



Strumenti di sviluppo

Programmazione Mobile

A.A. 2021/22

M.O. Spata



Gli strumenti di sviluppo

- Strumenti essenziali per lo sviluppo di una applicazione Android sono:
 - Java Development Kit
 - Android Studio (in precedenza era anche possibile scaricare Semplicemente l'SDK e integrarlo opportunamente con l'IDE Eclipse)
 - A meno che non siate masochisti, o quasi, sarebbe opportuno installare un IDE per facilitare il procedimento di sviluppo (stiamo parlando oggi di Android studio).

Gli strumenti di sviluppo

- Le versioni dei tools indicati sono disponibili per Windows, MacOS e Linux, pertanto è possibile scegliere il sistema operativo preferito per sviluppare Applicazioni Android.
- L'SDK inoltre include in emulatore per per tutti e tre i sistemi operativi; il fatto che l'applicazione Android girerà su una virtual machine non privilegia l'uso di alcun sistema operativo.

Gli strumenti di sviluppo: Gradle

- Come già detto il codice per applicazioni Android è scritto utilizzando la sintassi Java; inoltre le Librerie Core includono la maggior parte delle funzionalità offerte dalle API del Core JAVA.
- Prima di essere eseguito però il bytecode derivante dalla compilazione Java è trasformato in Dalvik byte code (.class --> .dex).
- Come risultato, si avranno benefici dall'aver utilizzato java come linguaggio e paradigma di programmazione ed in più si avrà un codice ottimizzato per l'esecuzione su dispositivi mobili con Android.

Gli strumenti di sviluppo

- L'SDK (distribuito ed integrato in Android Studio) include tutte le librerie Android, la documentazione ed applicazioni di esempio.
- Inoltre offrirà strumenti per aiutare la scrittura ed il debugging di applicazioni quali ad esempio l'Emulatore Android che permette di eseguire un progetto su una Dalvik Virtual Machine (ART) ed il Dalvik Debug Monitoring Service (DDMS) per aiutare nella fase di debugging.

Gli strumenti di sviluppo

- Chiunque abbia sviluppato per dispositivi mobili, indipendentemente dalla piattaforma e dal framework utilizzato, sa che determinate scelte progettuali sono dettate dalle dimensioni ridotte (specie del display), dalla limitata capacità di elaborazione e dalla limitata quantità di memoria.
- Non si deve essere esperti, comunque, per capire che non tutto quello che è stato creato per il web o per applicazioni desktop può funzionare su un dispositivo mobile.

Gli strumenti di sviluppo

- Un dispositivo mobile viene di solito utilizzato in movimento e rappresenta il più delle volte una distrazione piuttosto che un momento di concentrazione, pertanto le applicazioni sviluppate per essi dovrebbero essere veloci, reattive e facili da usare.
- Nel seguito della lezione sarà mostrato come scaricare ed installare e configurare l'ambiente di sviluppo, al fine di realizzare la prima applicazione.
 - Hello World

Gli strumenti di sviluppo

- Come già detto, poiché una applicazione Android viene eseguita su una Dalvik Virtual Machine o ART, essa può essere sviluppata su qualunque piattaforma purchè quest'ultima sia in grado di supportare gli strumenti di sviluppo.
- E' pertanto possibile scaricare i primi due strumenti di seguito riportati su una qualsiasi macchina che abbia Windows, Mac OS, Linux:

Gli strumenti di sviluppo

- Java Development Kit (JDK) :
 - Si tratta della piattaforma completa e dell'ambiente di run-time.
 - Il link per scaricare il kit è:
 - <https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>
- Android Studio
 - <https://developer.android.com/studio#Other>

In precedenza potevamo scaricare

- Android SDK:
 - Tale strumento è totalmente Open ovvero non si deve sostenere alcun costo per fare il download o utilizzare le API; il link per scaricare il kit è:
 - <http://developer.android.com/sdk/index.html#Other>
 - In realtà attraverso questo link vengono installati solo una serie di tools a supporto dell'SDK e viene creata la struttura di folder e file per ospitare le varie piattaforme Android e le relative API nelle diverse Versioni

Gli strumenti di sviluppo

- Scaricare Eclipse....
 - <https://eclipse.org/downloads/packages/release/Luna/SR2>
- Scaricare il Plug-In ADT per Eclipse seguendo la procedura seguente:
 - Dal menu help cliccare su Software Updates;
 - Dalla interfaccia che il sistema offre aggiungere il collegamento al sito
 - <https://dl-ssl.google.com/android/eclipse/>
 - Selezionare ed installare quindi i Developer Tools

Gli strumenti di sviluppo

- L'ADT prima ed oggi Android Studio offrono le seguenti funzionalità:
 - Un Wizard per la creazione di Progetti Android che semplifica la creazione di nuovi progetti e di fatto costruisce un modello di base dell'applicazione.
 - Un Editor "visuale" che consente la creazione di manifest, layout e risorse rappresentate attraverso XML.
 - Programmi per la compilazione e la costruzione dell'applicativo finale Android, compresa la generazione del file .dex, i file package e la routine di installazione per la Dalvik Virtual Machine (ART).

