
    - Data Sharing
    - Date Formats
    - Updates
    - Process Elevation
    - Passwords
    - Android SDK
    - Memory Settings
  - Notifications
  - Quick Lists
  - Path Variables
- Keymap
- Editor
- Build, Execution, Deployment
- Kotlin
- Tools
- Advanced Settings

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: /home/stradivarius/Android/Sdk Edit Optimize disk space

SDK Platforms | SDK Tools | SDK Update Sites

These sites are consulted for installing and updating SDK platforms and tools.

Enabled	Name	URL
☑	Android Automotive System Images	https://dl.google.com/android/repository/sys-img/...
☑	Android Desktop System Images	https://dl.google.com/android/repository/sys-img/...
☑	Android Repository	https://dl.google.com/android/repository/reposito...
☑	Android Repository v2	https://dl.google.com/android/repository/reposito...
☑	Android System Images	https://dl.google.com/android/repository/sys-img/...
☑	Android TV System Images	https://dl.google.com/android/repository/sys-img/...
☑	Android Wear System Images	https://dl.google.com/android/repository/sys-img/...
☑	Android Wear for China System Images	https://dl.google.com/android/repository/sys-img/...
☑	Automated Test Device System Images	https://dl.google.com/android/repository/sys-img/...
☑	Automated Test Device System Images With Googl...	https://dl.google.com/android/repository/sys-img/...
☑	Glass Development Kit, Google Inc.	https://dl.google.com/android/repository/glass/ad...
☑	Google API add-on System Images	https://dl.google.com/android/repository/sys-img/...
☑	Google API with Playstore System Images	https://dl.google.com/android/repository/sys-img/...
☑	Google Inc.	https://dl.google.com/android/repository/addon2-...
☑	Google TV System Images	https://dl.google.com/android/repository/sys-img/...
☑	Intel HAXM	https://dl.google.com/android/repository/extras/i...

□ Force https://... sources to be fetched using http://... □ Disable SDK diff patching

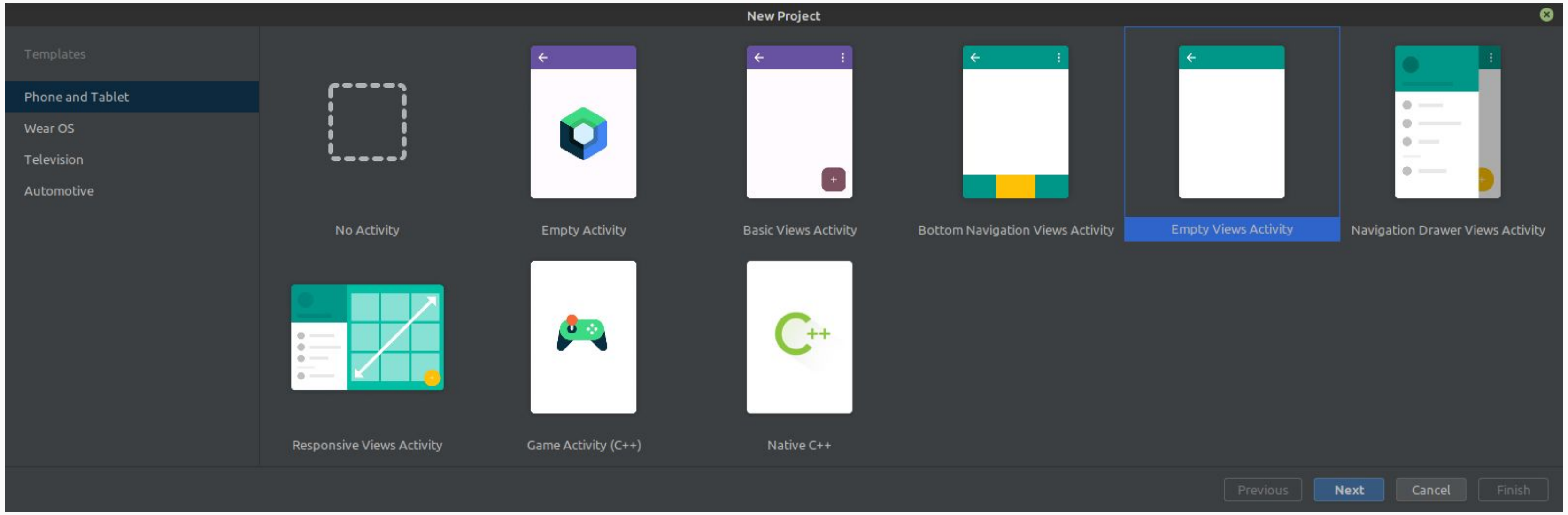
Project-level settings will be applied to new projects

OK Cancel Apply

Hello World App

Go to File > New Project

Newest version of Android Studio makes you choose first which kind of activity you want to start with.

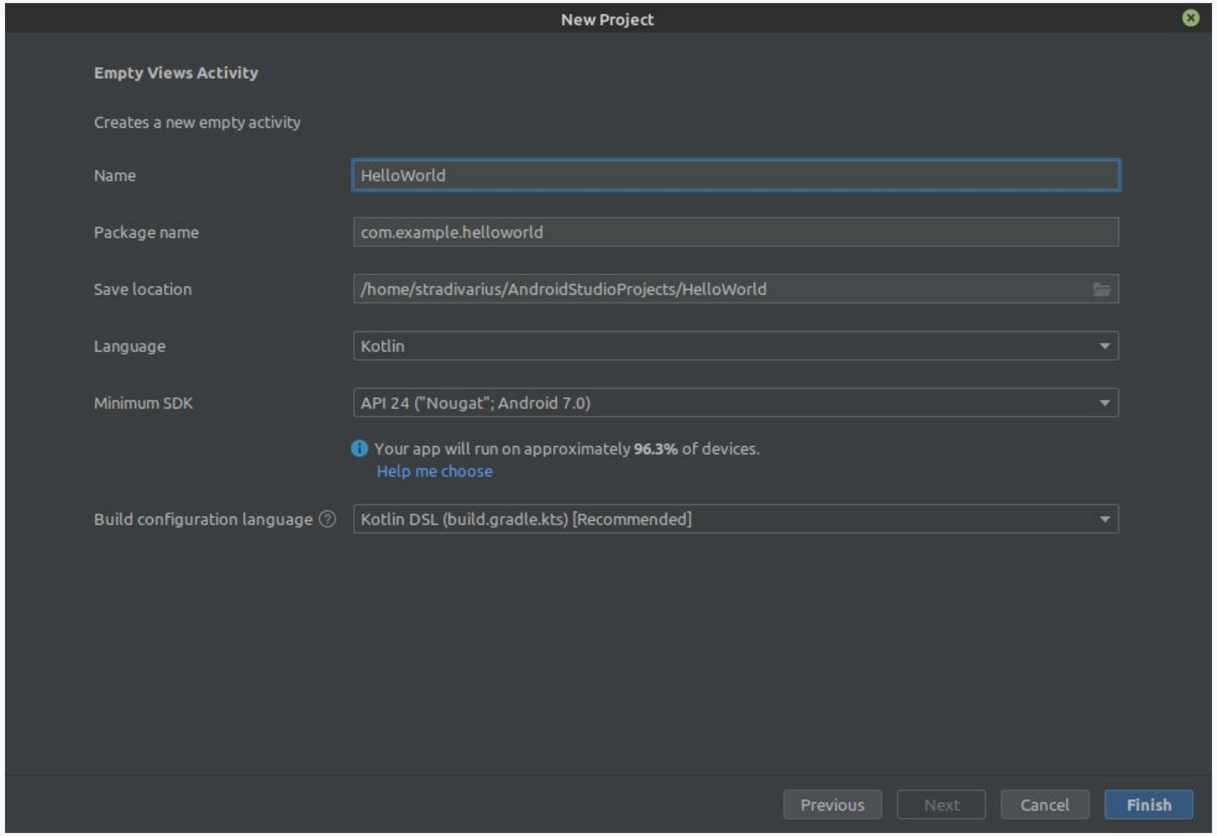


Hello World App

Go to File > New Project

Your activity will be named MainActivity by default (Java class).

