

Installation d'Android Studio

Thierry Vaira <tvaira@free.fr> <http://tvaira.free.fr/>

17/07/2016 (rev. 1)

Table des matières

Installation d'Android Studio	1
Présentation	1
Installation Java SE Development Kit	2
Installation d'Android Studio	2
Tests	4
Installation d'un périphérique Android	5
Commandes	6
Première application	6
Documentation	12

Installation d'Android Studio

Présentation

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA.

Android Studio permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.



Android est un système d'exploitation mobile basé sur le noyau Linux et développé actuellement par Google.

Les différents kits de développement sont :

- l'Android NDK (*Android Native Development Kit*) est une API du système d'exploitation Android permettant de développer directement dans le langage C/C++ du matériel cible, par opposition au Android SDK qui est une abstraction en bytecode Java, indépendante du matériel.
- l'**Android SDK** est un ensemble complet d'outils de développement (pour Linux, MAC OS ou Windows). Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux.
- l'Android ADK (*Accessory Development Kit*) a été mis en oeuvre par Google afin de développer des accessoires via des cartes d'extensions (comme Arduino) avec Android. La connexion s'effectue via le port USB ou en *Bluetooth*.

Les outils :

- l'ADB (*Android Debug Bridge*) est un outil inclus dans le package Android SDK. Il se compose d'un programme client et d'un programme serveur communicant entre eux et qui permet :
 - la copie de fichier ;
 - l'accès à la console Android ;
 - la sauvegarde de la mémoire ROM ;
 - l'installation de logiciel.

Lien : [Les différentes versions d'Android](#)

Pour développer des applications Android, il faut au moins disposer du Android SDK et du kit de développement Java. Pour faciliter le développement, il est conseillé d'utiliser un EDI/IDE (*Integrated Development Environment*) comme **Android Studio**.

Ces outils sont disponibles aussi bien sous Linux que sous Windows et Mac OS (sur processeur Intel). Les applications Android étant exécutées par une machine virtuelle, il n'y a pas d'avantages particuliers à développer sur un système plutôt qu'un autre ...

Installation Java SE Development Kit

JRE (*Java Runtime Environment*) est le kit destiné au client pour pouvoir exécuter un programme Java. Il se compose essentiellement d'une machine virtuelle Java (JVM) capable d'exécuter le *bytecode* et les bibliothèques standard de Java.

Le kit destiné au programmeur, appelé avant JDK (*Java Development Kit*) et renommé depuis la version 1.2.2 en SDK (*Standard Development Kit*), est composé d'un JRE, d'un compilateur, de nombreux programmes utiles, d'exemples de programmes Java et des les sources de toutes les classes de l'API.

Le développement Android nécessite le SDK Java, le JRE seul n'est pas suffisant. **La version 2.1 d'Android Studio utilisée ici nécessite la version 1.8 de Java SDK.**

Si Java est déjà installé sur le poste, sa version peut être vérifiée en ligne de commande par :

```
$ javac -version
```

Sinon télécharger [Java SE Development Kit 8](#) puis l'installer :

```
$ sudo cp jdk-8u102-linux-x64.tar.gz /usr/local
$ cd /usr/local/
$ sudo tar zxvf jdk-8u102-linux-x64.tar.gz
$ sudo rm jdk-8u102-linux-x64.tar.gz
$ vim $HOME/.bashrc
export PATH=/usr/local/jdk1.8.0_102/bin:$PATH
$ source $HOME/.bashrc
```

