Android course preliminaries

The Android initiation course starts soon. To get off on the right foot, follow this thorough procedure in order to have the required software downloaded and installed correctly.
Because this process involves quite a lot of download and time spent bootstrapping things, it is imperative that you do all this before the first lesson.

1° The JAVA machine

Developing on Android requires a JAVA machine. More specifically the Java Development Kit (JDK) 8 from Oracle.

- Sun Java SE Development Kit 8 (get version 8u191)
- https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Note: 64b versions are fine if your system can handle it.

➔ Download and install JDK 8 for your system.

2° The IDE

The IDE we are going to use, and also the one Google recommends, is Android Studio.

- Android Studio IDE (current version is 3.2.1 rev 181.5540.7.32.5056338)
- https://developer.android.com/studio/

The IDE is available for Windows, Mac OSX and Linux so pick the one that suits your system.

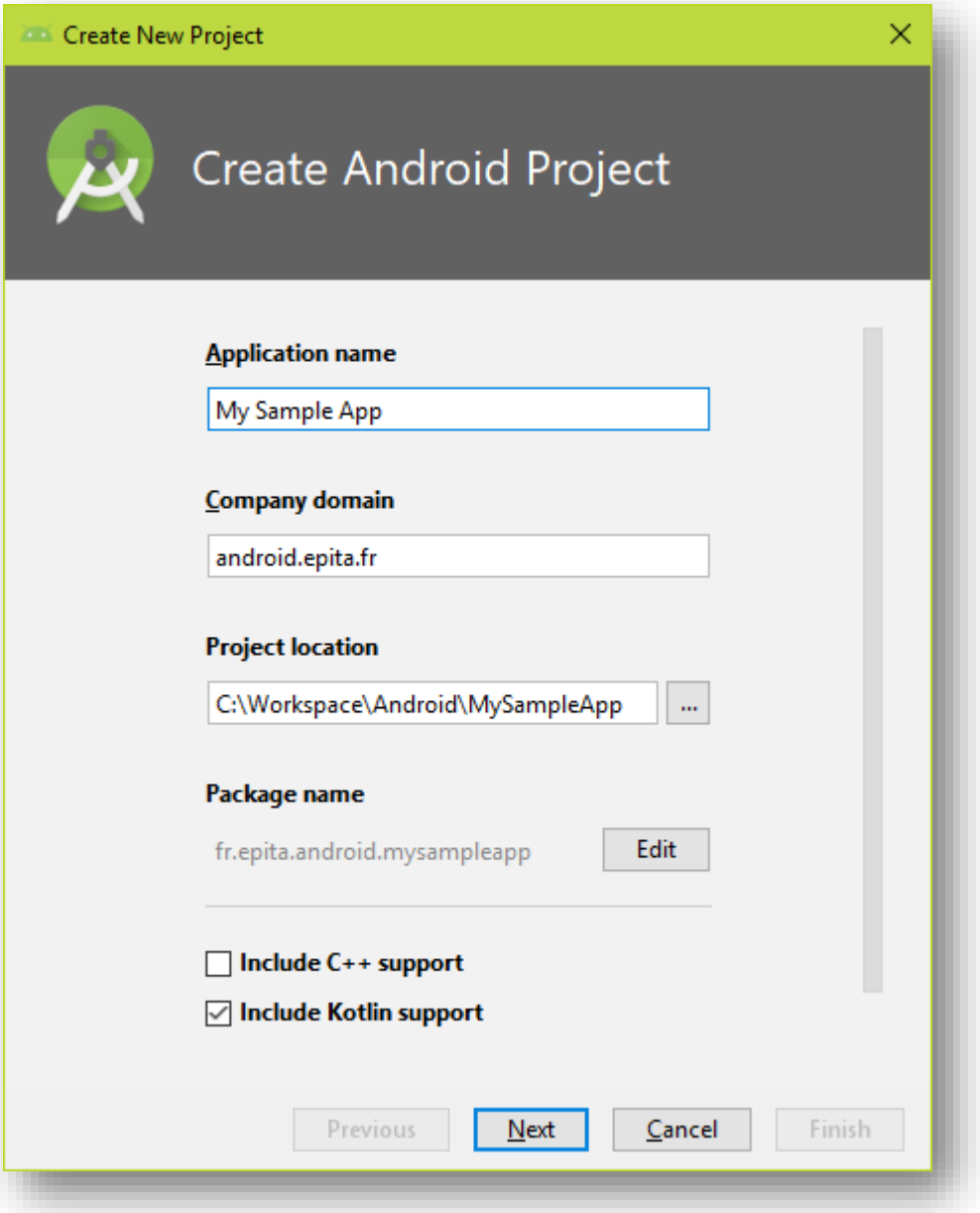
➔ Download and install Android Studio for your system.