Of course you can refactor it ...



Empty Views Activity

Creates a new empty activity

Name: HelloWorld

Package name: com.example.helloworld

Save location: /home/stradivarius/AndroidStudioProjects/HelloWorld

Language: Kotlin

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Your app will run on approximately **96.3%** of devices.
Help me choose

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

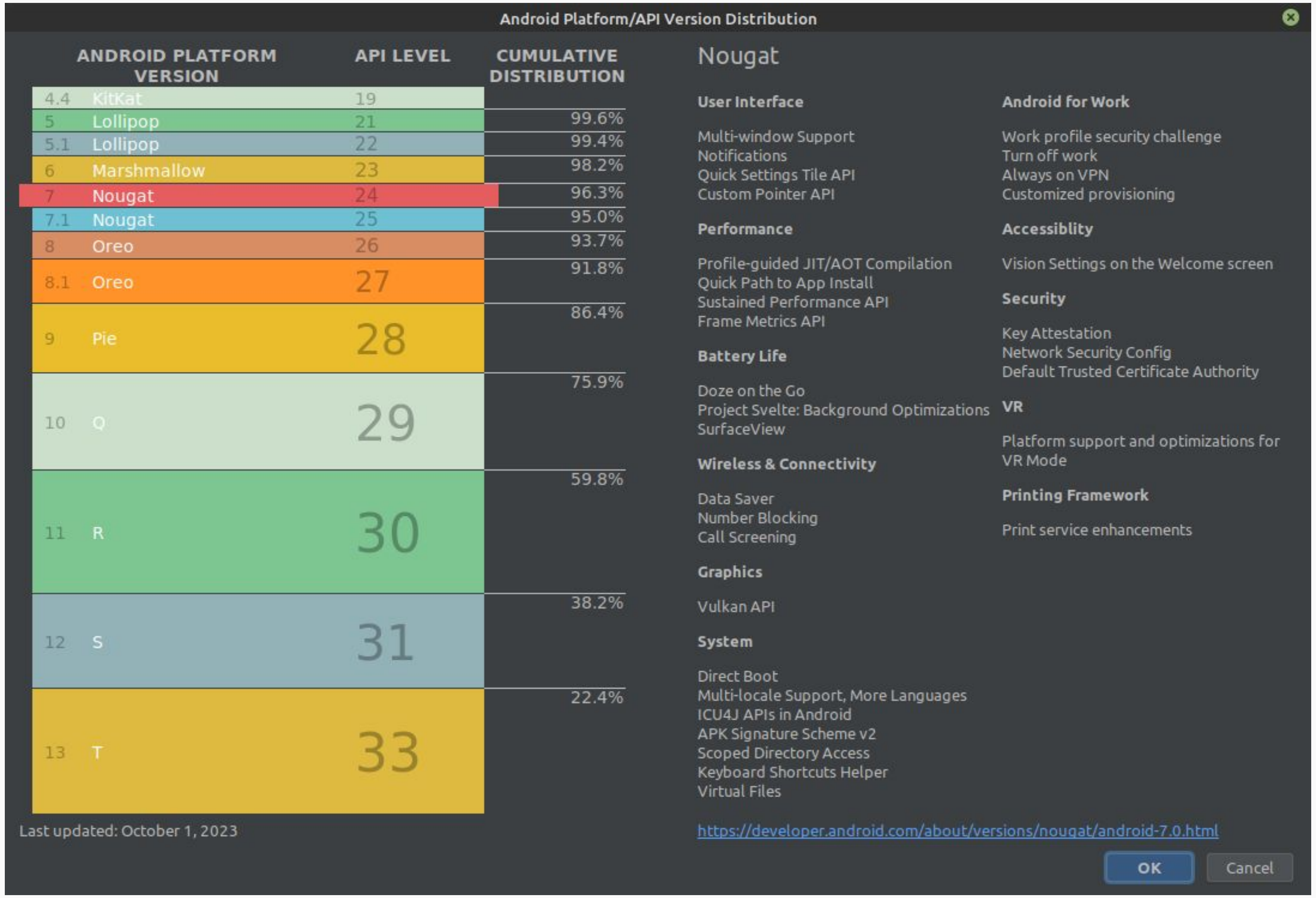
Hello World App

Go to File > New Project

Choose carefully which API version to use.

Low version =
high compatibility

Low version =
less features



Android Platform/API Version Distribution

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.4	KitKat	19	
5	Lollipop	21	99.6%
5.1	Lollipop	22	99.4%
6	Marshmallow	23	98.2%
7	Nougat	24	96.3%
7.1	Nougat	25	95.0%
8	Oreo	26	93.7%
8.1	Oreo	27	91.8%
9	Pie	28	86.4%
10	Q	29	75.9%
11	R	30	59.8%
12	S	31	38.2%
13	T	33	22.4%

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

Last updated: October 1, 2023

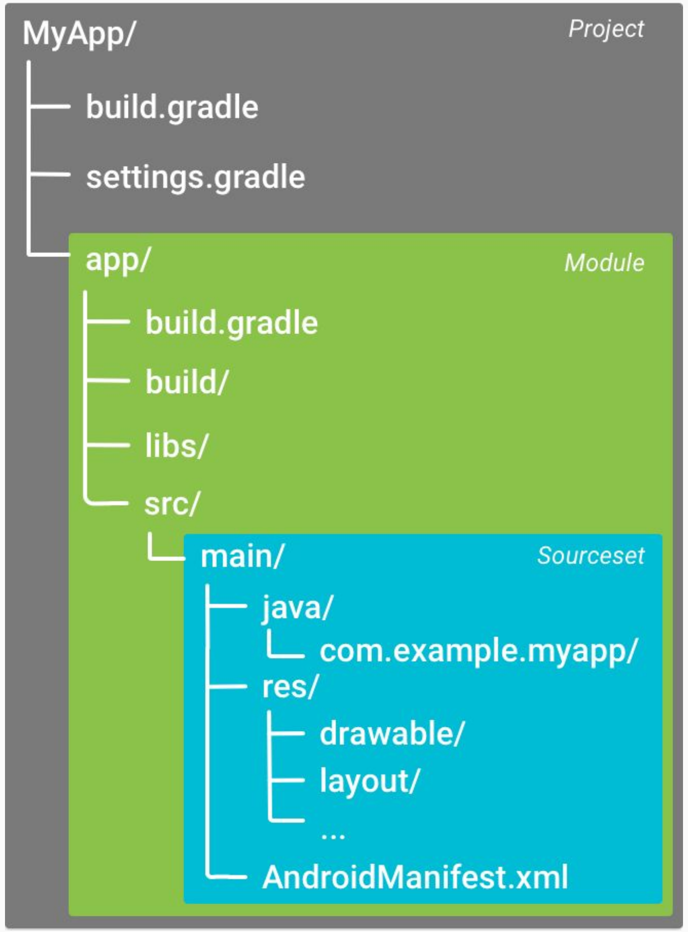
https://developer.android.com/about/versions/nougat/android-7.0.html

Hello World App

“Whatever you do in IntelliJ IDEA, you do that in the context of a project. A project is an organizational unit that represents a complete software solution.

Your finished product may be decomposed into a series of discrete, isolated modules, but it's a project definition that brings them together and ties them into a greater whole.”