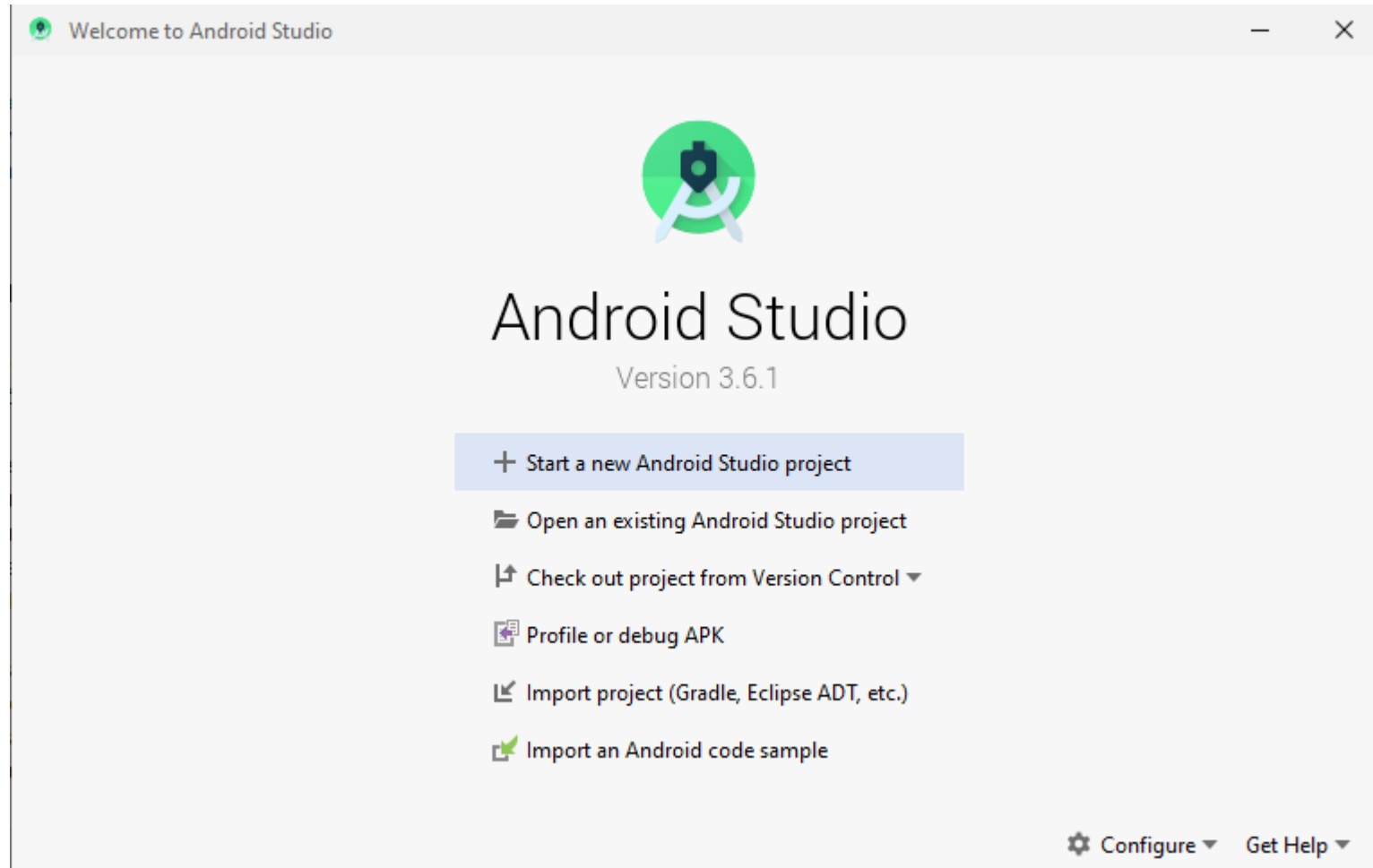
Gli strumenti di sviluppo

- L'Emulatore Android completo nelle sue parti:
 - interfaccia,
 - wizard per il setting delle caratteristiche della rete,
 - routine per la simulazione di chiamate ed SMS entranti.
 - Il DDMS (Dalvik Debug Monitoring Service) che include il port forwarding, il monitoraggio di heap, stack thread e processi ed altre utilities.
 - L'accesso al file System del dispositivo e/o dell'emulatore.
 - Debugging a run-time (breakpoints etc)
 - L'accesso a tutte le console ed i log di Android e Dalvik.
- Dal Menu Preferences potete impostare la directory per Android (directory di installazione dell'SDK).