3° SDK and other build components

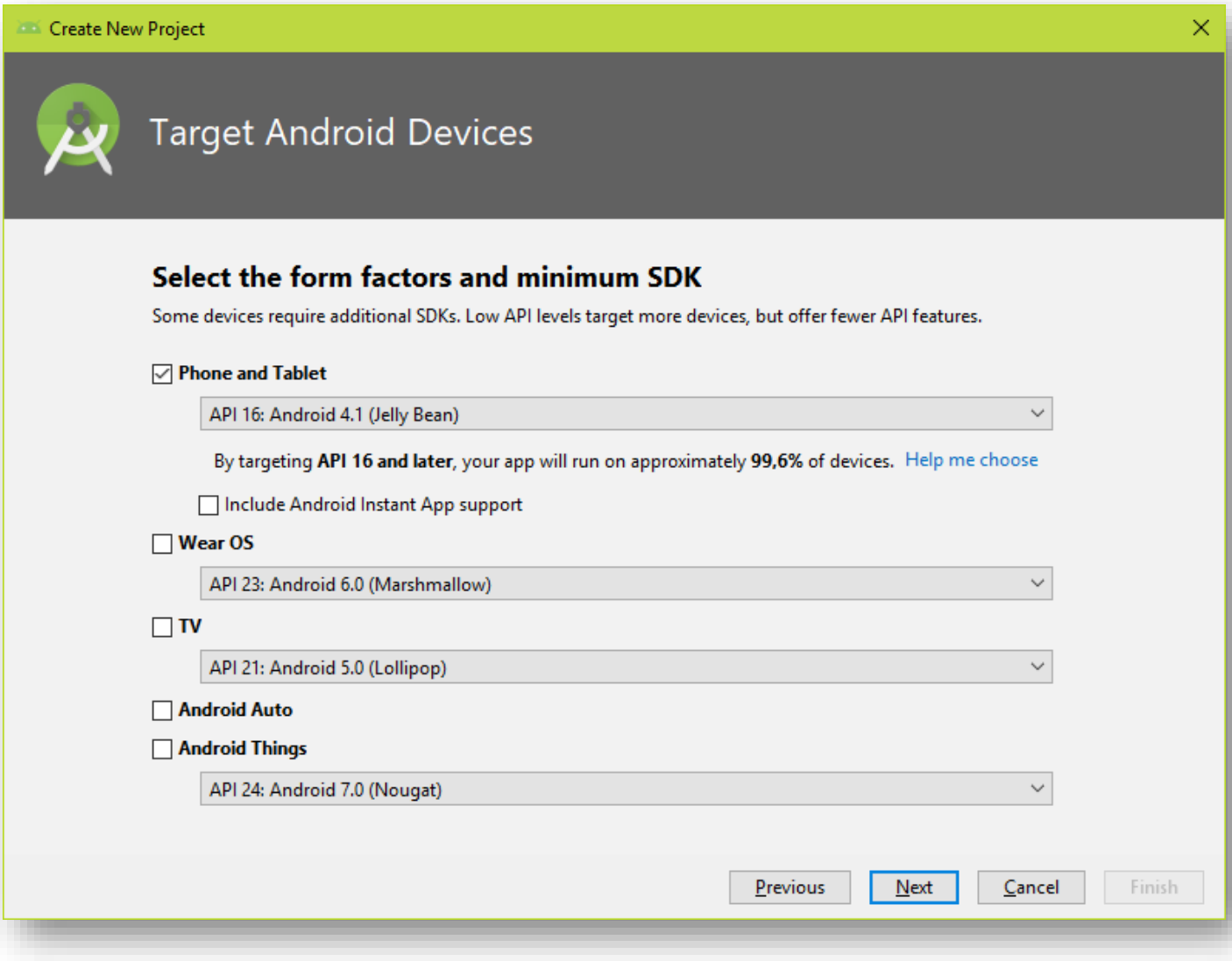
Now you can start the Studio for the first time. During this first start, an initialization wizard will download some more required components, as well as one version of the Android SDK APIs (at the time of writing, should be API 28, aka Android 9, aka Pie). In this wizard window, select ***Standard*** setup.