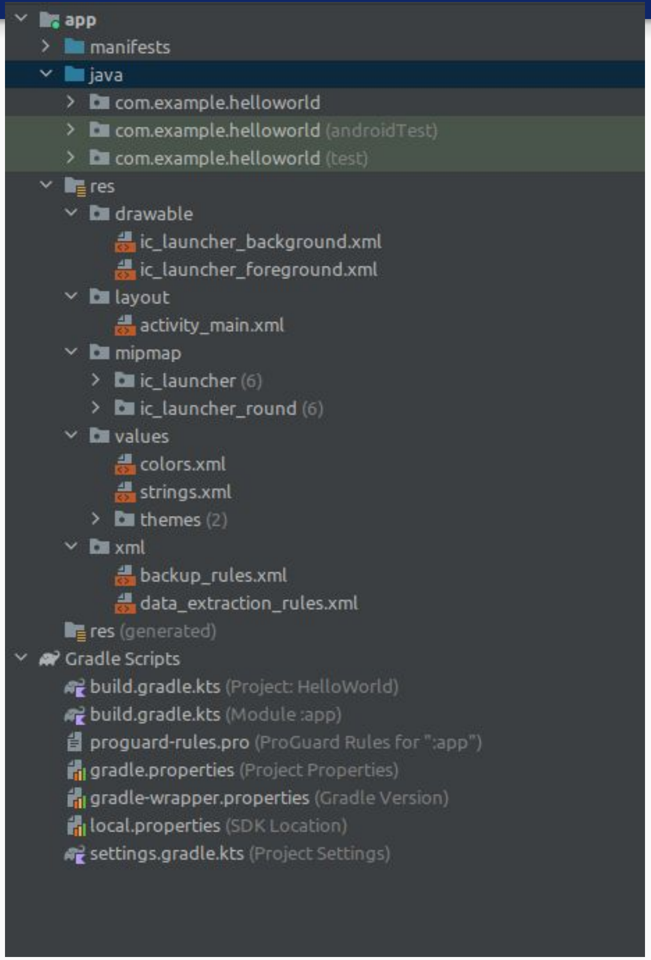
This means that in theory you can develop more that one app within the same project but you will hardly want to do so.



Hello World App

The code of your application:

- The java folder contains… the java code!
  - … or Kotlin actually
- Inside res there are a lot of resources
  - Images
  - Layouts
  - Xml files
  - Strings
- AndroidManifest.xml



```
app
  manifests
  java
    com.example.helloworld
    com.example.helloworld (androidTest)
    com.example.helloworld (test)
  res
    drawable
      ic_launcher_background.xml
      ic_launcher_foreground.xml
    layout
      activity_main.xml
    mipmap
      ic_launcher (6)
      ic_launcher_round (6)
    values
      colors.xml
      strings.xml
      themes (2)
    xml
      backup_rules.xml
      data_extraction_rules.xml
  res (generated)
Gradle Scripts
  build.gradle.kts (Project: HelloWorld)
  build.gradle.kts (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  local.properties (SDK Location)
  settings.gradle.kts (Project Settings)
```

Hello World App

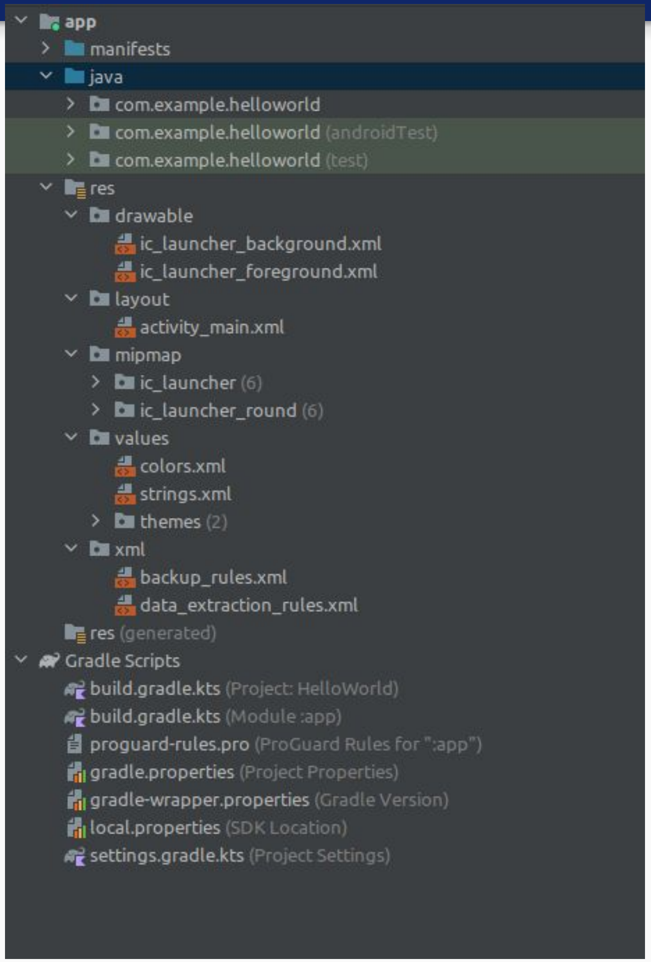
<u>**Android Manifest**</u>

the only file exposed to the OS

Mandatory file for every application

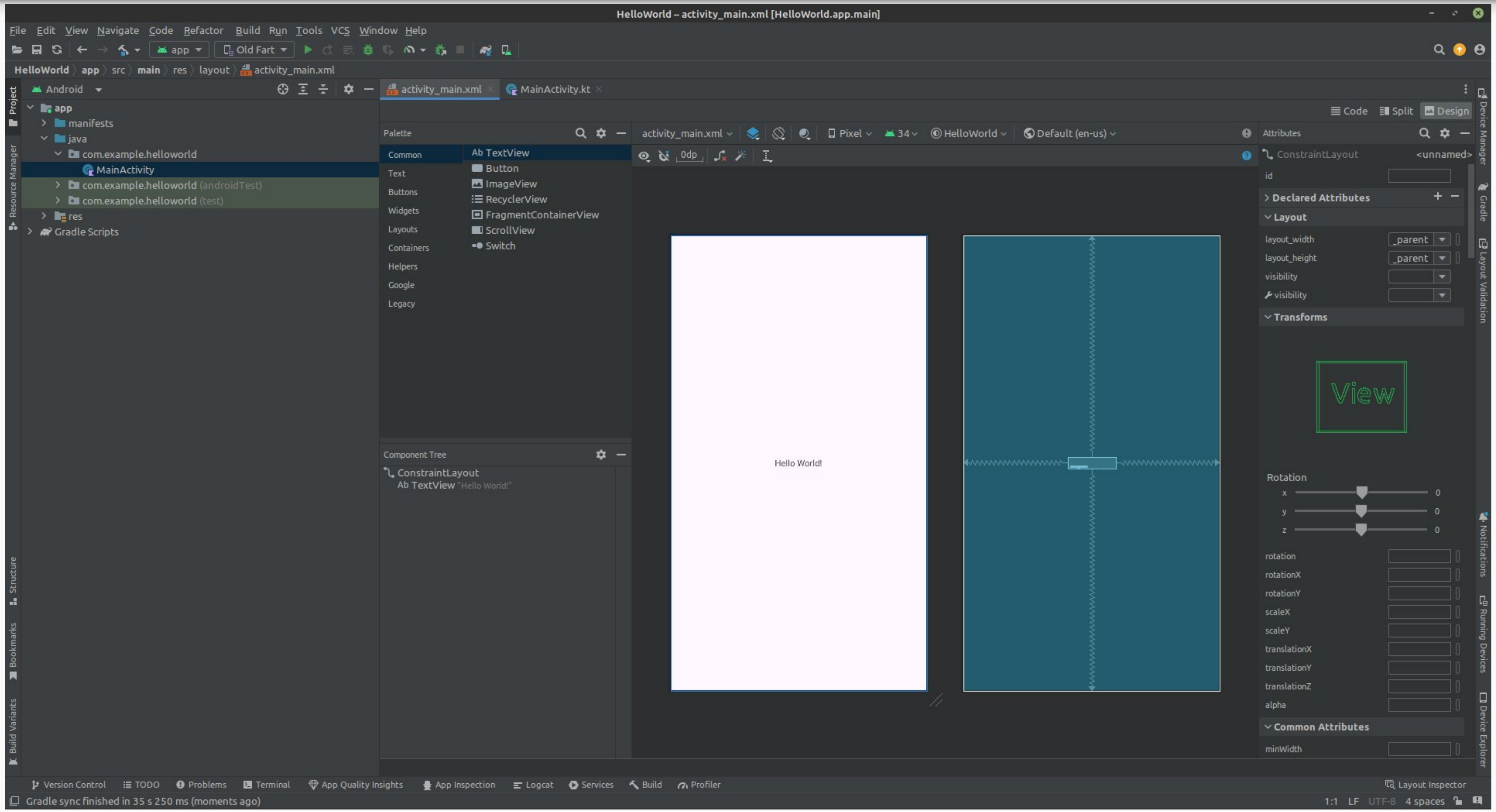
Contains:

