

TD/TP 1

Introduction au SDK d'Android

Romain Raveaux

1 Introduction

Android est un système d'exploitation pour téléphone portable de nouvelle génération développé par Google.

Celui-ci met à disposition un kit de développement (SDK) basé sur le langage Java. Ce TP va vous expliquer comment installer ce SDK et vous présenter un exemple d'utilisation au travers le développement d'une application de type "HelloWorld" ainsi que d'applications plus complexes.

Démarrer la machine sous Ubuntu (la machine Virtuelle VMWare s'appelle TP Dev mobile).

Tout ce dont vous avez besoin pour ce TP se trouve sur </home/polytech/dev>

Vous trouverez **Eclipse** prêt à fonctionner (</home/polytech/eclipse>).

Créer un répertoire « **code**» pour y placer vos développements.

Le SDK est déjà installé et à jour !!!! mais voici la démarche à suivre si vous souhaitez l'installer chez vous.

2 Installation

2.1 Installation du SDK

Basé sur le langage Java, le SDK Android nécessite d'avoir un compilateur JAVA. Un JDK (6 ou 7) doit être installé sur sa machine.

JDK disponible :

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Le SDK Android est disponible :

- <http://developer.android.com/sdk/index.html>

Sous windows :

Vous n'avez qu'à suivre les instructions de l'installateur.

Sous Linux :

Editez votre fichier `~/.bash_profile` ou `~/.bashrc` et cherchez la ligne définissant le PATH afin d'y ajouter le chemin `<android_sdk>/tools`. Si cette ligne n'existe pas, ajoutez celle-ci : `export PATH=${PATH}:<android_sdk>/tools`. Vous pouvez également taper directement cette commande dans une fenêtre shell mais celle-ci ne sera alors valide que pour ce shell (et ces fils). Il vous faudra donc ressaisir la commande si vous fermez votre shell.

Sous Max :

Ça marche aussi. Vous n'avez qu'à suivre les instructions de l'installateur.

2.2 Eclipse

Il y a plusieurs solutions pour développer simplement avec les outils fournis par Google et son SDK Android : Soit utiliser Android Studio fondé sur [IntelliJ IDEA](#), Soit utiliser l'IDE Eclipse, soit utiliser Gradle avec ou sans IDE (un builder fusionnant en quelque sorte Ant et Maven).

Dans ce TD/TP nous allons nous limiter à Eclipse. Le plugins Eclipse pour Android s'appelle ADT. Les développeur Android de chez Google utilise Android studio pour développer.

2.2.1 Installation d'Eclipse

Eclipse est prêt à l'emploi et installé sur votre machine : [/home/polytech/dev/eclipse](#)
Sinon, rendez vous à l'adresse <http://www.eclipse.org/downloads/> et téléchargez la version « Eclipse IDE for Java Developers ». Extrayez l'archive et exécutez Eclipse.

Si vous êtes derrière un proxy avec ou sans authentification vous devez paramétrer Eclipse : Eclipse/Windows > Preferences > general > Network Connections

Active Provider : Manual

Remplissez les variables HTTP et HTTPS

Proxy by pass : Par exemple : « *.univ-tours.fr » pour éviter d'utiliser le proxy pour les adresses internes à l'ULR.

Adresse du proxy de l'ULR : <http://wwwcache.univ-tours.fr>

Port du proxy : 3128

2.3 Plugin Eclipse

Sachez que Google fournit un plugin pour cet IDE nommé Android Development Tools (ADT), sur votre machine, **le plugin ADT est déjà installé !!!** Mais voici la démarche pour l'installer chez vous :

Voici la marche à suivre pour installer ce plugin sur Eclipse :

- Démarrez Eclipse puis sélectionnez le menu Help > Software Updates
- Dans la fenêtre qui vient de s'ouvrir, cliquez sur l'onglet Available Software.
- Cliquez sur Add Site...
- Dans la boîte de dialogue qui apparaît, indiquez un nom (par exemple Android Plugin) et l'URL <https://dl-ssl.google.com/android/eclipse/> . Appuyez sur le bouton OK. (en cas de problème, utilisez l'adresse <http://dl-ssl.google.com/android/eclipse/>)
- Dans l'onglet Available Software, sélectionnez Developer Tools et cliquez sur Install...
- Finissez l'installation puis redémarrez Eclipse.
- Une fois Eclipse redémarré, sélectionnez le menu Window > Preferences... (ou Eclipse > Preferences si vous êtes sous Linux).
- Sélectionnez Android dans le panel de gauche.
- Indiquez le chemin où vous avez installé le SDK Android (bouton Browse pour parcourir le système de fichier).
- Appuyez sur le bouton OK.