To build applications and manage dependences, Android Studio uses Gradle (a build system like Ant, or Maven). To make the Studio download the correct version of the Gradle plugin, you have to actually create and build a new app.

➔ In Studio start menu, select ***Start a new Android Studio project***.

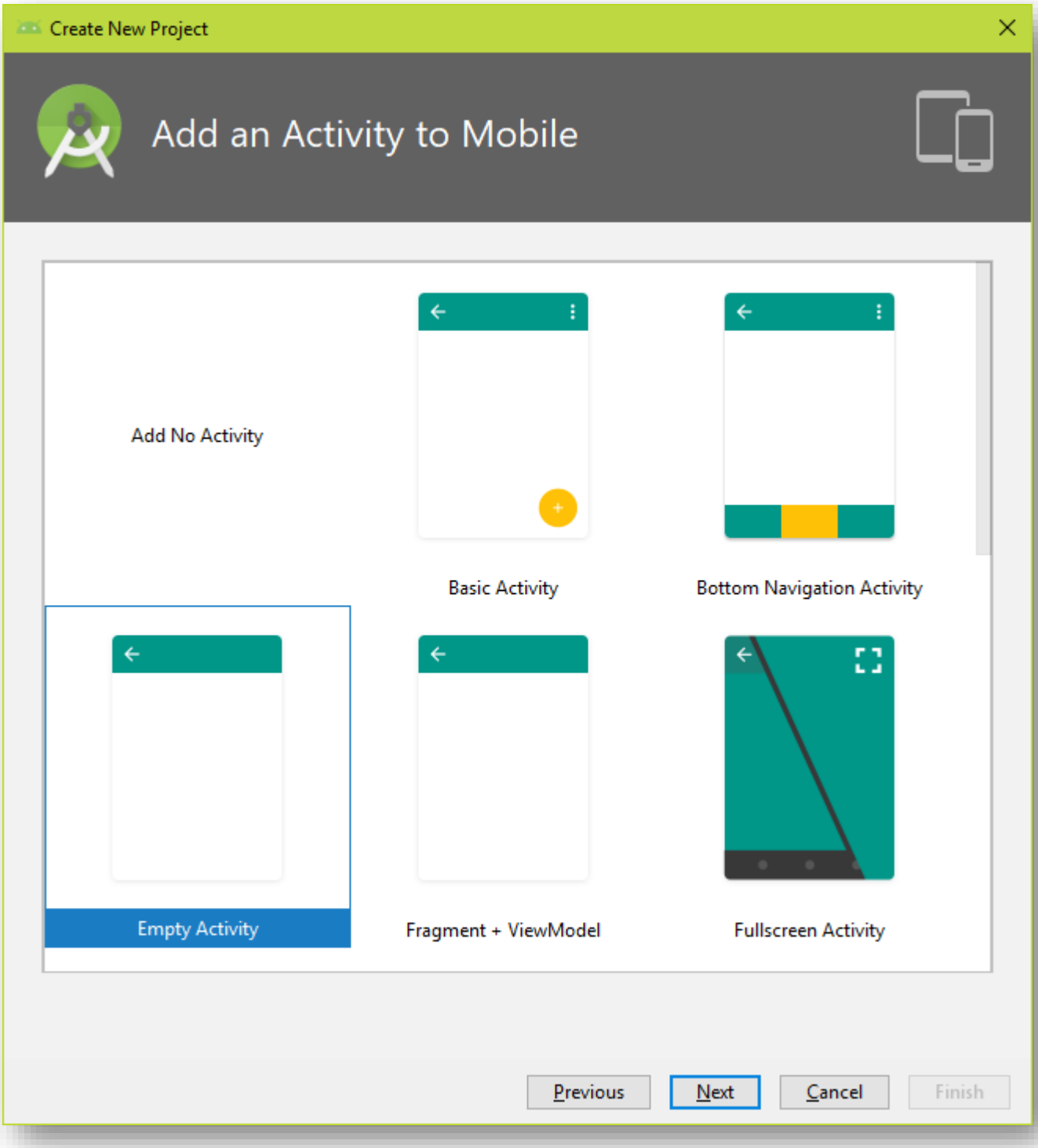


Configure your new project:
➔ leave default choices here but make sure the *Include Kotlin support* checkbox is ticked.