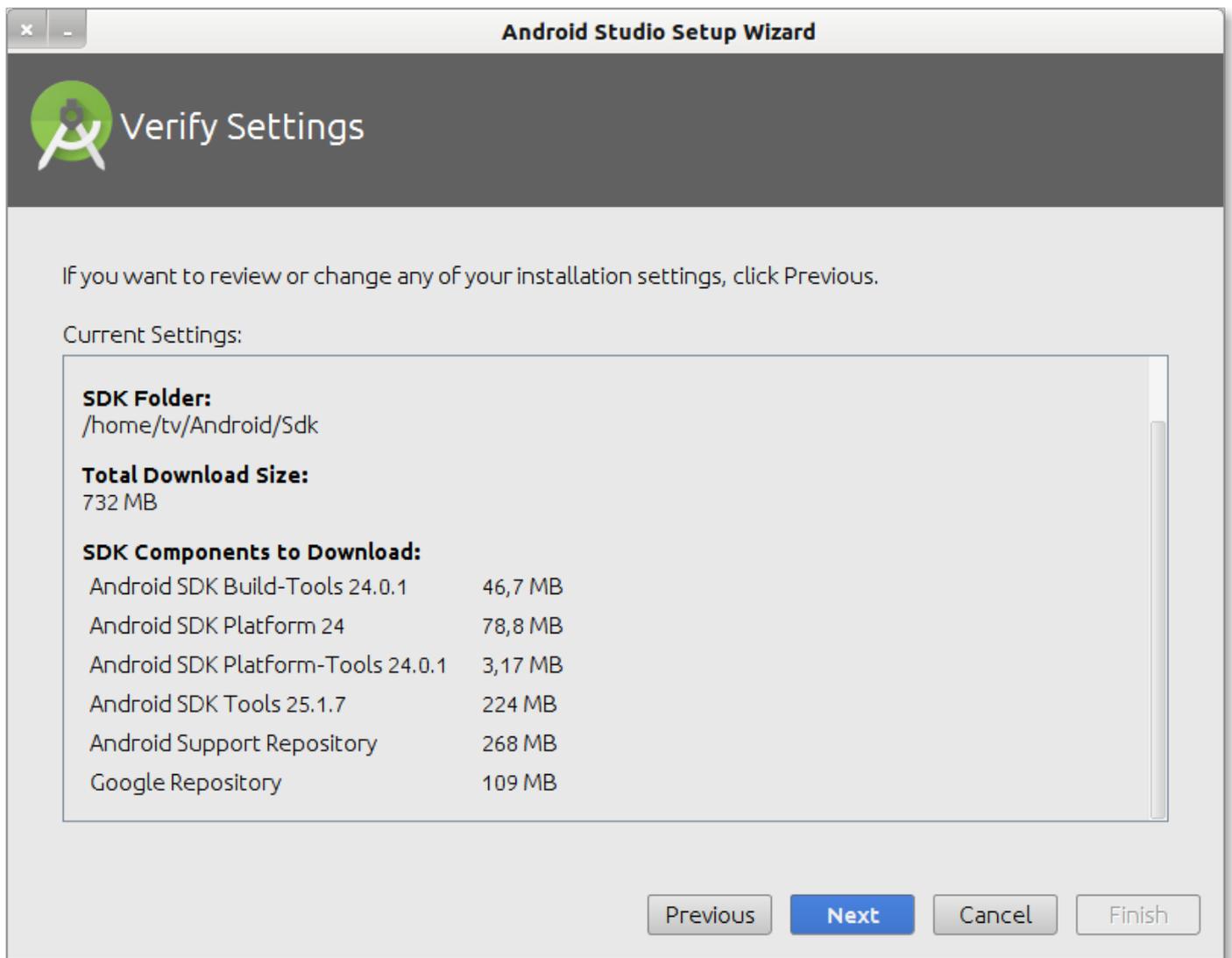
Installation d'Android Studio

Pour les systèmes 64-bits, il faudra tout d'abord installer les bibliothèques 32-bits suivantes :

```
$ sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc++6
```

Télécharger [Android Studio](#) puis l'installer :

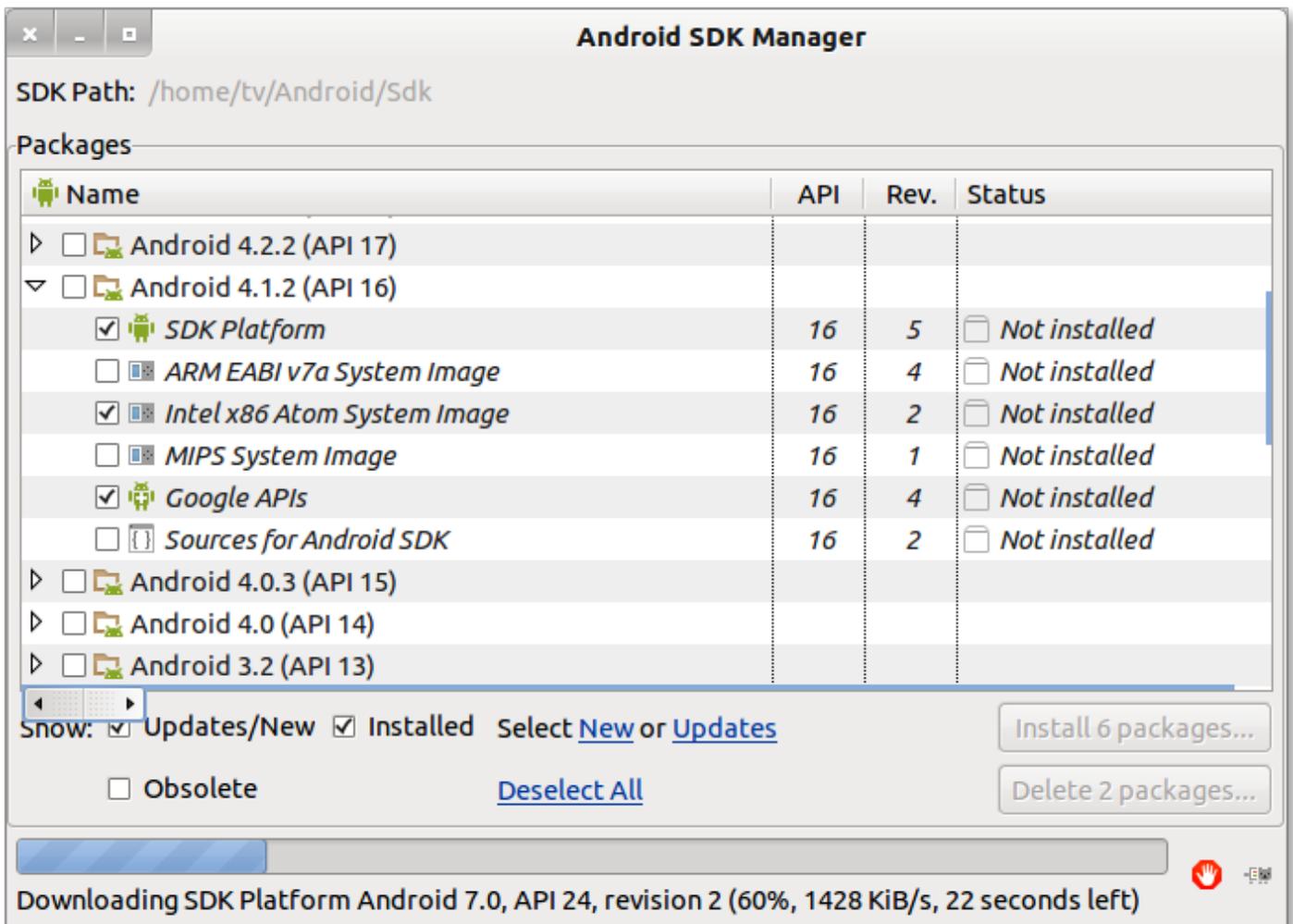
```
$ sudo cp android-studio-ide-143.3101438-linux.zip /usr/local
$ cd /usr/local/
$ sudo unzip android-studio-ide-143.3101438-linux.zip
$ sudo rm android-studio-ide-143.3101438-linux.zip
$ vim $HOME/.bashrc
export PATH=/usr/local/jdk1.8.0_102/bin:$PATH:/usr/local/android-studio/bin:$HOME/Android/Sdk/platform-tools:
    $HOME/Android/Sdk/tools
$ source $HOME/.bashrc
$ studio.sh
```