3 Android SDK Manager

Android SDK manager est fourni par google est contient plusieurs éléments :

- Un compilateur JAVA pour Android (l'outil dx).
- Un logiciel de virtualisation pour les émulateurs Android
- Des exemples de projets avec des codes sources
- Un outils de communication entre l'ordinateur et l'émulateur
- Les images (iso) des différentes versions d'Android
- Un gestion de versions du SDK

Démarrer un terminal de commande et exécuter le gestionnaire de versions du SDK.

Le répertoire `~/dev/android-sdk-linux/tools` et l'exécutable `./android &`

Question 1 : Identifier les éléments installer sur votre machine ? Par contre de rien mettre à jour cela prend du temps.

4 Prise en main de l'émulateur (AVD)

AVD est le nom de l'émulateur Android. Il existe un outil pour gérer les AVD. Cet outil s'appelle l'AVD Manager. Pour y accéder démarrer le SDK manager et aller dans le menu « Manager AVD ».

Question 2 : Créer un AVD et identifier les points configurables de l'AVD. A la fin détruire votre AVD

Noter qu'il y a des AVD génériques : Nexus, Android Wear, ..

Noter qu'il y a déjà une AVD de créer. Ce dernier s'appelle « testphone ».

Question 3 : Décrire la configuration de l'AVD « testphone » (version d'Android, matériel, RAM, ...)

Question 4 : Lancer l'AVD « testphone » via le terminal de commande et l'outil « emulator »
`~/dev/android-sdk-linux/tools$./emulator`

5. Les outils du SDK

- Allez regarder dans le répertoire <AndroidSDK>/platforms-tools (<AndroidSDK> = /home/polytech/dev/android-sdk-linux)
 - Lancer dans un terminal le fichier **adb**
 - A quoi sert **adb** ?
 - Arrêter et démarrer le server ADB. **Cela peut être pratique lorsque parfois Eclipse ne « voit » plus votre émulateur comment étant actif.**
 - Lister les périphériques connectés
 - Lancer la commande adb shell. Essayer quelques commandes Linux (ls, pwd, who, id, ...) . Essayer certains appels systèmes propre à Android :
 - o service check phone
 - o service call phone 2 s16 "+332....."
 - o service list
 -
 - Regarder les autres commandes possibles avec ADB (Par exemple, l'installation d'APK, la désinstallation)
- Regarder le répertoire : /home/polytech/.android/avd
 - o Que contient ce répertoire ?

6. Première application avec Android : Helloworld

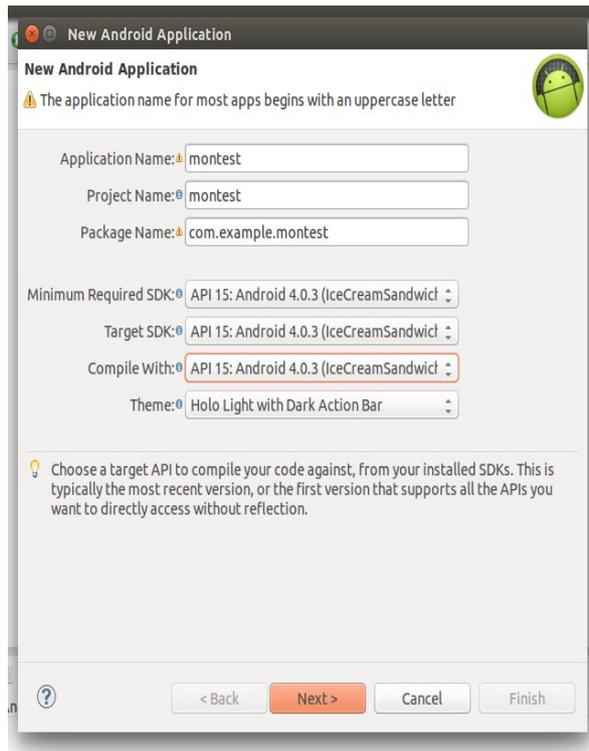
6.1 Développement avec le plugin Eclipse