- Application declaration
- Permissions
- Intent filters
- Targets



```
app
  manifests
  java
    com.example.helloworld
    com.example.helloworld (androidTest)
    com.example.helloworld (test)
  res
    drawable
      ic_launcher_background.xml
      ic_launcher_foreground.xml
    layout
      activity_main.xml
    mipmap
      ic_launcher (6)
      ic_launcher_round (6)
    values
      colors.xml
      strings.xml
      themes (2)
    xml
      backup_rules.xml
      data_extraction_rules.xml
  res (generated)
Gradle Scripts
  build.gradle.kts (Project: HelloWorld)
  build.gradle.kts (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  local.properties (SDK Location)
  settings.gradle.kts (Project Settings)
```

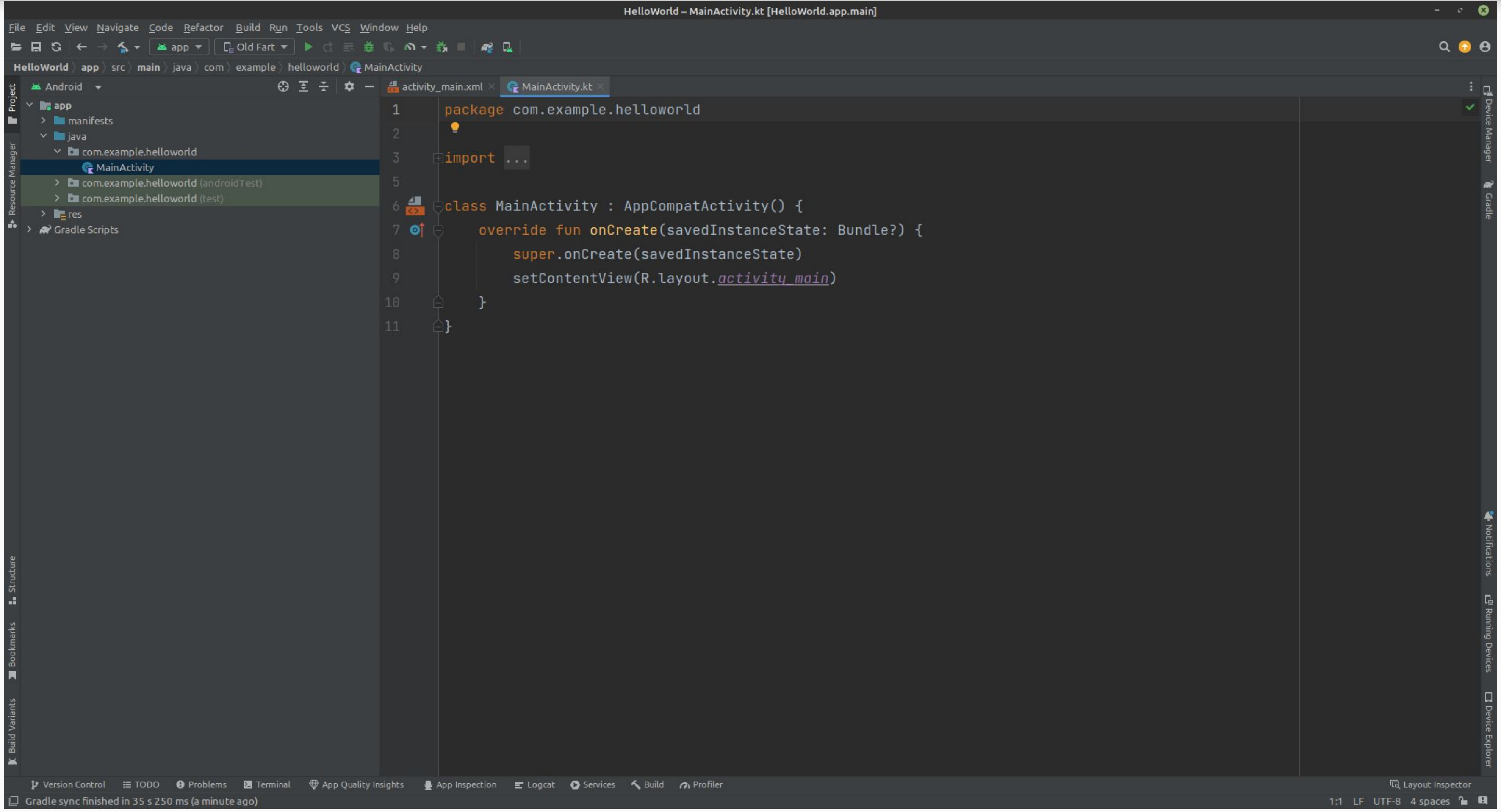
Hello World App



HelloWorld – activity_main.xml [HelloWorld.app.main]

Gradle sync finished in 35 s 250 ms (moments ago)