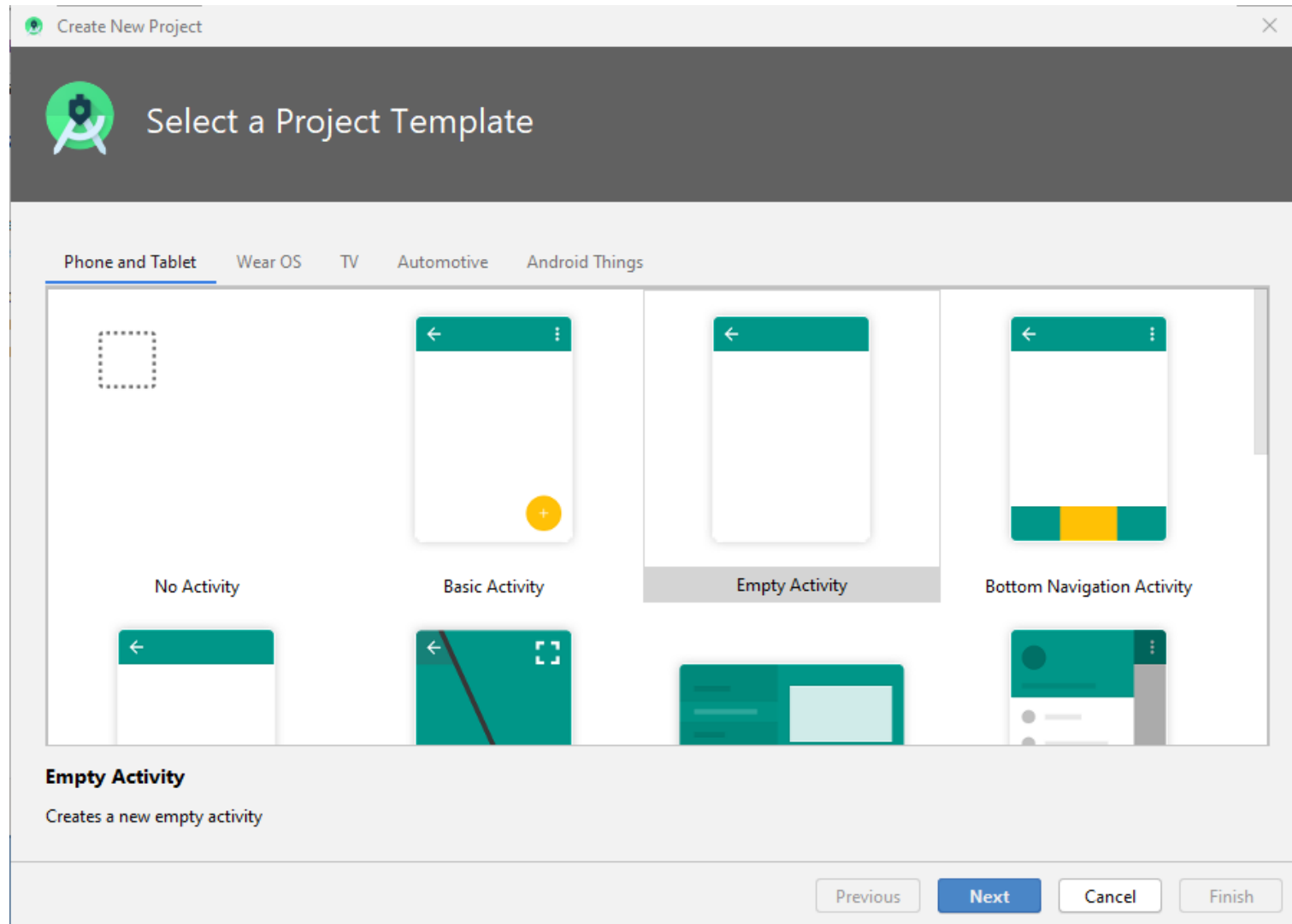
Hello World

- Una volta installato tutto il necessario è possibile provare a creare un nuovo progetto e impostare l'ambiente per mandare in esecuzione e/o debuggare l'applicazione creata.
- Ovviamente per creare il nuovo progetto ci avvaliamo del Wizard.
- Lanciamo Android Studio:

