

Introduction au SDK Android



Par Yann D'Isanto  

Date de publication : 11 février 2008

Dernière mise à jour : 11 février 2008

Ce tutoriel est une introduction au SDK Android. La première partie concerne son installation et la deuxième illustre la réalisation d'une petite application de type Helloworld.

I - Introduction.....	3
II - Installation.....	3
II-A - Installation du SDK.....	3
II-B - Plugin Eclipse.....	3
III - Première application avec Android : Helloworld.....	4
III-A - Développement avec Ant.....	4
III-B - Développement avec le plugin Eclipse.....	7
IV - Remerciements.....	9
V - Téléchargements.....	9
VI - Liens.....	9


I - 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 tutoriel va vous expliquer comment installer ce SDK et vous présenter un exemple d'utilisation au travers le développement d'une application de type "HelloWorld".

II - Installation

II-A - Installation du SDK

 **Basé sur le langage Java, le SDK Android nécessite d'avoir un JDK (5 ou 6) installé sur sa machine pour pouvoir être utilisé.**

Le SDK est disponible en téléchargement pour les plateformes Linux, Mac et Windows à l'adresse suivante : <http://developer.android.com/sdk/index.html>

Décompressez l'archive zip ; vous obtenez alors un répertoire contenant le SDK nommé selon le format suivant : `android_sdk_<platform>_<release>_<build>`

Notez le chemin complet vers ce répertoire (nous y ferons référence par la suite sous la dénomination `<repertoire_sdk>`).

Afin de faciliter l'utilisation des outils du SDK, nous allons ajouter le sous-répertoire `tools` du répertoire `<repertoire_sdk>` dans la variable d'environnement `PATH` de votre système.

- **Linux** : éditez votre fichier `~/.bash_profile` ou `~/.bashrc` et cherchez la ligne définissant le `PATH` afin d'y ajouter le chemin `<repertoire_sdk>/tools`. Si cette ligne n'existe pas, ajoutez celle-ci : `export PATH=${PATH}:<repertoire_sdk>/tools`
- **Mac** : éditez le fichier `.bash_profile` présent dans votre répertoire personnel (ou créez-le s'il n'existe pas) puis procédez comme pour Linux
- **Windows** : Faites un clic droit sur le "Poste de travail" et sélectionnez "Propriétés". Dans l'onglet "Avancé" cliquez sur le bouton "Variables d'environnement". Une boîte de dialogue apparaît, double-cliquez sur l'entrée "Path" présente dans la partie "Variables Systèmes". Ajoutez-y le chemin `<repertoire_sdk>\tools`.


II-B - Plugin Eclipse

Si vous utilisez Eclipse pour vos développements, sachez que Google fournit un plugin pour cet IDE nommé *Android Development Tools* (ADT). Voici la marche à suivre pour installer ce plugin :

- Démarrez Eclipse puis sélectionnez le menu **Help > Software Updates > Find and Install...**
- Sélectionnez **Search for new features to install** et appuyez sur le bouton **Next**.
- Appuyez sur le bouton **New Remote 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**.
- Le site est ajouté à la liste et sélectionné. Appuyez sur le bouton **Finish**.
- Sélectionnez **Android Plugin** et appuyez sur le bouton **Next**.
- Lisez la licence d'utilisation, sélectionnez l'option **I accept the terms in the license agreement** et cliquez sur le bouton **Next**, puis sur le bouton **Finish**.
- Le plugin n'est pas signé, un message vous le signale, appuyez sur le bouton **Install All**.
- Redémarrez Eclipse.
- Une fois Eclipse redémarré, sélectionnez le menu **Window > Preferences...** (ou **Eclipse > Preferences** si vous êtes sous Mac OS X).
- 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**.

III - Première application avec Android : Helloworld

 Afin de pouvoir compiler les projets du SDK Android, il est nécessaire d'avoir l'outil **Ant** (version 1.6.5 ou plus pour Linux et Mac, version 1.7 ou plus pour Windows) ou l'**IDE Eclipse** (version 3.2, 3.3 plus le plugin ADT) installé sur votre système.

III-A - Développement avec Ant

Pour commencer, il est nécessaire de créer un projet Android. Pour cela il suffit d'utiliser le script *activityCreator* (activityCreator.py pour Linux/Mac, activityCreator.bat pour Windows) fournit par le SDK auquel on indique le nom du projet (via l'option *--out*) ainsi que sa classe principale.

```
activityCreator.py --out HelloWorld com.developpez.android.HelloWorld
```

Un répertoire portant le nom du projet (ici *HelloWorld*) est alors créé. Celui-ci contient l'arborescence suivante :

- **AndroidManifest.xml** : Le fichier manifest de l'application.
- **build.xml** : Un fichier xml pour compiler avec Ant.
- **res** : Le répertoire des ressources.
- **src** : Le répertoire des sources.
- **src/com/developpez/android/HelloWorld.java** : La classe HelloWorld.