Le SDK Android est constitué de paquets modulaires qui peuvent être téléchargés séparément au moyen de l'outil Android SDK Manager. Cet outil est particulièrement pratique lorsqu'il est nécessaire de faire une mise à jour liée à une évolution du niveau d'API. Plusieurs niveaux d'API peuvent cohabiter.

Mettre à jour le SDK et choisir son API :

```
$ android
```



Remarque : cette commande est devenue obsolete et elle est remplacée par `sdkmanager` (25.2.3 et supérieur). Elle permet de visualiser, d'installer, de mettre à jour et de désinstaller des *packages* pour le SDK d'Android. On peut aussi le faire directement à partir d'Android Studio. La commande se trouve dans `$HOME/Android/Sdk/tools/bin`.

Tests

Les outils en ligne de commande (CLI) :

```
$ adb
Android Debug Bridge version 1.0.36
Revision af05c7354fe1-android
...
```

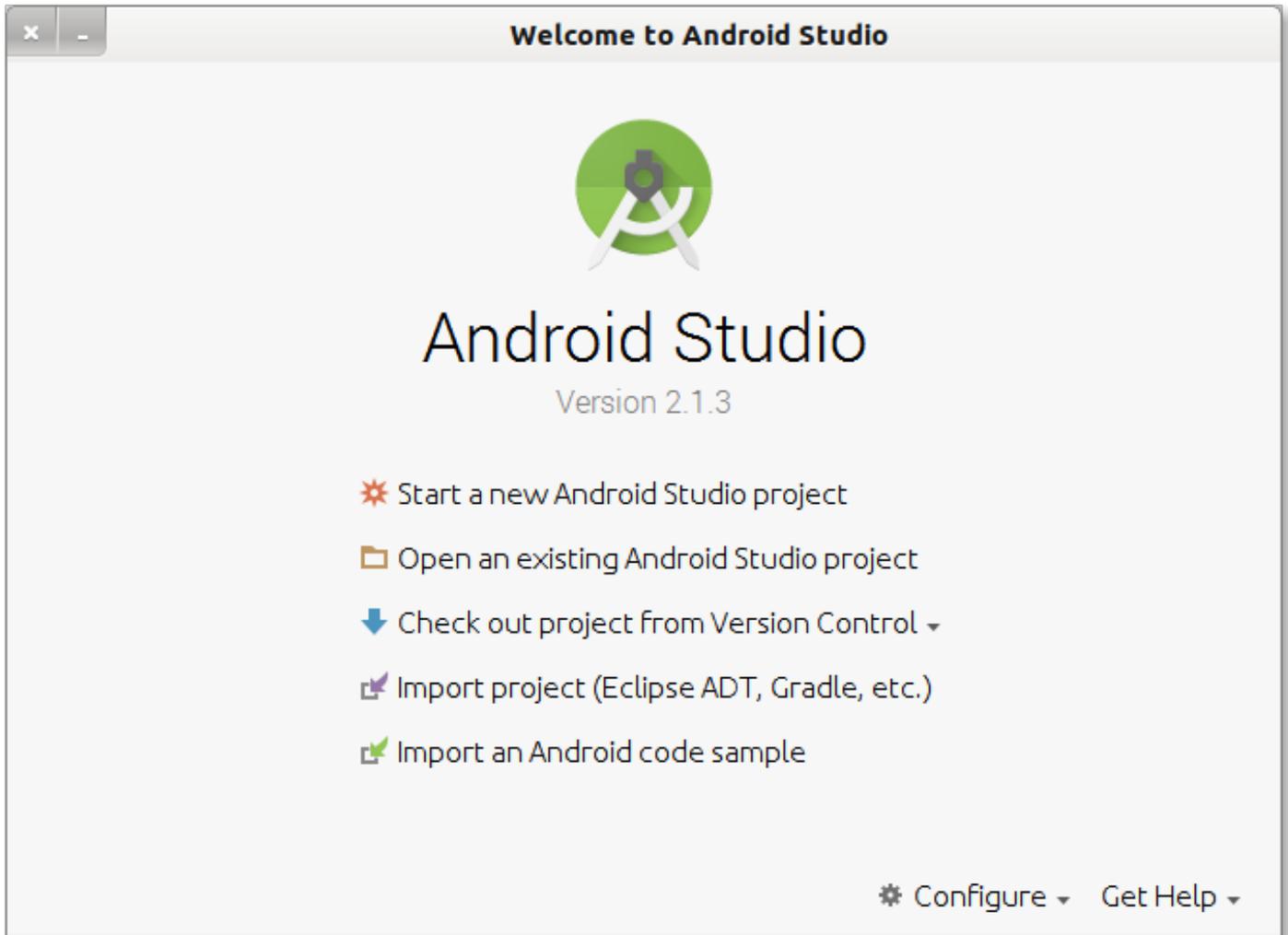
```
$ android list
Available Android targets:
-----
id: 1 or "android-24"
  Name: Android 7.0
  Type: Platform
  API level: 24
...
```