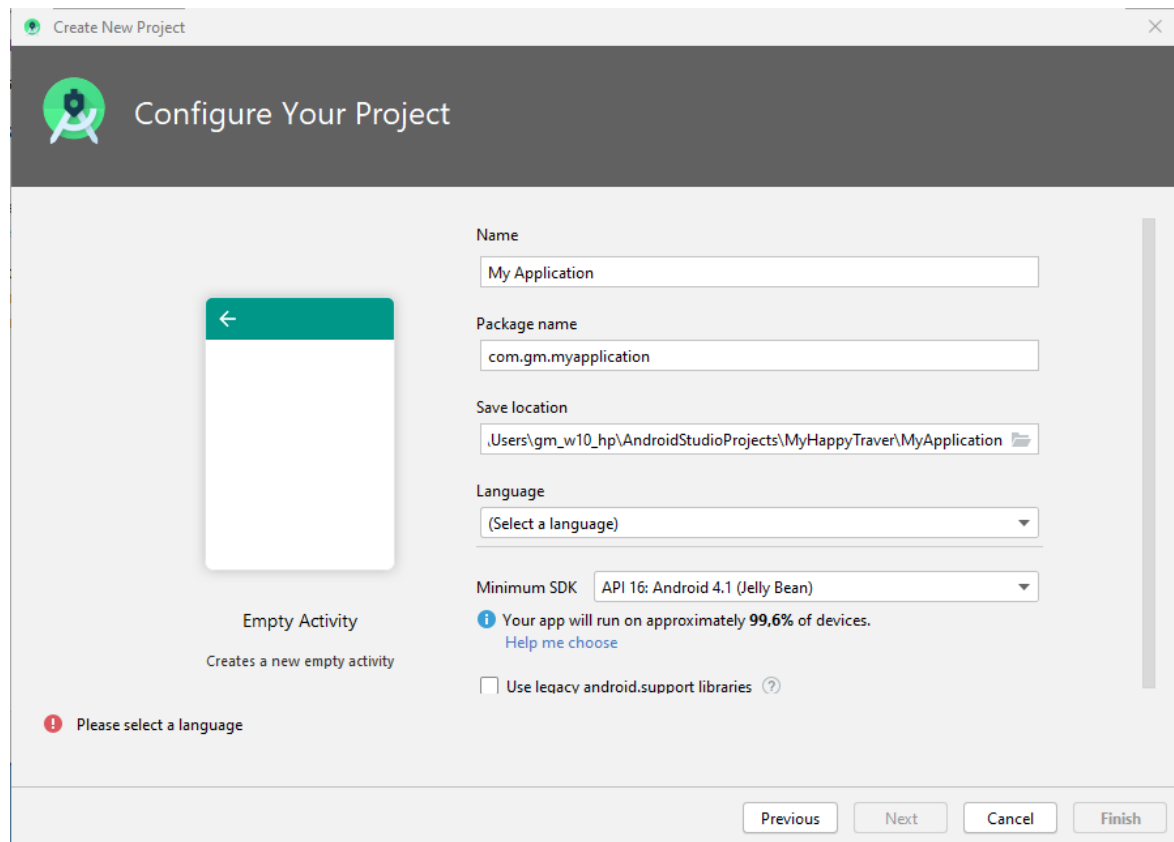
Andoid studio



Android studio



Android studio

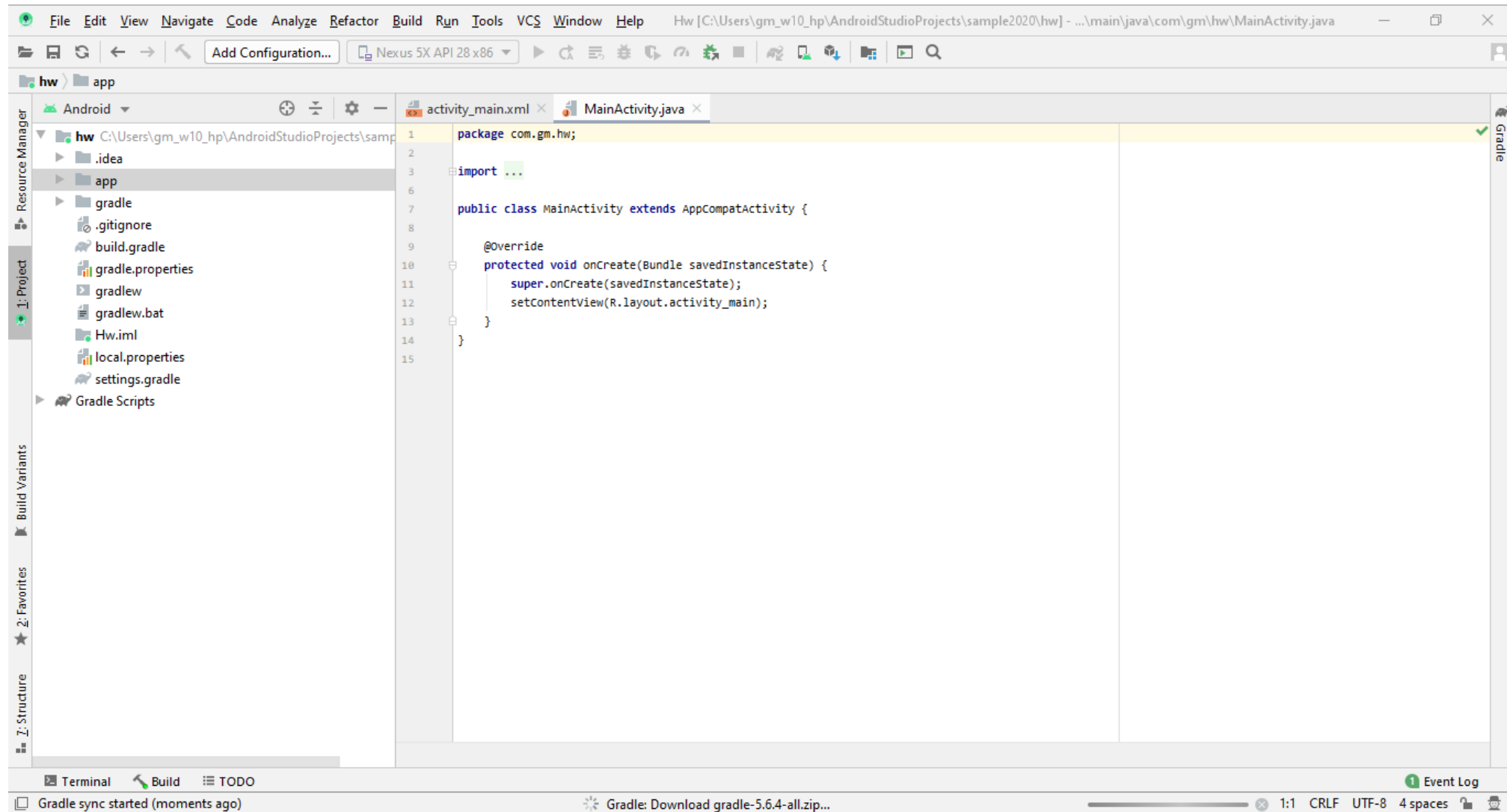


- Impostare quindi i parametri richiesti quali:
 - Name: è quello che apparirà nel play store, e nei settings per la gestione dell'app.
 - Package_name: identificatore univoco dell'app...
 - Il linguaggio da utilizzare (Kotlin o Java)
 - Minimo SDK supportato

Android studio

- A questo punto l'IDE provvede alla creazione dei file di progetto necessari tra cui una nuova classe Activity.
- Il progetto non sarà completamente vuoto ma rappresenta l'implementazione del programma "Hello World", che può essere usato come modello per la propria applicazione.