Hello World App



HelloWorld – MainActivity.kt [HelloWorld.app.main]

```kotlin
package com.example.helloworld

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Gradle sync finished in 35 s 250 ms (a minute ago)

Running the App

Hit Tools >

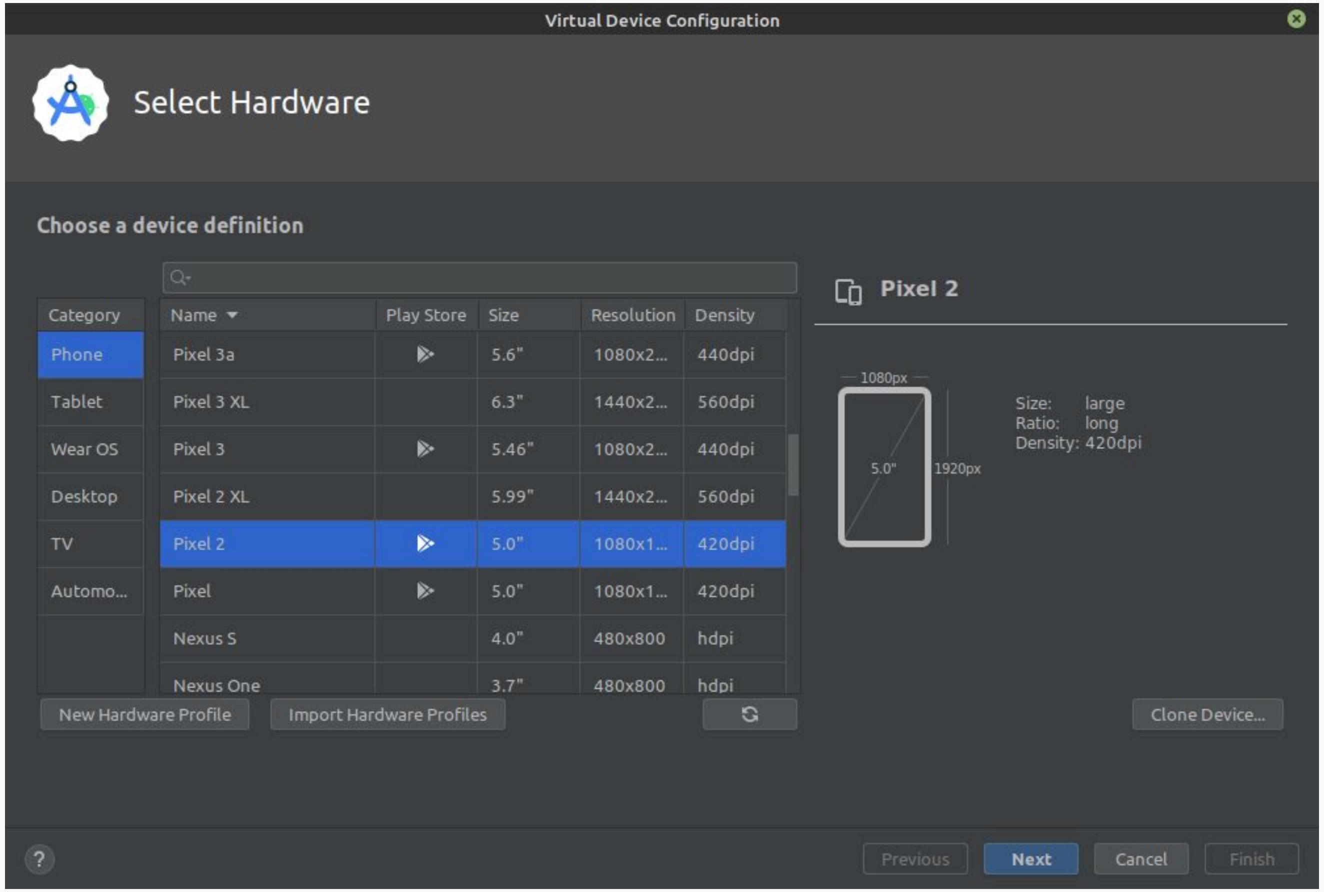
Device Manager

Previously AVD

AVD = Android Virtual Device

You can select options for the emulator

You can create as many as you want



Virtual Device Configuration

Select Hardware

Choose a device definition

Category
Phone
Tablet
Wear OS
Desktop
TV
Automo...

Name ▾	Play Store	Size	Resolution	Density
Pixel 3a	▶	5.6"	1080x2...	440dpi
Pixel 3 XL		6.3"	1440x2...	560dpi
Pixel 3	▶	5.46"	1080x2...	440dpi
Pixel 2 XL		5.99"	1440x2...	560dpi
Pixel 2	▶	5.0"	1080x1...	420dpi
Pixel	▶	5.0"	1080x1...	420dpi
Nexus S		4.0"	480x800	hdpi
Nexus One		3.7"	480x800	hdpi

Pixel 2

Size: large
Ratio: long
Density: 420dpi

Running the App

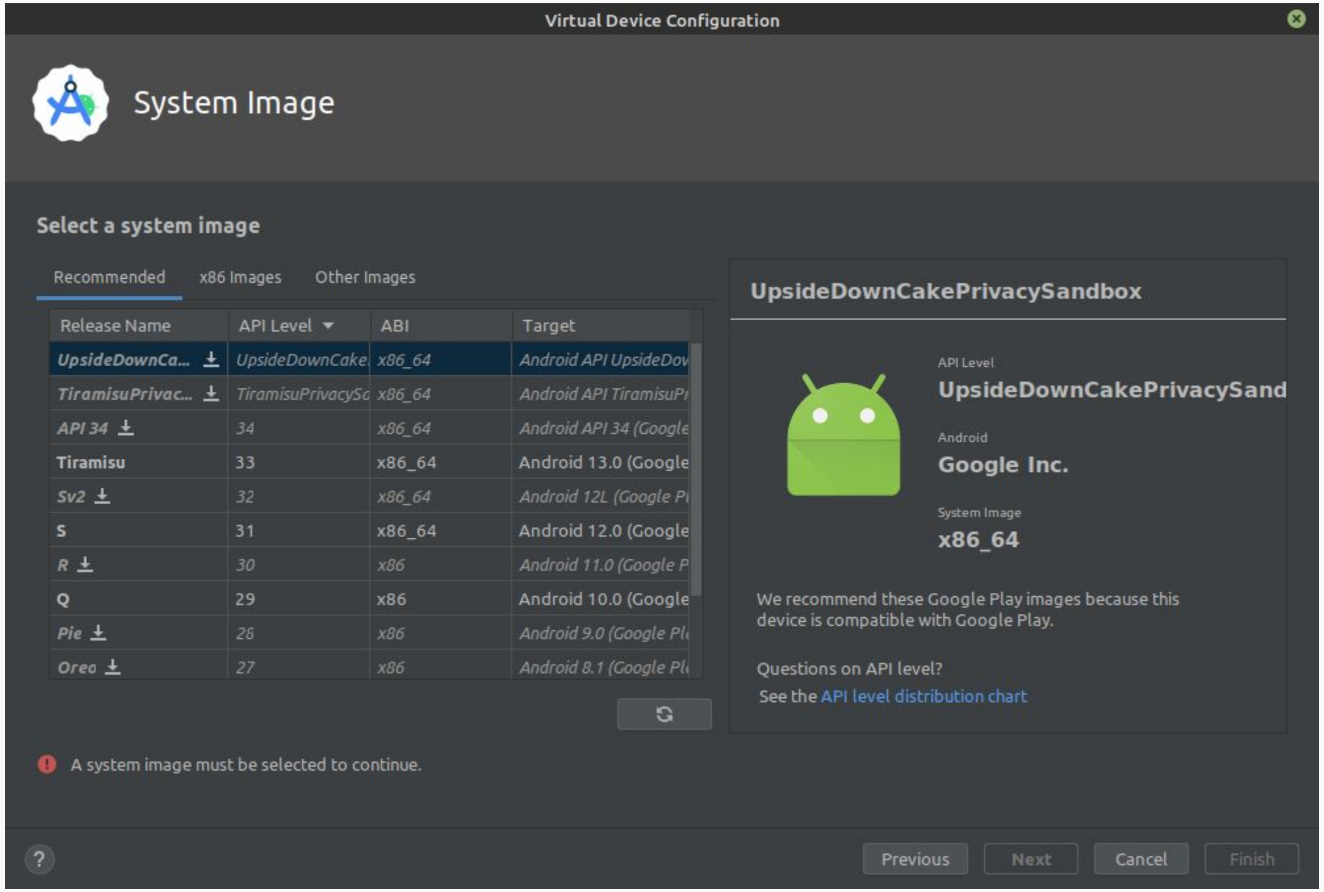
Hit Tools >

Device Manager

Previously AVD

AVD = Android Virtual Device

You obviously need to download the Android system image for the version you want.



System Image

Select a system image

Release Name	API Level ▼	ABI	Target
UpsideDownCa...	*UpsideDownCake*	*x86_64*	*Android API UpsideDov*
TiramisuPrivac...	*TiramisuPrivacySa*	*x86_64*	*Android API TiramisuPı*
API 34	*34*	*x86_64*	*Android API 34 (Google*
Tiramisu	33	x86_64	Android 13.0 (Google
Sv2	*32*	*x86_64*	*Android 12L (Google Pl*
S	31	x86_64	Android 12.0 (Google
R	*30*	*x86*	*Android 11.0 (Google P*
Q	29	x86	Android 10.0 (Google
Pie	*28*	*x86*	*Android 9.0 (Google Pl*
Oreo	*27*	*x86*	*Android 8.1 (Google Pl*

UpsideDownCakePrivacySandbox

API Level
UpsideDownCakePrivacySand

Android
Google Inc.

System Image
x86_64

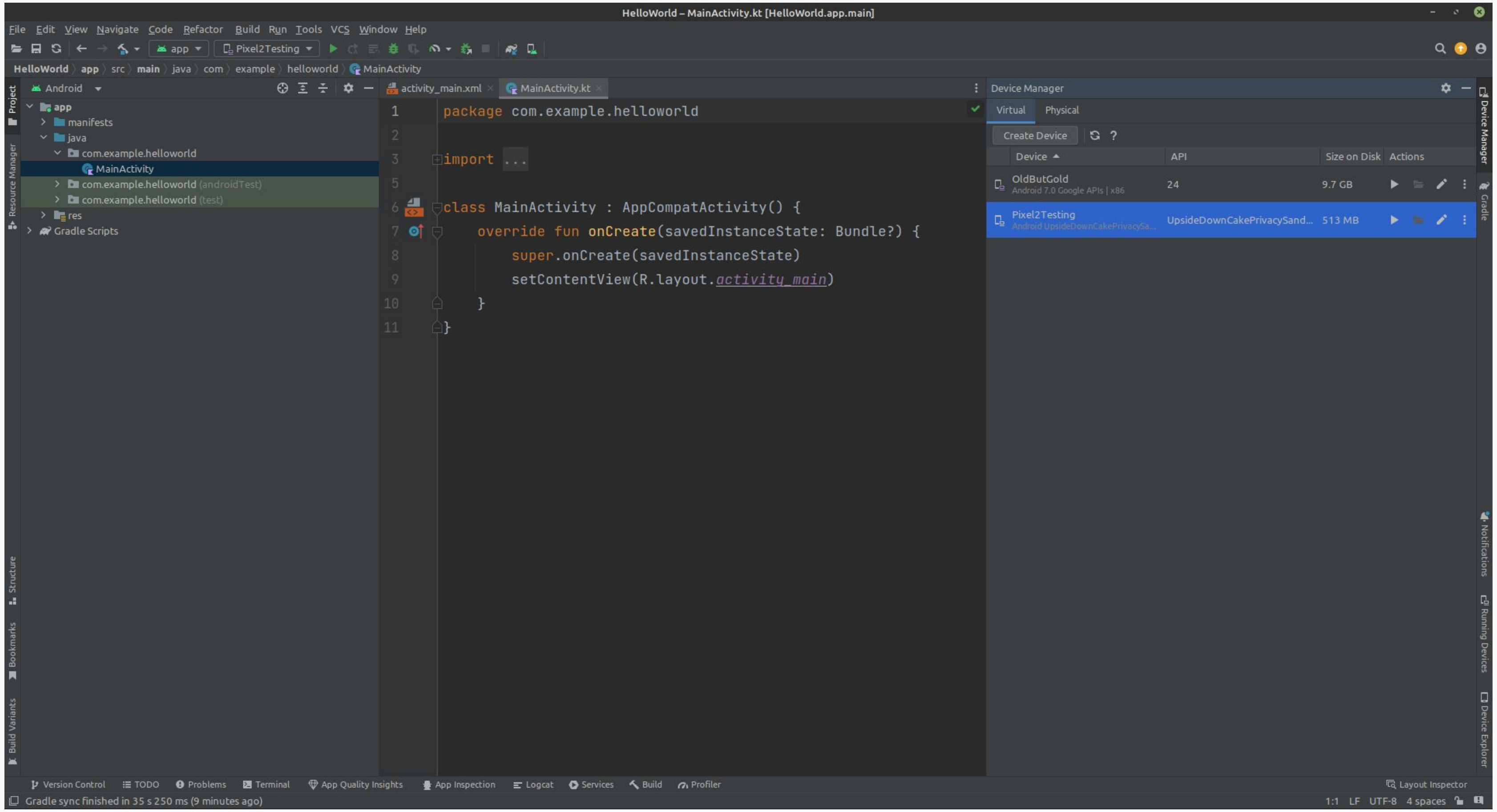
We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
See the API level distribution chart

A system image must be selected to continue.

Running the App

Hit the play button for testing!



HelloWorld – MainActivity.kt [HelloWorld.app.main]