Voici les étapes pour développer une première application avec Eclipse et le plugin Android Development Tools. Il faut déjà lancer eclipse. Pour se faire taper dans le terminal la commande suivante :

```
polytech@polytech-VirtualBox:~/dev/eclipse$ ./eclipse &
```

Question 5 : Commencez par sélectionner le menu File > New > Project... puis Android > Android project.

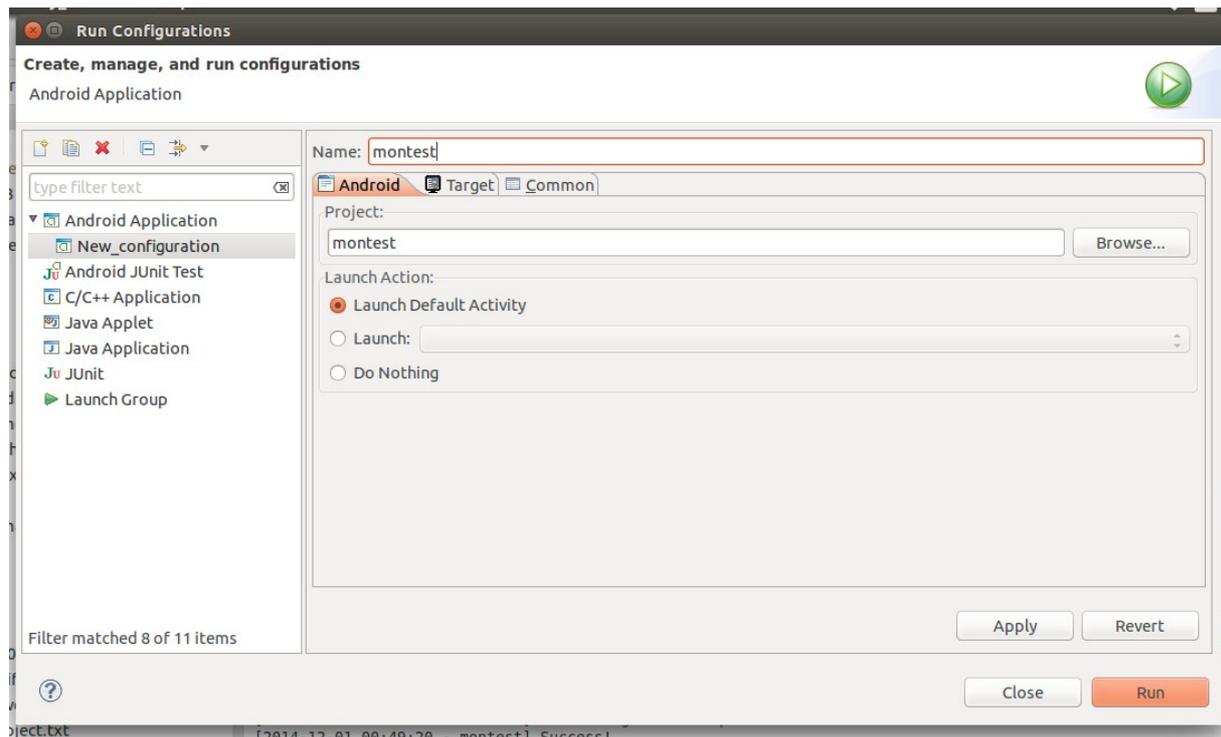
Dans un premier temps, laissez les paramètres par défaut, ne remplissez que ce qui est nécessaire (nom de l'application) .