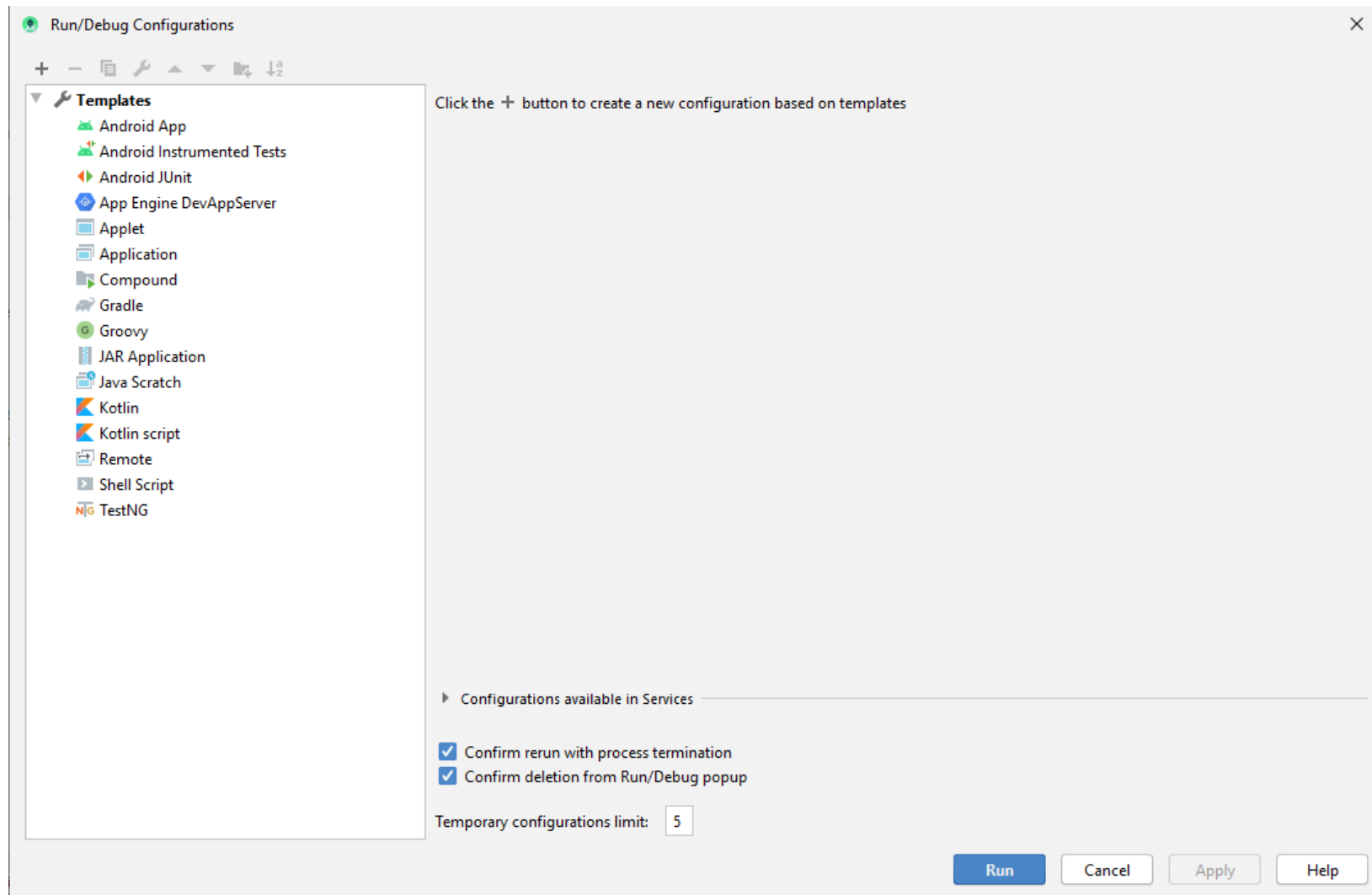
Una volta effettuate le scelte



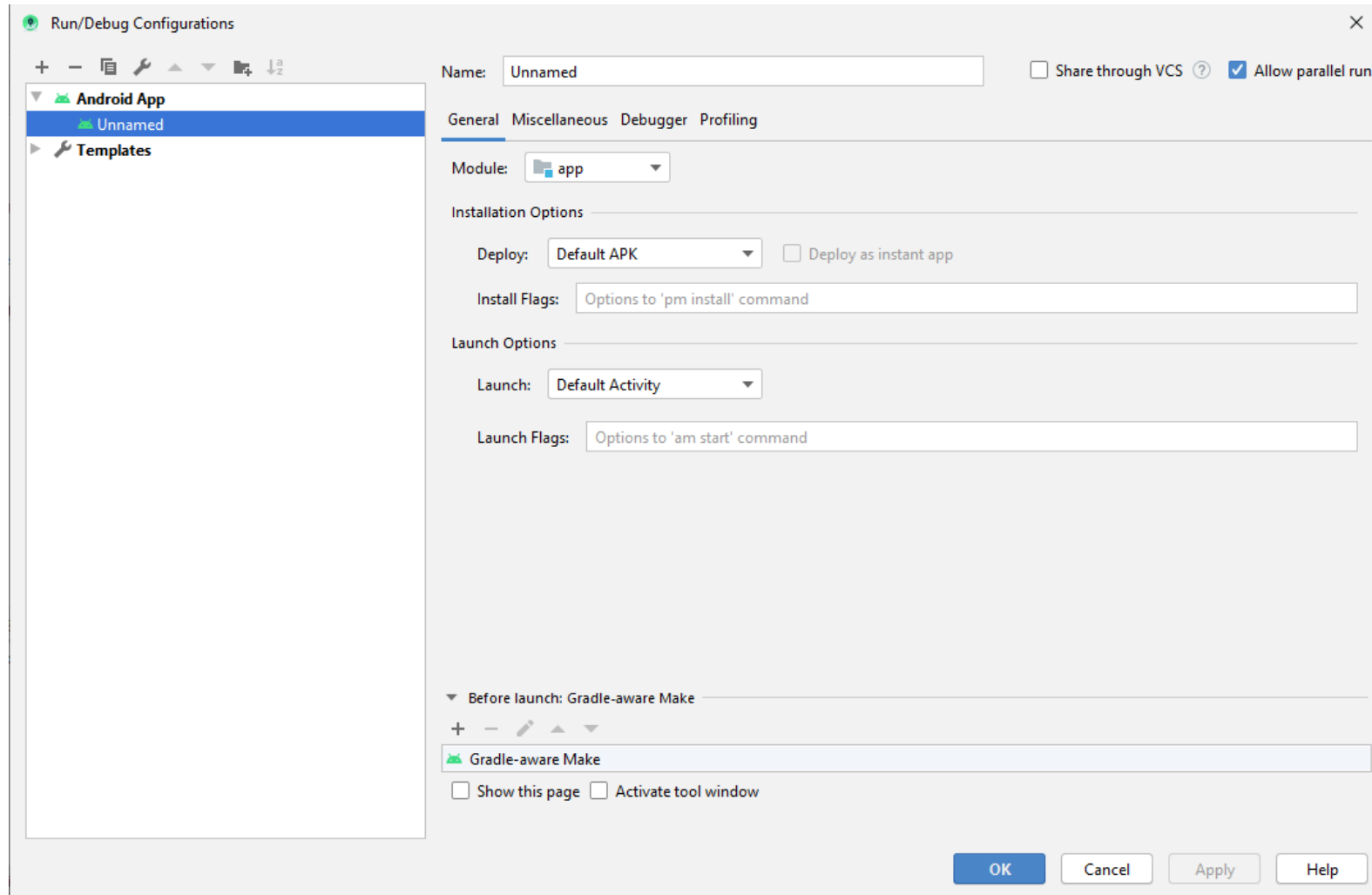
Android studio

- Prima di modificare l'applicazione è opportuno creare la configurazione utile per lanciare (in esecuzione e/o in debug) l'applicazione.
- In particolare attraverso questa configurazione si specificano:
 - Il progetto e/o l'attività da lanciare
 - Le opzioni per l'Emulatore
 - Alcuni settings di I/O
- La procedura prevede di aprire l'opportuna dialog per il setting attraverso la voce di menu Run - Edit Configurations....