```
package com.example.helloworld

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Device ▲	API	Size on Disk	Actions
OldButGold Android 7.0 Google APIs | x86	24	9.7 GB	▶
Pixel2Testing Android UpsideDownCakePrivacySa...	UpsideDownCakePrivacySand...	513 MB	▶

Gradle sync finished in 35 s 250 ms (9 minutes ago)

1:1 LF UTF-8 4 spaces

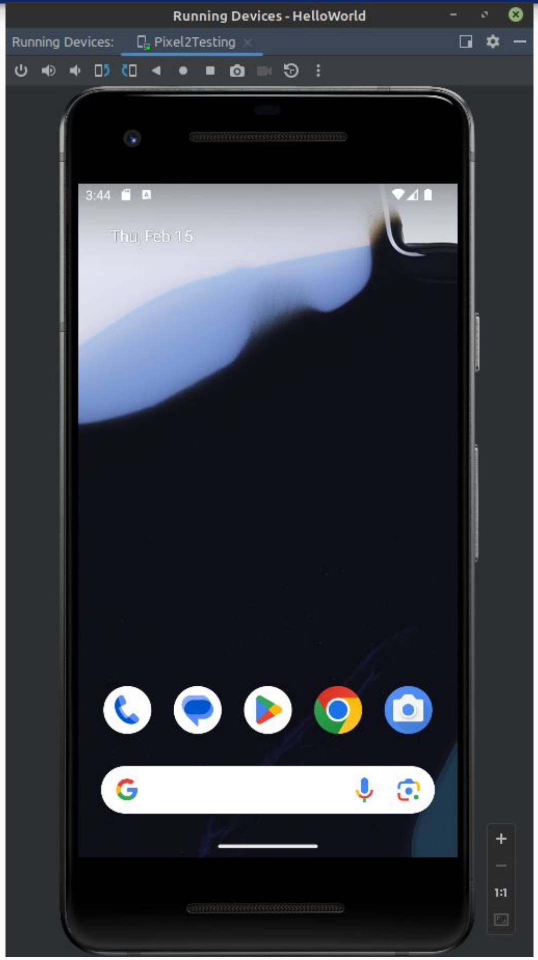
Running the App

If you run it, it will turn on and resemble pretty much a real device.

It is legit a virtual machine.

YES you can use the internet (by default each AVD is individually NATted).

NO you can't call (lol)

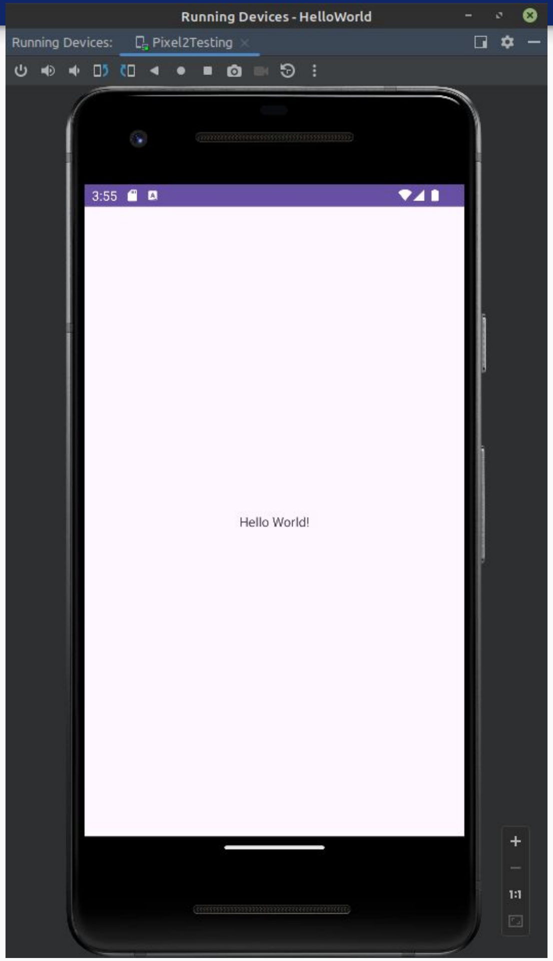


Running the App

Hit Run > Run ‘app’