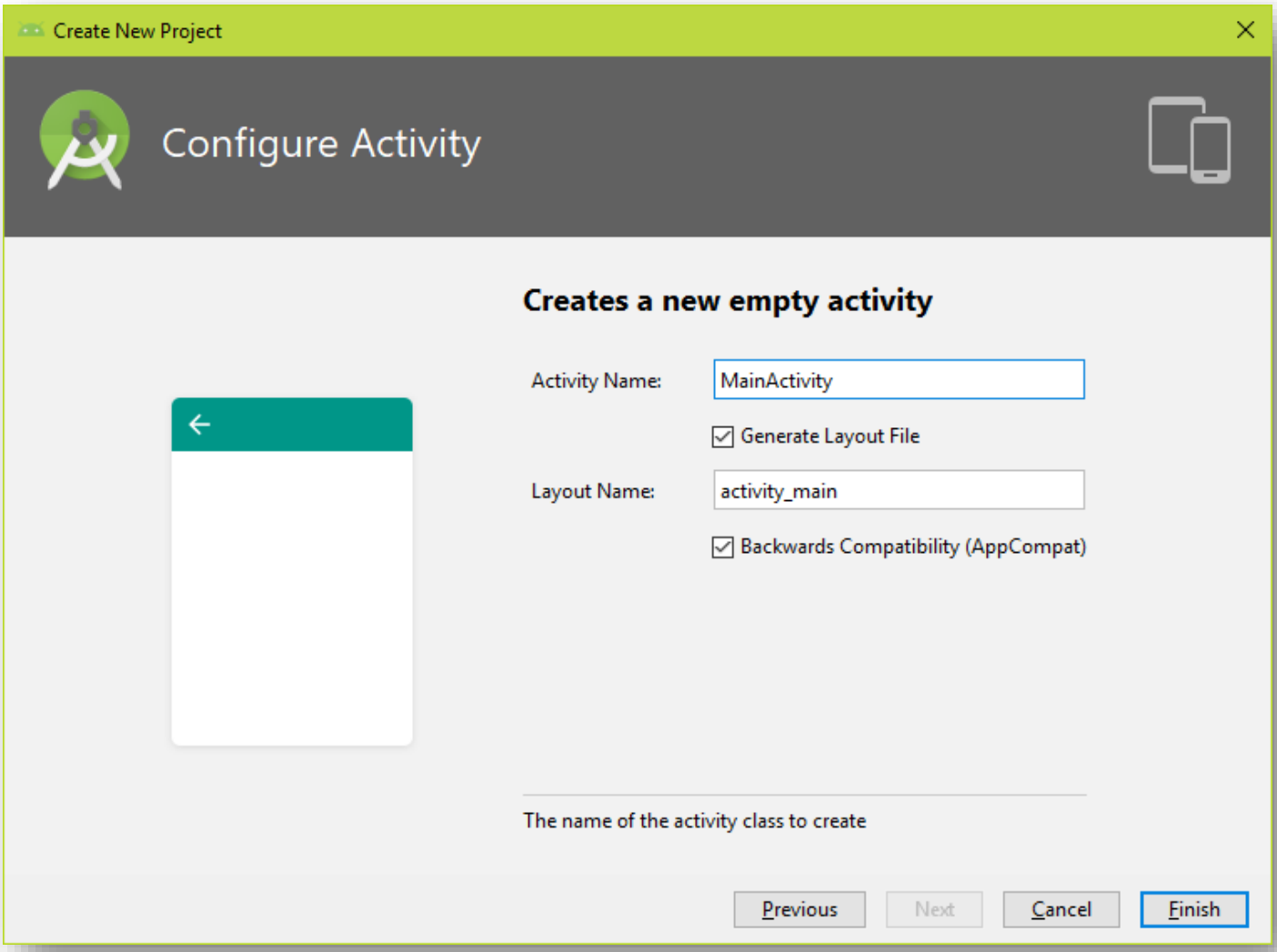


Select the form factors your app will run on:

➔ *Phone and Tablet* should be already checked. In the combo box, select *Minimum SDK API 16 : Android 4.1 (JellyBean)*