Android studio



Android studio



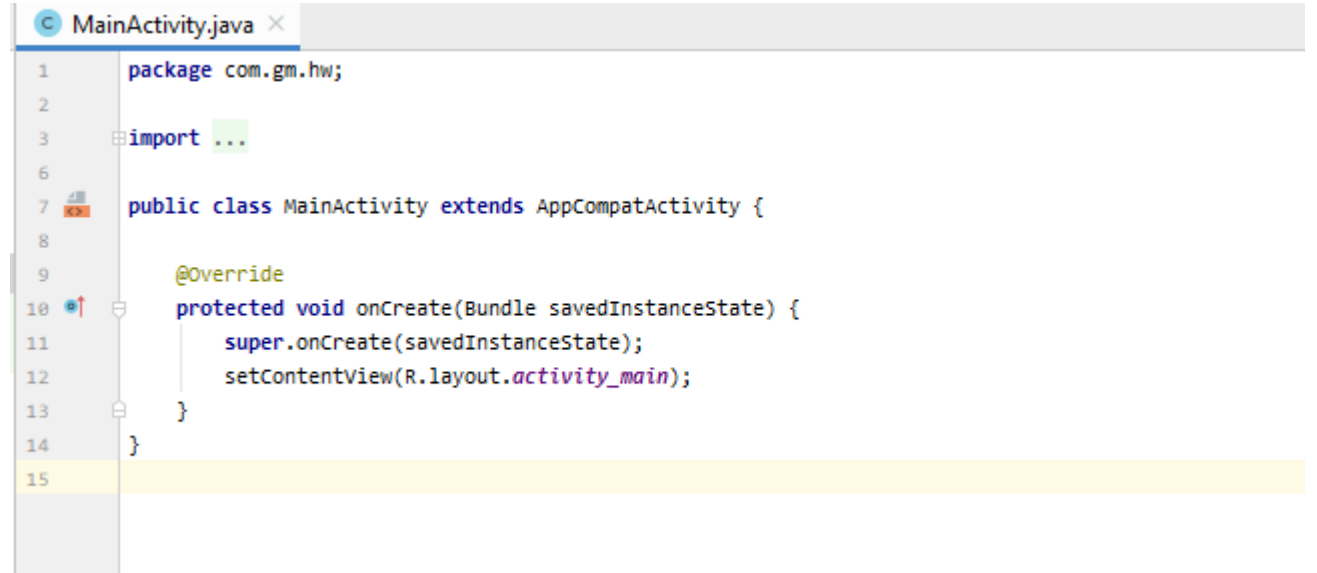
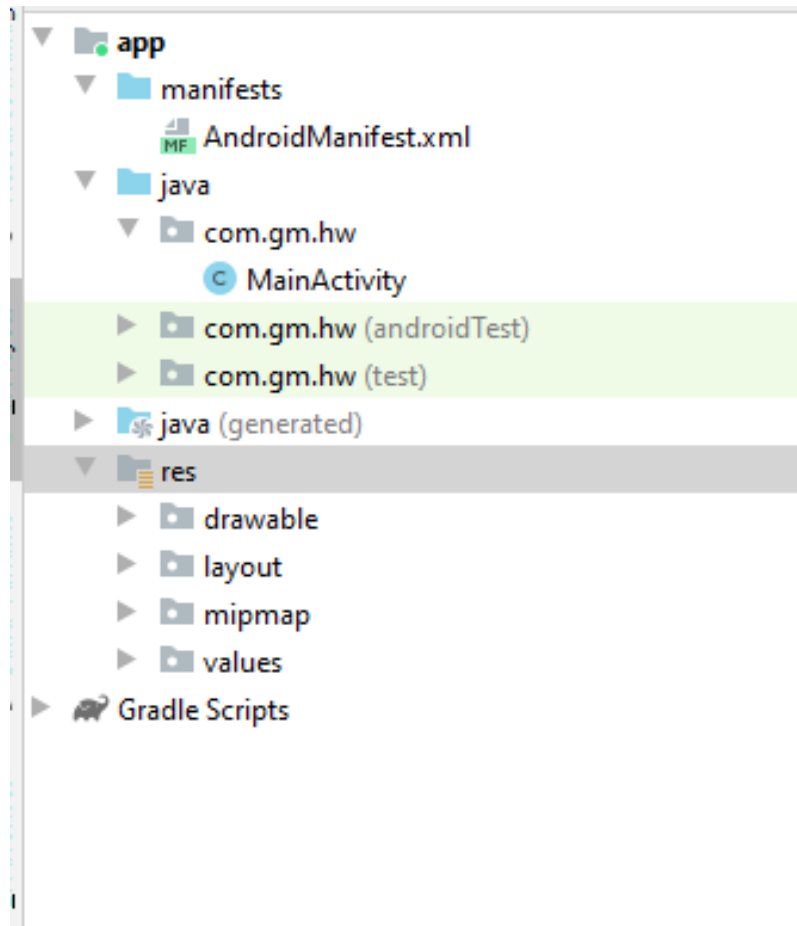
Android studio

- Ogni nuova configurazione richiede un nome, e per essa è possibile andare a specificare:
 - Le opzioni di start-up: relativamente al progetto selezionato, permette di scegliere quale attività deve essere lanciata nel momento in cui si esegue o debugga l'applicazione.