// Ou :

```
$ sdkmanager --list
```

On démarre l'IDE Android Studio :

```
$ studio.sh
```



Installation d'un périphérique Android

Passer le smartphone ou la tablette en mode développeur (Paramètres -> A propos du téléphone -> Numéro de build) puis activer le débogage USB (Paramètres -> Options pour les développeurs).

Brancher le smartphone ou la tablette via le port USB (ici un Acer Z120) et vérifier qu'il est détecté par le système :

```
$ dmesg
...
[72893.663574] usb 3-9.4.2: New USB device found, idVendor=0502, idProduct=3472
[72893.663581] usb 3-9.4.2: New USB device strings: Mfr=2, Product=3, SerialNumber=4
[72893.663584] usb 3-9.4.2: Product: MT65xx Android Phone
[72893.663587] usb 3-9.4.2: Manufacturer: MediaTek
[72893.663589] usb 3-9.4.2: SerialNumber: AAEEQCBQAYRCRWDQ
[72893.664509] scsi14 : usb-storage 3-9.4.2:1.0
[72894.660646] scsi 14:0:0:0: Direct-Access Linux File-CD Gadget 0000 PQ: 0 ANSI: 2
[72894.660945] scsi 14:0:0:1: Direct-Access Linux File-CD Gadget 0000 PQ: 0 ANSI: 2
[72894.662178] sd 14:0:0:0: Attached scsi generic sg10 type 0
[72894.662552] sd 14:0:0:1: Attached scsi generic sg11 type 0
[72894.663606] sd 14:0:0:0: [sdj] Attached SCSI removable disk
[72894.663818] sd 14:0:0:1: [sdk] Attached SCSI removable disk

$ lsusb
...
```

```
Bus 003 Device 016: ID 0502:3472 Acer, Inc.
```

Sous Linux, il est nécessaire de créer un fichier de règles *udev* qui contient une configuration USB pour chaque type de périphérique réel. En tant que root, créer le fichier `/etc/udev/rules.d/51-android.rules` et y inscrire la ligne suivante (en précisant en hexadécimal votre *idVendor*) :

```
$ sudo vim /etc/udev/rules.d/51-android.rules
SUBSYSTEM=="usb", ATTR{idVendor}=="0502", MODE="0666"
```

Brancher le téléphone et vérifier qu'il est reconnu :

```
$ adb devices
List of devices attached
AAEEQCBQAYRCRWDQ    device
```

Commandes

Les principales commandes sont :

- Lancer Android Studio : `studio.sh`
- Lancer Android SDK Manager : `android` ou `sdkmanager`
- Tester la connexion avec un périphérique Android : `adb devices`