Indiquez les propriétés du projet puis appuyez sur le bouton Next :

Compile with : Le code sera compilé avec les API de la version Android 4.0 dans votre cas.
Min SDK Version : C'est la version minimum d'Android nécessaire pour faire tourner votre application. A vous de vous assurer que vous n'utilisez pas de fonctions non-supportées par la version minimum d'Android que vous avez définie.

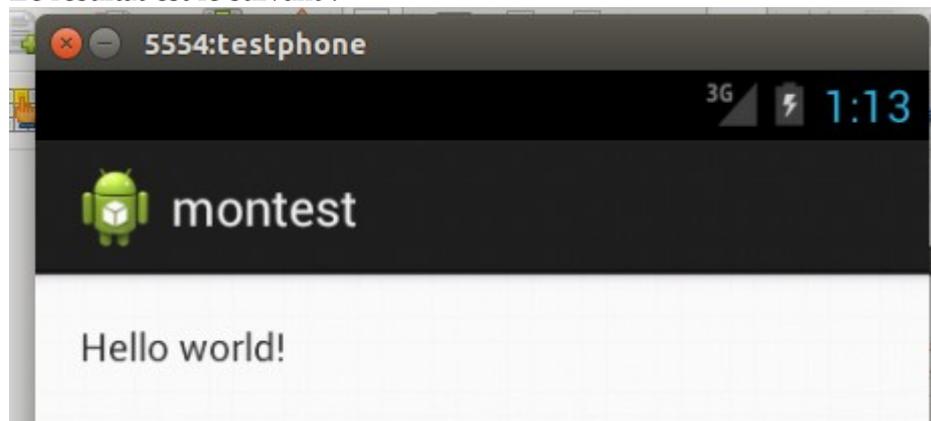
Pour exécuter l'application, sélectionnez le menu Run > Run Configurations.
Dans la boîte de dialogue qui s'ouvre, double cliquez sur l'entrée Android Application.
Un nouveau lanceur nommé New_configuration doit apparaître, sélectionnez le et renseignez le panel de droite :



Regarder l'onglet « Target » et choisissez votre AVD.

Il ne vous reste plus qu'à lancer l'application en appuyant sur le bouton Run.

Le résultat est le suivant :



6.2 Petite modification de l'application

Dans le package explorer sélectionner le fichier `com.example.montest.MainActivity.java`

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

Commençons notre HelloWorld en modifiant ce fichier afin d'afficher du texte à l'écran (nous détaillerons le code ensuite) :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    TextView text = new TextView(this);
    text.setText("Salut");
    setContentView(text);
}
```

Dans Android, tous les composants graphiques (bouton, animation, champ texte, etc) sont basés sur la classe View.

Ainsi un champ texte est représenté par la classe TextView.

```
TextView tv = new TextView(this);
```

L'argument du constructeur de la classe TextView est une instance de la classe Context qui fournit entre autre des services tel que la récupération des ressources, des accès base de données et des préférences.

Comme notre classe MainActivity hérite d'Activity qui elle-même hérite de Context, nous pouvons passer la référence 'this' au TextView.

Une fois le TextView instancié, il faut lui spécifier le texte à afficher : `tv.setText("Salut »);`

La dernière étape est d'afficher le TextView à l'écran : `setContentView(tv);`

7. La perspective DDMS (Dalvik Debug Monitor Server)

Dans Eclipse : Menu -> Windows -> Open Perspective -> Other -> DDMS

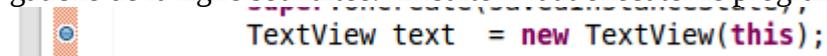
Regarder et indiquer l'utilité des Panels suivants :

- Devices
- Emulator Control
- Logcat
- File Explorer

a°) Pour écrire dans le Logcat des messages de debug il peut utiliser la commande suivant à la fin de la méthode onCreate :

```
Log.d(""+this.getComponentName(), "ça marche");
```

b°) Pour déboguer il suffit de mettre un point d'arrêt en double cliquant dans la colonne de gauche de la ligne souhaitée. Ensuite il faut exécuter le programme en mode débogue



```
TextView text = new TextView(this);
```