Test in on the emulator

You should see something similar to this



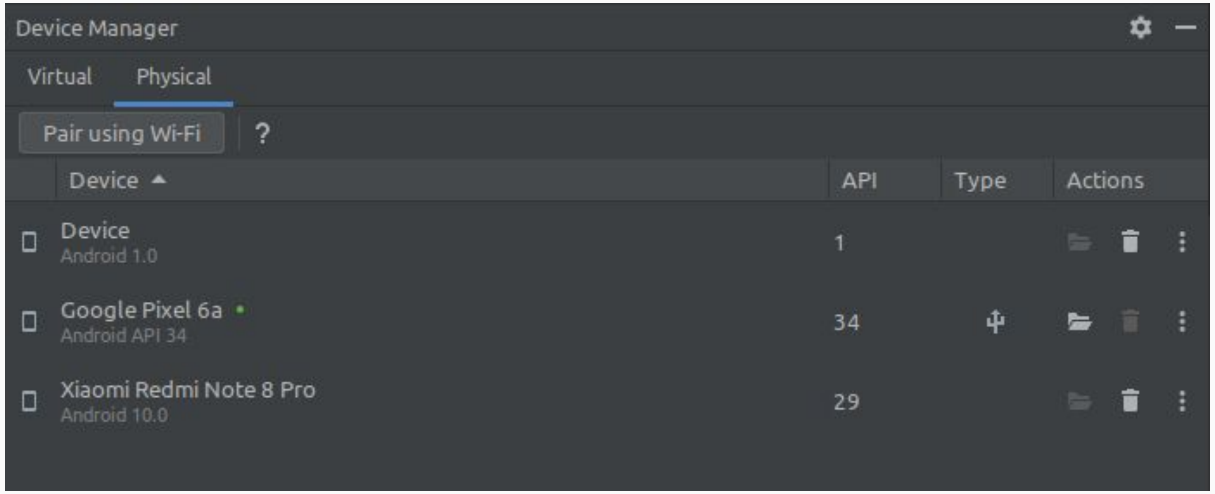
Running the App

You can also test the app on a real device: there are two ways to do it:

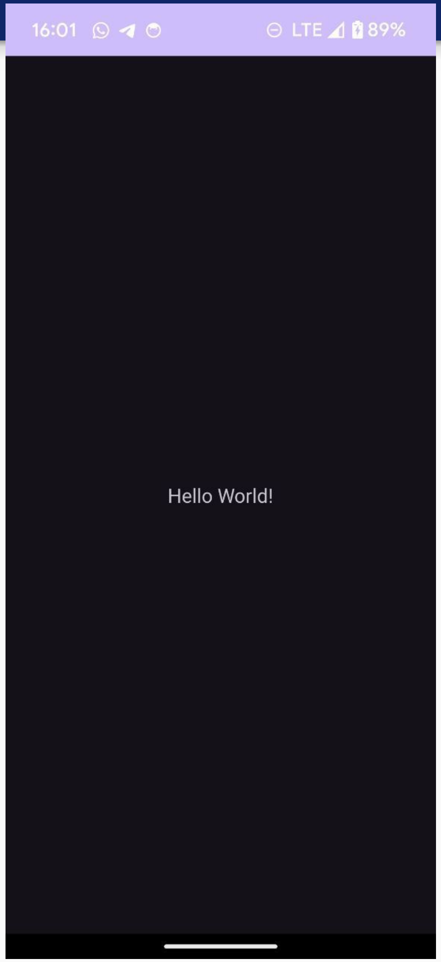
- Via USB debugging (more for debugging)
  - Phone must have developer options and USB debugging enabled (https://developer.android.com/studio/debug/dev-options.html#enable)
  - PC's OS must have the correct driver/module (https://developer.android.com/studio/run/device#setting-up)
  - You can then run apps just by hitting the Run > Run 'app'
  - You can use the newest WiFi pairing!
- Create an ***apk*** (more for releasing and sharing)
  - Must be signed

Running the App

Here’s the app running on my phone



Device	API	Type	Actions
Device Android 1.0	1		
Google Pixel 6a ● Android API 34	34	⚲	
Xiaomi Redmi Note 8 Pro Android 10.0	29		



```
Hello World!
```

Versioning

Hit “File > Project Structure > Modules” and you’ll see two sections that’ll help you deal with versioning.

Properties

versioning and other stuff at Compile time

- Mostly having to do with Gradle

Default Config

versioning and other stuff at Run Time

Versioning

What is Gradle?

It’s the official build automation tool for Android, coming with a lot of optimizations.

It has config files that can be modified through the Android Studio GUI.

It manages build configurations that no longer are assigned to the Android Developer.

Versioning

Gradle features

- Build types
  - Release, debug, etc…
- Product Flavors
  - Free and paid versions…
- Manifest Entries
  - Override values on the manifest files
- Dependencies
  - Reference to libraries to import (Maven style) that are not included by default in the Android build.
- Signing
  - Configuration for signing your app during the build process…
- …

Versioning

compileSdkVersion

- Used by Gradle to compile the project
- i.e. which set of classes and functions should I use?
- It’s the newest possible SDK theoretically supported by your app (watch out, NOT the API).
- Suggested to use the latest available (unless you haven’t learned it yet…)
- It’s COMPILED, therefore retro-compatibility is structurally ensured.

Versioning

minSdkVersion

- Indicates which is the oldest release of the SDK (but also API) your app is compatible with...
- … though it is compiled with another version.
- Obviously you cannot implement certain functionalities (e.g. channels).
- In practice if a customer has a phone that’s too old, then the app is neither installable nor visible.

Versioning

targetSdkVersion

- Indicates which is the newest release of the SDK (but also API) your app is compatible with...
- … in practice it tells what is the expected version.
- It is ideally the same as the compileSdkVersion, however it can be older if newer versions had not been tested.

In short:

minSdkVersion <= targetSdkVersion <= compileSdkVersion

Even though it’s better:

minSdkVersion <= targetSdkVersion == compileSdkVersion

Deploying

Android applications must be signed before installing them on a real device.
Hit: Build >
Generate Signed Bundle / APK

You can generate a Bundle (ABB) alternatively:

a Bundle is Google Play's new app serving model, called Dynamic Delivery, then uses your app bundle to generate and serve optimized APKs for each user's device configuration, so they download only the code and resources they need to run your app. You no longer have to build, sign, and manage multiple APKs to support different devices, and users get smaller, more optimized downloads. It is a **publishing** format.

https://developer.android.com/guide/app-bundle

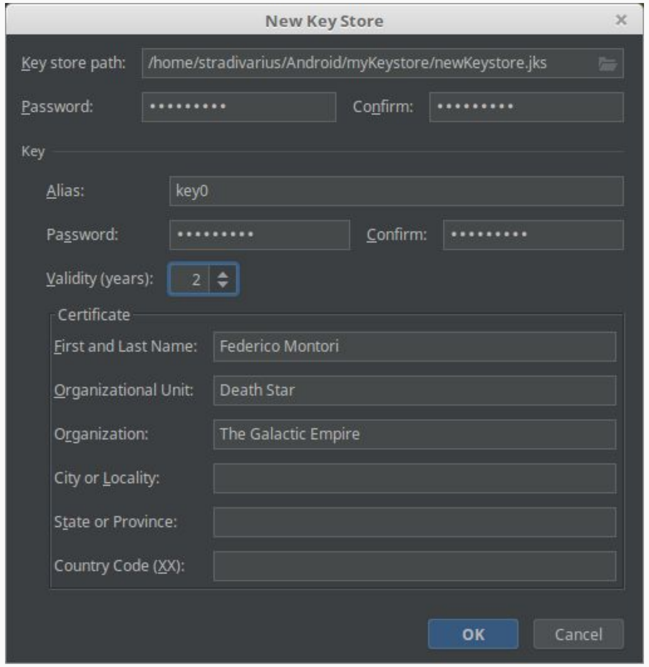
You need a key for this and you can generate one from the menu.