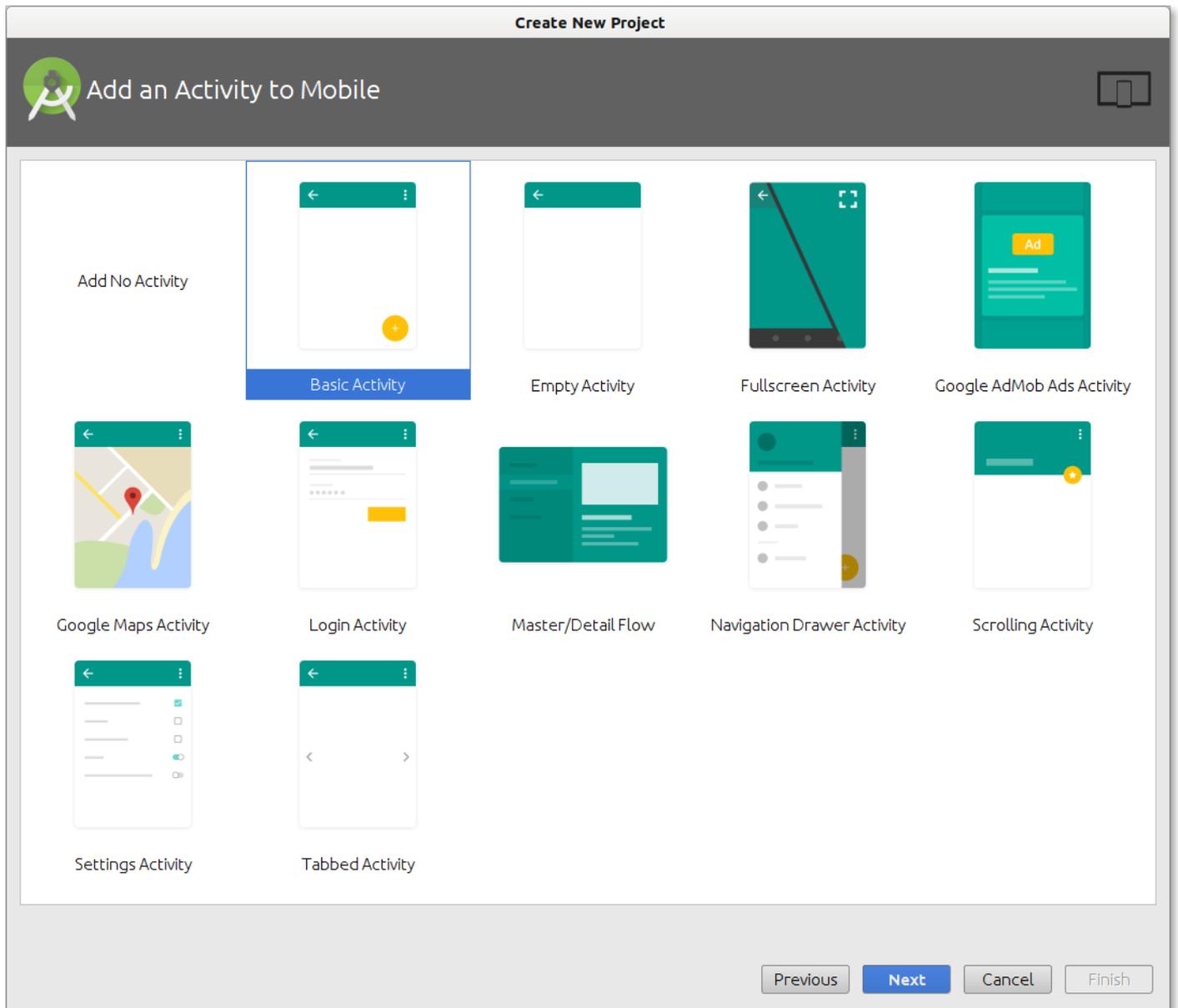
Première application

Il s'agit maintenant de tester la chaîne de développement installée avec une application minimale de type « Hello World »
...