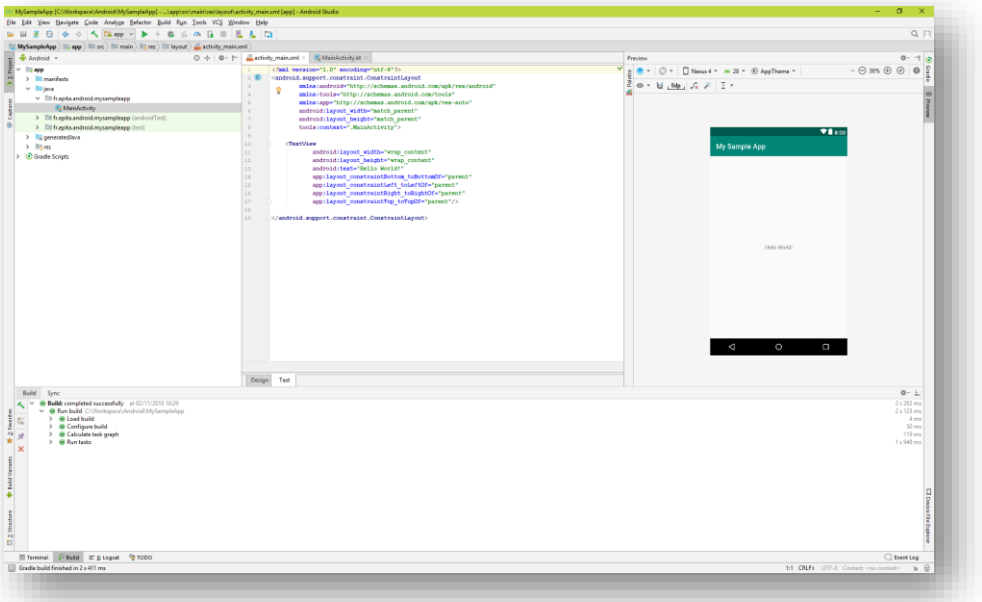
Add an activity to Mobile:
➔ select *Empty Activity*.



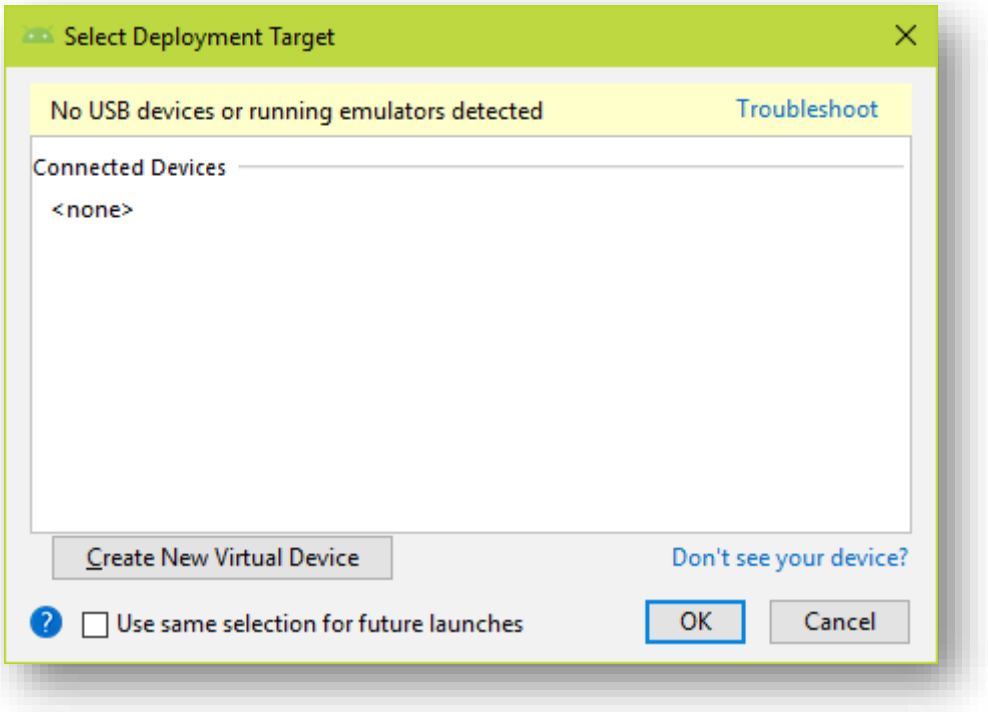
Customize the Activity :

➔ leave default choices in this page too and click on the ***Finish*** button.