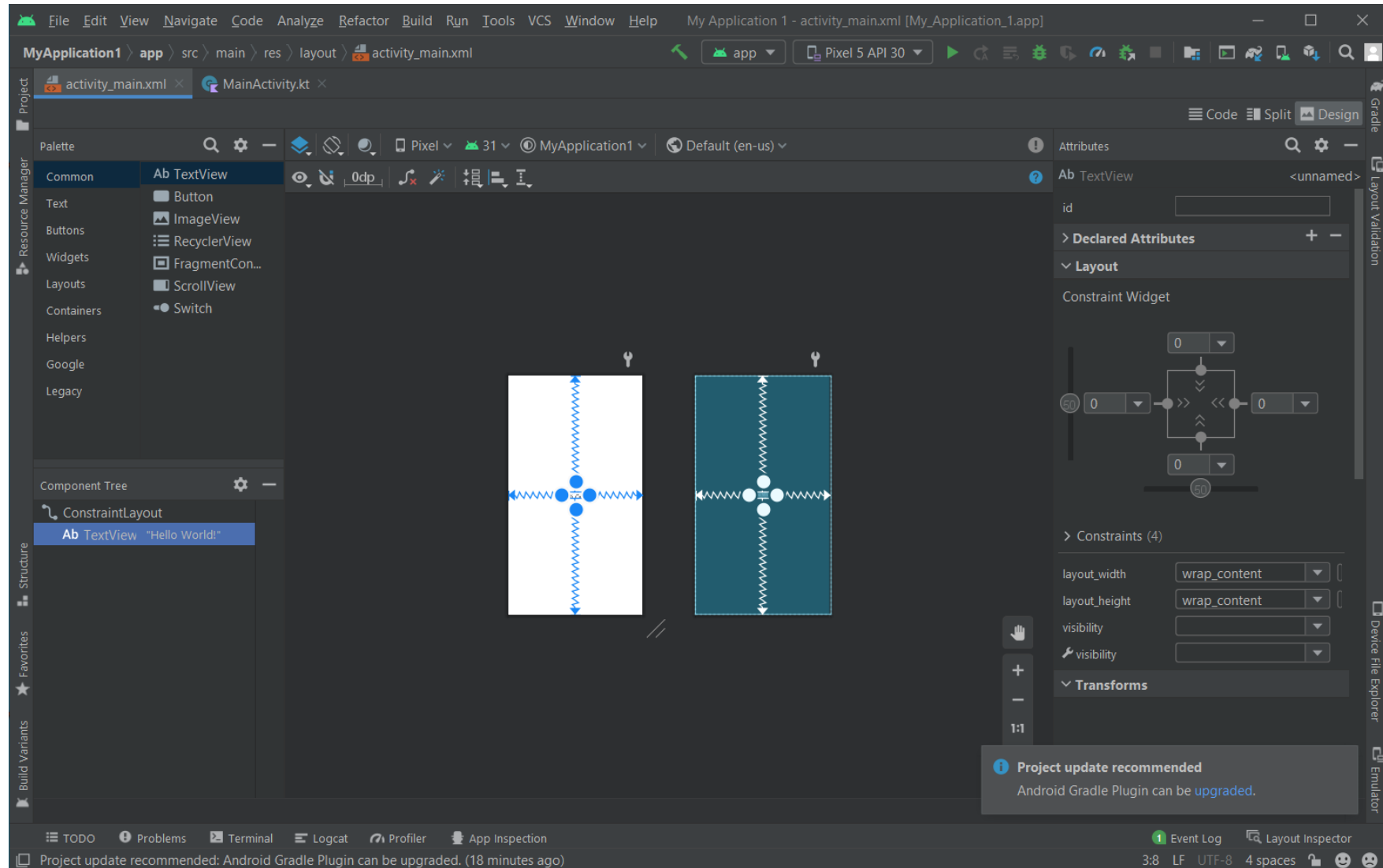
Android studio

- Il progetto è pronto per essere eseguito e/o debuggato; per far ciò scegliamo dal menu Run il comando Run o Debug; Android studio si occuperà di :
 - Compilare il progetto corrente e convertire "l'eseguibile" in un Eseguibile Android (.dex).
 - Costruire un Package Android (.apk) contenente l'eseguibile e le risorse esterne.
 - Mandare in esecuzione l'emulatore
 - Installare l'applicazione sull'emulatore
 - Mandare in esecuzione l'applicazione.

Android studio



Layout di un app



Codice di Hello World

- Se guardiamo nel progetto nella cartella java, in corrispondenza del package, specificato in fase di setting up dell'app, troviamo una unica classe
- Si tratta della classe *MainActivity* che estende la classe *AppCompatActivity*: quest'ultima rappresenta una classe fondamentale per i componenti interattivi e visuali delle applicazioni.
- Nel codice appena visto non vi sono riferimenti al layout dell'interfaccia dell'applicazione.
- A proposito in Android i componenti visuali, per intenderci gli equivalenti ai controlli delle applicazioni tradizionali, sono chiamate Views.