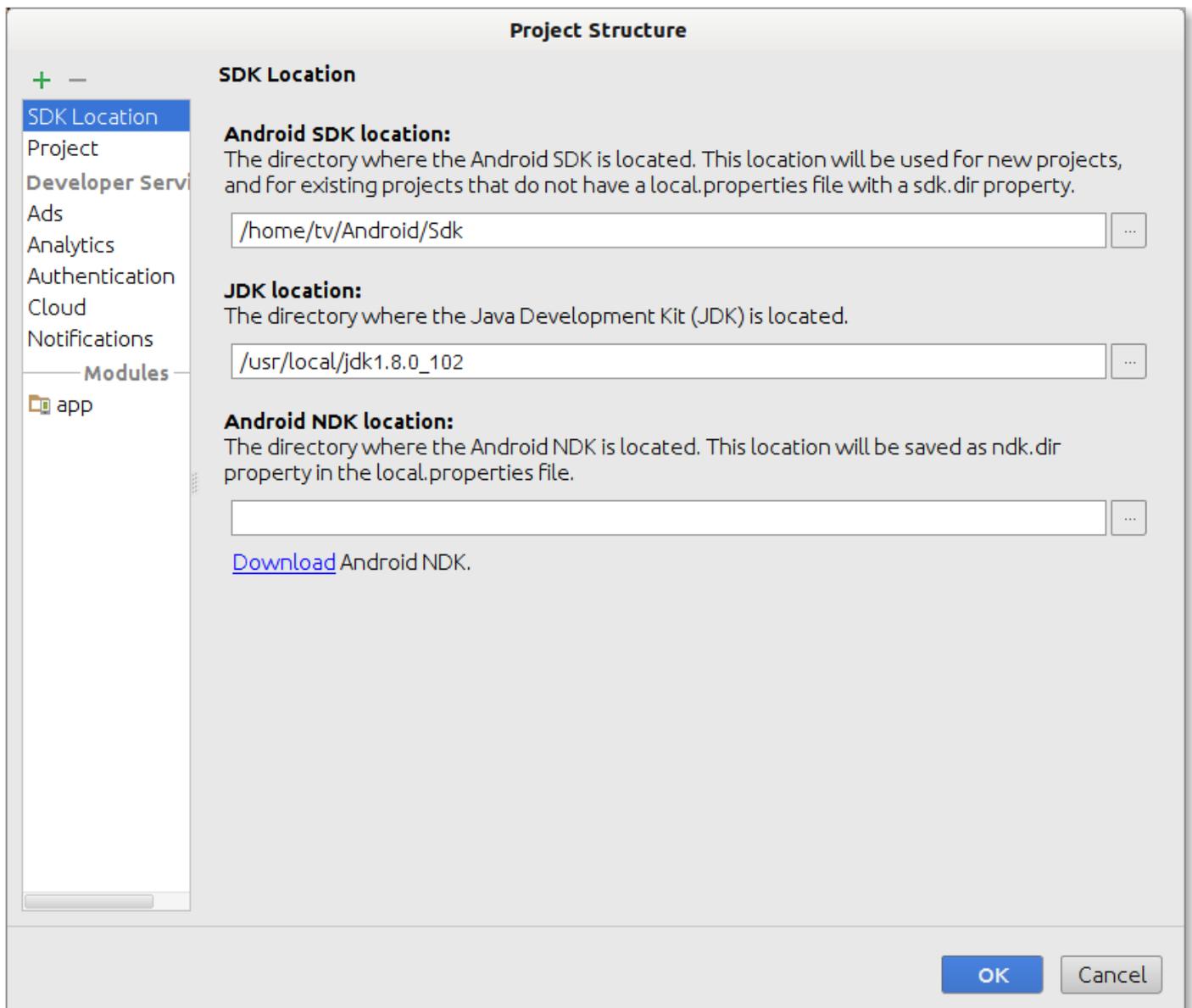
Démarrer Android Studio :

```
$ studio.sh
```

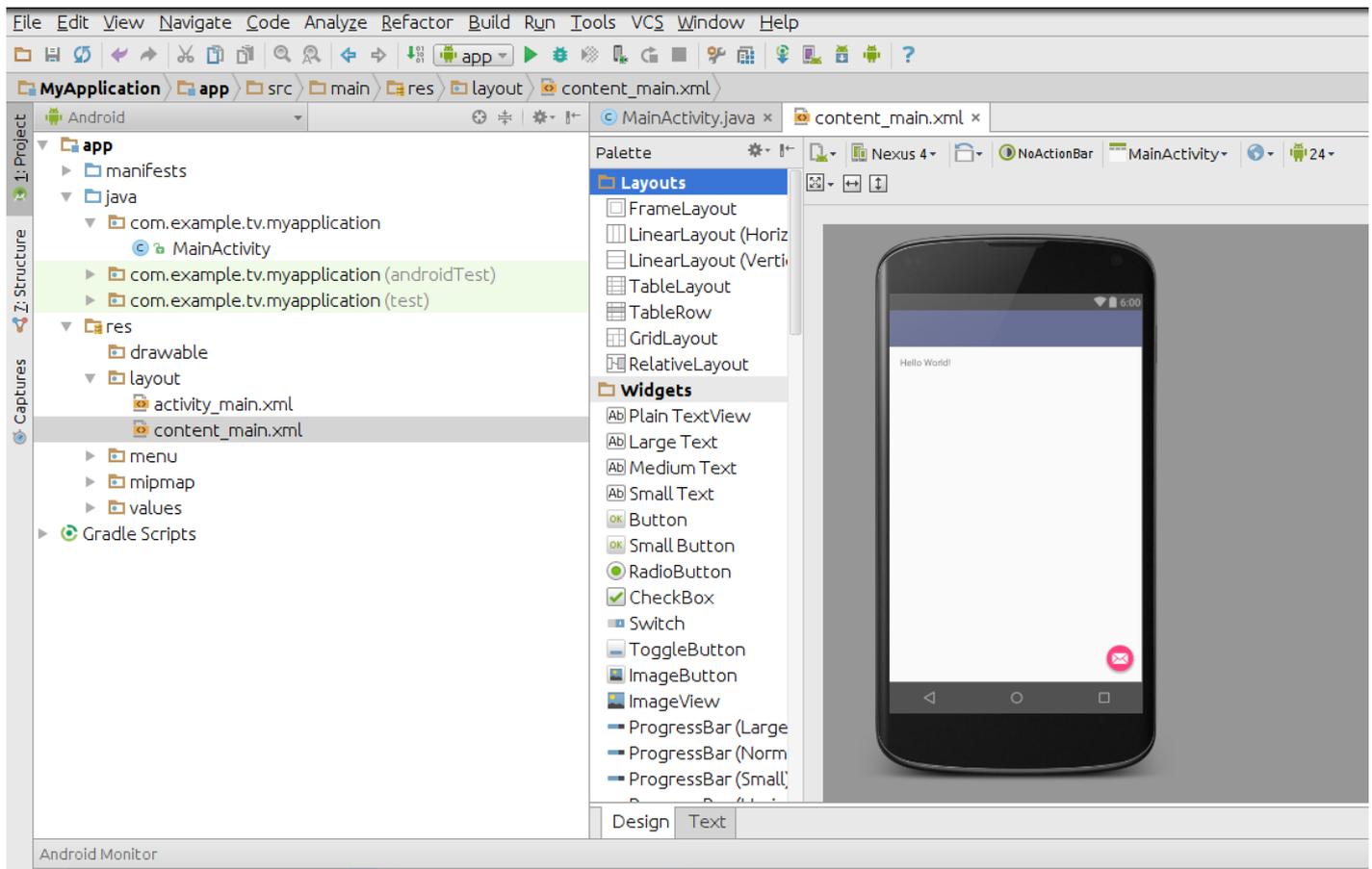
Choisir 'Basic Activity' :