Now it can take some time but you should finally have the project window open. Note that you can monitor at the bottom of the window if the Studio is busy Building, Indexing...



➔ when everything is finished and ready, build your app by clicking the ***Run app*** button (green triangle button) in the tool bar. Gradle will then build your app, and download any missing files. If the app builds correctly, you should reach the Deployment window (which you can close for now).



If you reach this window, then all should be ok. Next step is to actually deploy the built application, and in order to do so, you need an Android device.

4° An Android device

To run your applications, you will need an Android device. To get one, you have three possibilities, listed hereafter:

4 - 1 Use the embedded AVD emulator

When running an application you must select a device, as shown in the above screenshot. From here, you can click the ***Create new Emulator*** button. During this process, and depending upon what you may have already installed during startup, you may be prompted to download additional components (hamx accelerator, android system image…).

➔ Click the Create new Emulator button
➔ Just follow the steps and create a Nexus 5 phone emulator running at least Android Kitkat.

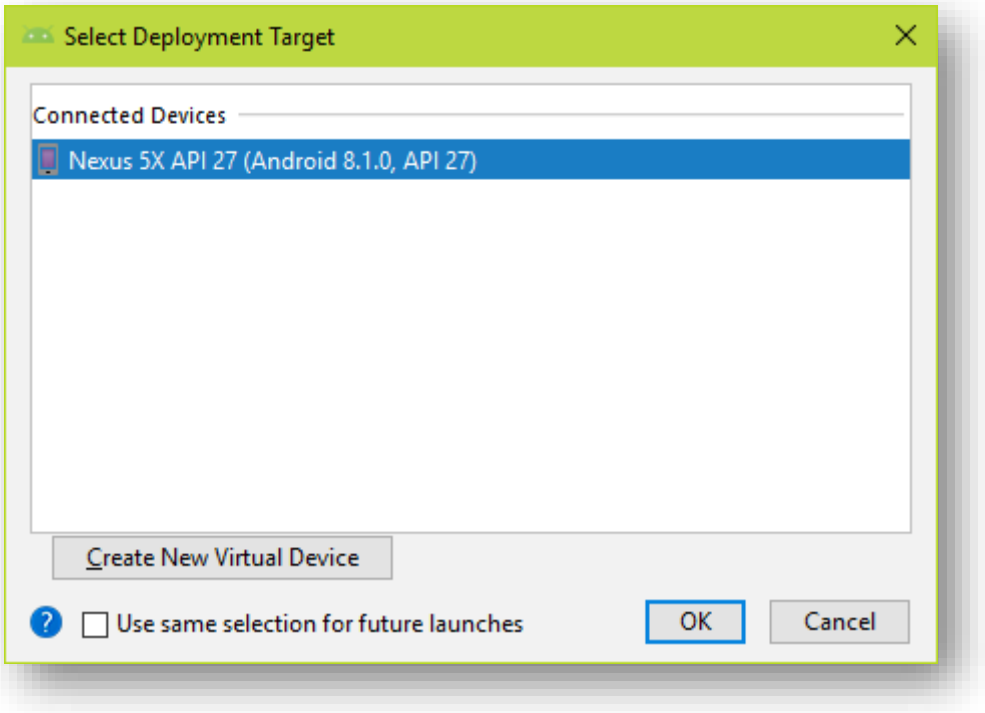
4 - 2 Own a real Android device.

Phone or tablet, it does not matter, as long as it runs Android version 4.1 at least (Jelly Bean)

Note: Windows 7~8 users, if you plan to use your own device, check on the web if it requires a specific driver to be recognized by Android Studio (like Samsung devices, or most HTC...). The driver for Nexus devices can be downloaded directly from the Studio SDK manager (It is called Google USB driver). Windows 10 and MacOS should install everything as needed automatically.

Test your app

➔ start your device (avd or real)
➔ once the device is started, enable USB debugging (should be ok by default if you use AVD). Check on the web how to enable developer mode and USB debugging on your specific device. If you use a real device, do not forget to connect it to your pc. When the phone is recognized usable by Android Studio, an authorization window should pop on the phone.
➔ Run your app again and check that the device appears in the list and that you can deploy your test app on the device.



Final test :

➔ close everything and try to create and run a new project with no Internet connection. If this final step is ok, you are ready.