You can potentially use no key, but it will generate a debug version.

Deploying

Using here V2 Signature (faster, since Android 7.0)

https://developer.android.com/about/versions/nougat/android-7.0.html#apk_signature_v2



New Key Store

Key store path: /home/stradivarius/Android/myKeystore/newKeystore.jks

Password: ••••••••• Confirm: •••••••••

Key

Alias: key0

Password: ••••••••• Confirm: •••••••••

Validity (years): 2

Certificate

First and Last Name: Federico Montori

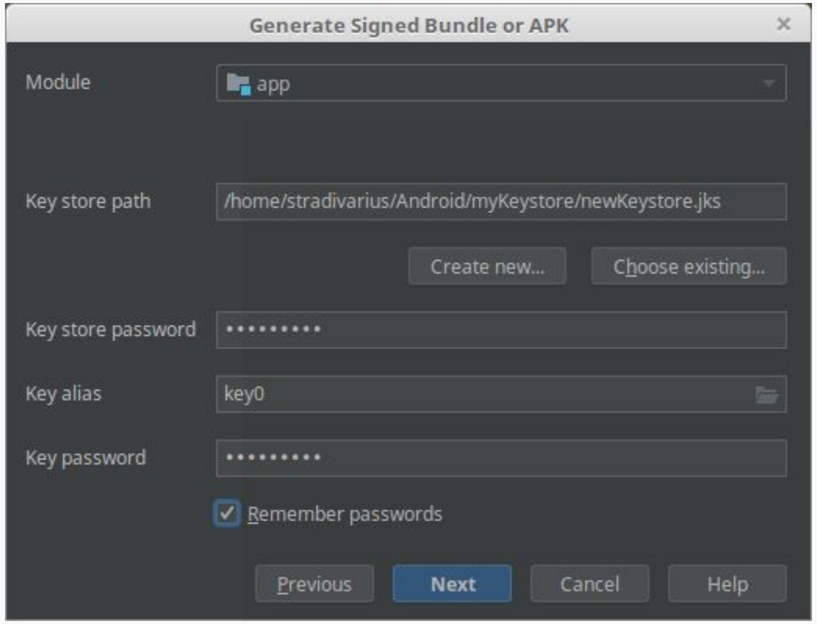
Organizational Unit: Death Star

Organization: The Galactic Empire

City or Locality:

State or Province:

Country Code (XX):



Generate Signed Bundle or APK

Module: app

Key store path: /home/stradivarius/Android/myKeystore/newKeystore.jks

Key store password: •••••••••

Key alias: key0

Key password: •••••••••

☑ Remember passwords

… transfer the .apk file to your phone and you’re done.

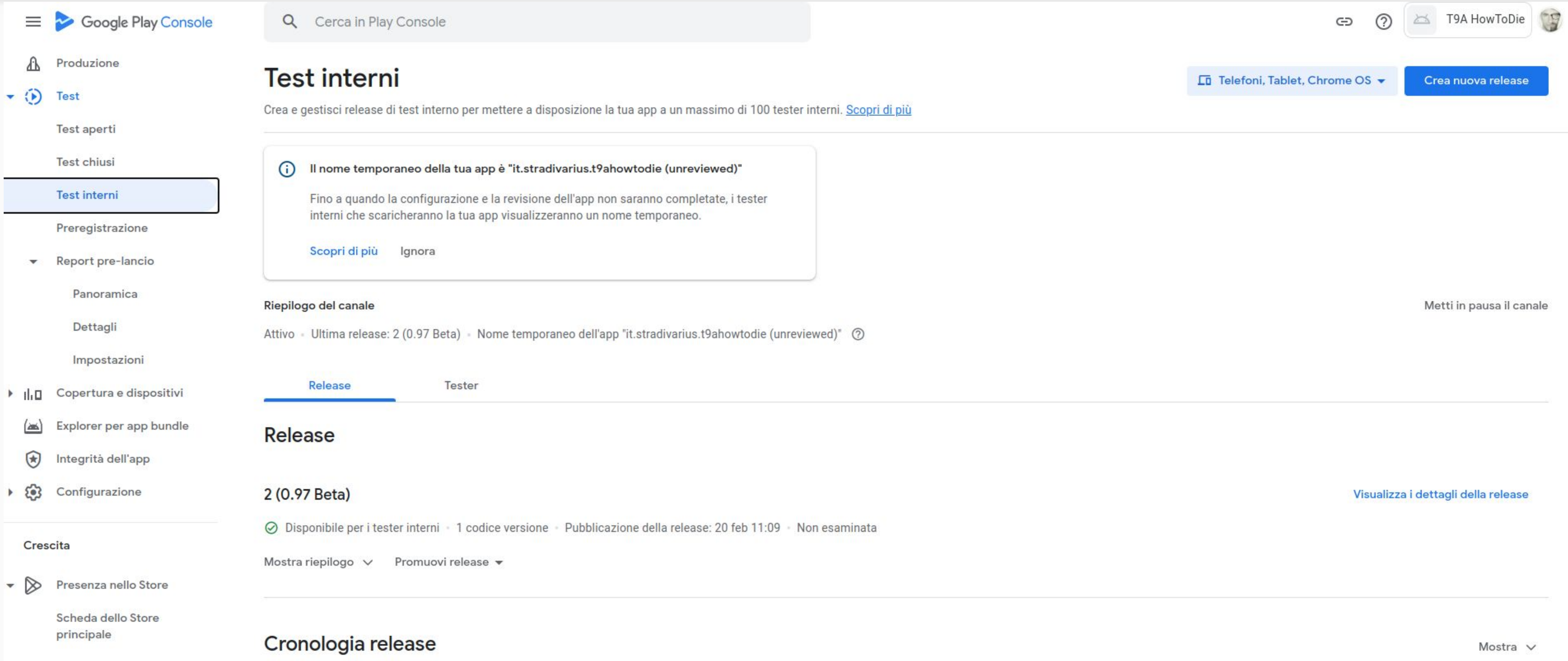
Deploying

To be published on the market, you have to pay 25 $

- Lifetime fee, unlimited APPs
- Not required for the LAM class
- Upload the ABB, and in few hours/days the APP is on the play store
    - Since a few years the APK is not accepted anymore.
- Receive comments, improve, update
- Smartphone specific bugs? AVDs

Deploying

https://play.google.com/console



Test interni

Crea e gestisci release di test interno per mettere a disposizione la tua app a un massimo di 100 tester interni. Scopri di più

Il nome temporaneo della tua app è "it.stradivarius.t9ahowtodie (unreviewed)"

Fino a quando la configurazione e la revisione dell'app non saranno completate, i tester interni che scaricheranno la tua app visualizzeranno un nome temporaneo.

Scopri di più Ignora

Riepilogo del canale Metti in pausa il canale

Attivo • Ultima release: 2 (0.97 Beta) • Nome temporaneo dell'app "it.stradivarius.t9ahowtodie (unreviewed)"

Release Tester

Release

2 (0.97 Beta) Visualizza i dettagli della release

✓ Disponibile per i tester interni • 1 codice versione • Pubblicazione della release: 20 feb 11:09 • Non esaminata

Mostra riepilogo Promuovi release

Cronologia release

Mostra

Deploying

- **Privacy Policy**
- **A lot of claims about what you are going to do with user data…**
- **Internal Tests**
  - a.k.a. distribute the apk with a handful of friends
- **Closed Tests (Mandatory)**
  - create a distribution list via the dashboard of at least 20 ppl
- **Open Tests**
  - every time you update you can pre-release the test version

Questions?

federico.montori2@unibo.it