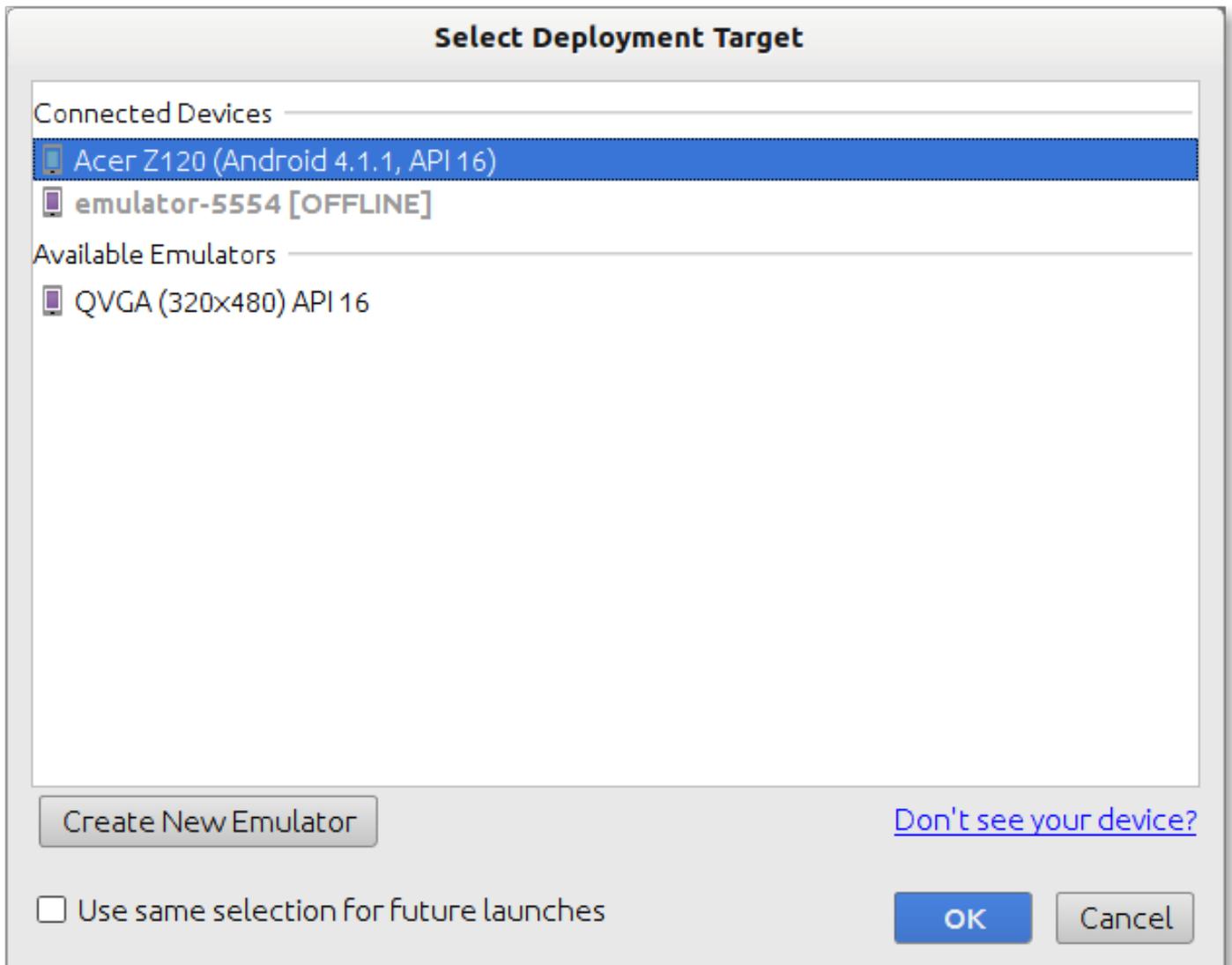
Configurer le chemin du Java SDK :