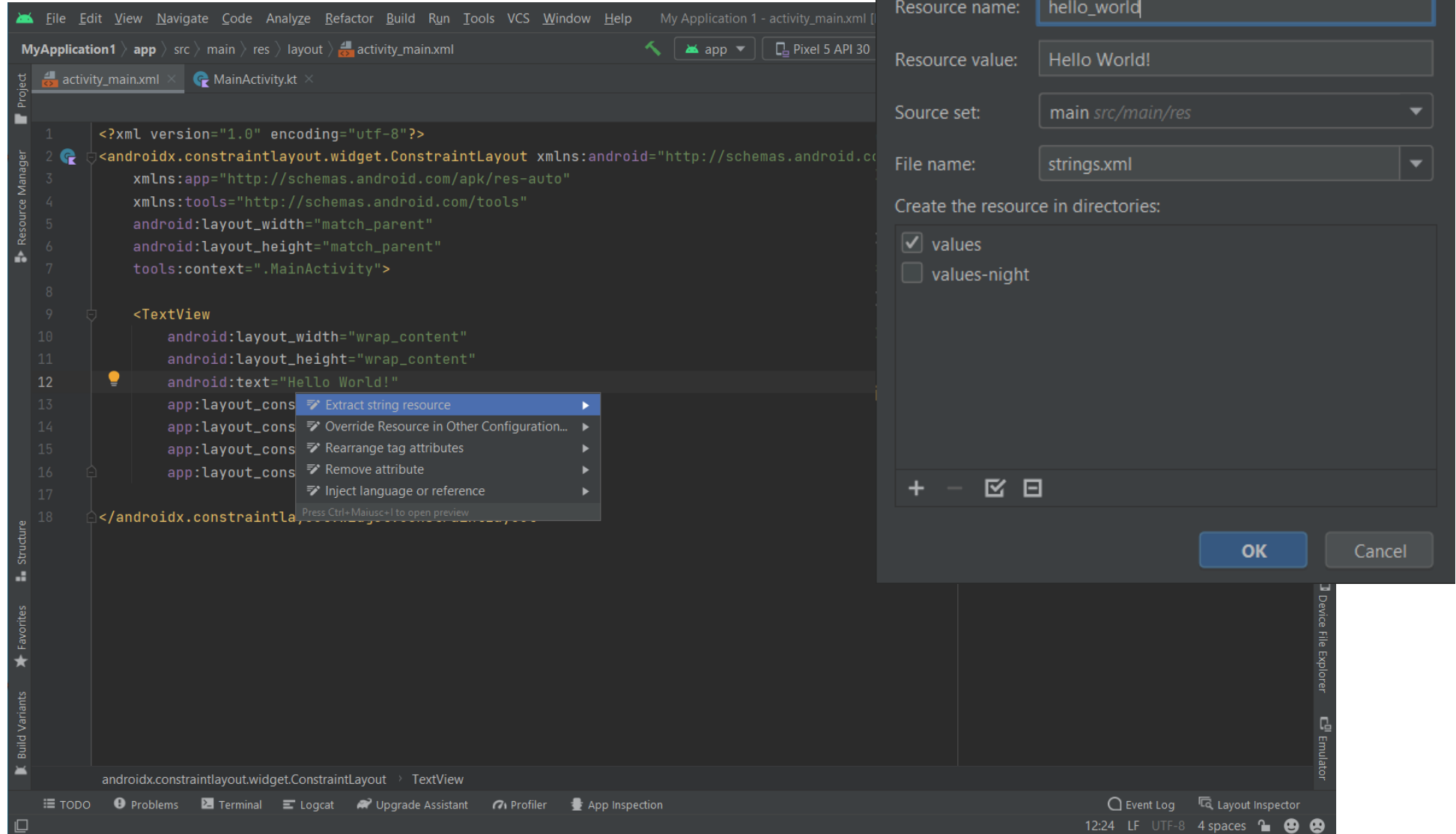
Android studio

- Definire le risorse in XML e fare riferimento ad esse attraverso oggetti specificatamente creati è la via preferita per implementare interfacce utente: ciò permette di disaccoppiare la logica delle applicazioni dalla specifica interfaccia.
- Il sistema fornisce un metodo per accedere a tutte le risorse in fase di codifica. Lo fa permettendo di aggiungere degli identificatori alle risorse nel file XML ed utilizzare un metodo per ottenere un riferimento ad esse attraverso questi id.

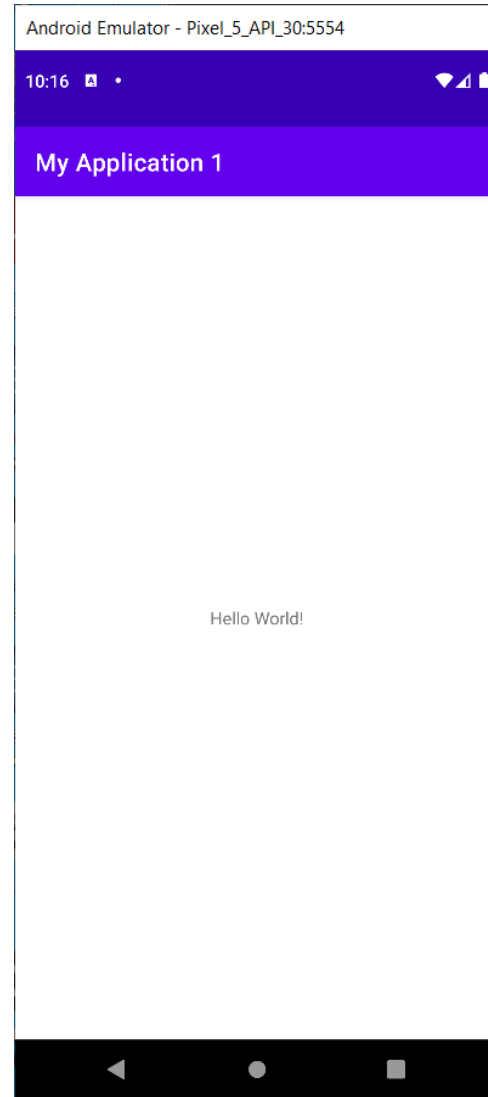
Android studio

- L'unico riferimento nel template al layout è la chiamata al metodo `setContentView(R.layout.activity_main)`. Questa chiamata ci permette di visualizzare l'interfaccia utente utilizzando opportune risorse.
- Le risorse per un progetto Android sono memorizzate sotto forma di XML nella cartella di progetto `res` e suddivise in `drawable`, `layout`, `values` (descritte già precedentemente).
- Android Studio interpreta questi file XML e fornisce accesso (at design time) attraverso codice grazie all'oggetto `R` (meglio descritto in seguito).

Android studio



Il simulatore di device android



Android studio

- Nessuno vieta di definire risorse e layout direttamente via codice:
-// codice di onCreate

```
LinearLayout.LayoutParams lp;  
lp = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,  
LinearLayout.LayoutParams.FILL_PARENT);  
LinearLayout.LayoutParams textViewLP;  
textViewLP = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,  
LinearLayout.LayoutParams.WRAP_CONTENT);  
LinearLayout ll = new LinearLayout(this);  
ll.setOrientation(LinearLayout.VERTICAL);  
TextView myTextView = new TextView(this);  
myTextView.setText("Hello World, HelloWorld");  
ll.addView(myTextView, textViewLP);  
this.addView(ll, lp);
```