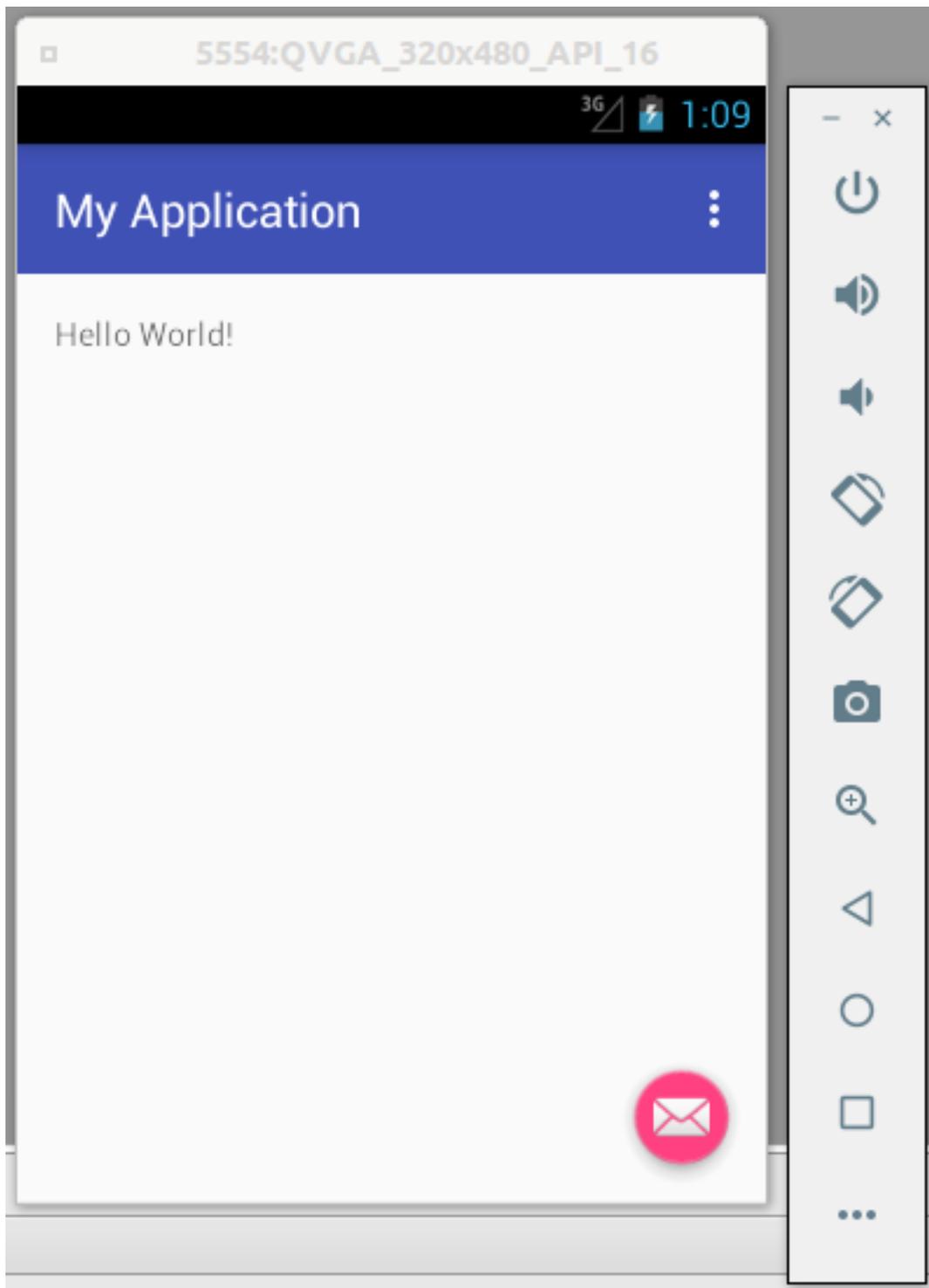
L'environnement Android Studio :



Cliquer sur 'Run app' :



Pour la cible, vous avez le choix entre le *smartphone* (ou la tablette) connecté ou un émulateur (cliquer sur 'Create New Emulateur' pour en créer un) :



Par défaut, le dossier de l'application est situé dans `$HOME/AndroidStudioProjects`.

Les fichiers "inutiles" (à ne pas mettre notamment dans un dépôt svn subversion) sont :

- le dossier `.gradle`
- les dossiers `build` et `app/build`
- le fichier `local.properties`
- le fichier `workspace.xml`
- le dossier `.idea/libraries`
- les fichiers `*.iml` et `app/*.iml`

Voici un script pour les supprimer :

```
#!/bin/bash
```

```
if [ $# -lt 1 ]
then
    echo "Usage: $0 <dossier>"
    exit
fi

dossier=$1

if [ ! -d $dossier ]
then
    echo "Erreur: $dossier introuvable"
    exit
fi

rm -rf $dossier/.gradle
rm -rf $dossier/build
rm -rf $dossier/app/build
rm -f $dossier/local.properties
rm -f $dossier/.idea/workspace.xml
rm -rf $dossier/.idea/libraries
rm -f $dossier/*.iml
rm -f $dossier/app/*.iml
```

Documentation

– [Introduction Android Studio](#)

[Retour](#)