Voici le fichier HelloWorld.java généré :

```

HelloWorld.java
package com.developpez.android;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorld extends Activity
{
    /** Called with the activity is first created. */
    @Override
    public void onCreate(Bundle icle)
    {
        super.onCreate(icle);
        setContentView(R.layout.main);
    }
}

```

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

```

HelloWorld.java
package com.developpez.android;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloWorld extends Activity
{
    /** Called with the activity is first created. */
    @Override
    public void onCreate(Bundle icle)
    {
        super.onCreate(icle);
    }
}

```

HelloWorld.java

```
TextView textView = new TextView(this);
textView.setText("Hello world !");
setContentView(textView);
}
```

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 *HelloWorld* 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("Hello, Android");
```


La dernière étape est d'afficher le *TextView* à l'écran :

```
setContentView(tv);
```

Il ne reste plus qu'à compiler et déployer notre application. La compilation s'effectue avec Ant dans le répertoire du projet.

Un répertoire bin est alors créé dans lequel vous trouverez notamment le fichier *HelloAndroid.apk* qui est le package Android de l'application (celui qui sera déployé).

Pour tester l'application, commencez par lancer l'émulateur avec la commande *emulator* et attendez l'apparition de la page d'accueil.

 Notez que le démarrage de l'émulateur peut être assez long, selon la machine de développement utilisée.

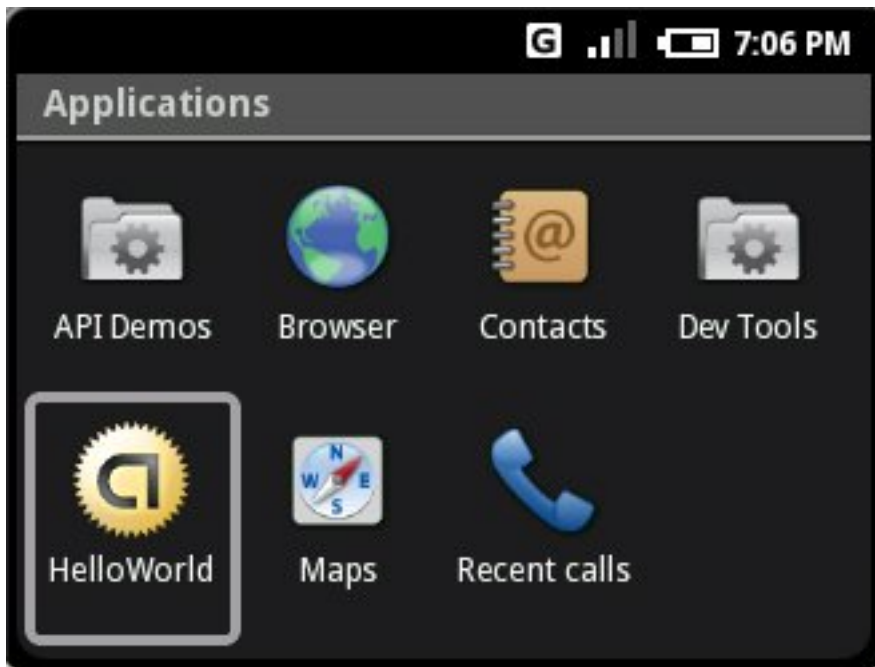
```
emulator
```

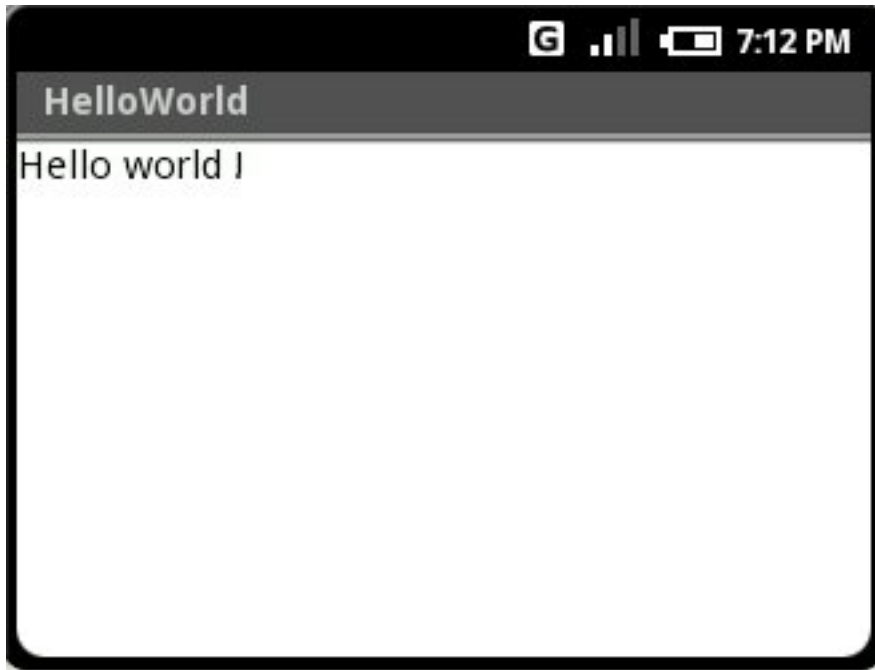


Le déploiement de l'application (le fichier HelloWorld.apk) s'effectue avec l'outil *adb*. Ainsi depuis la racine du projet :

```
adb install bin/HelloWorld.apk
```

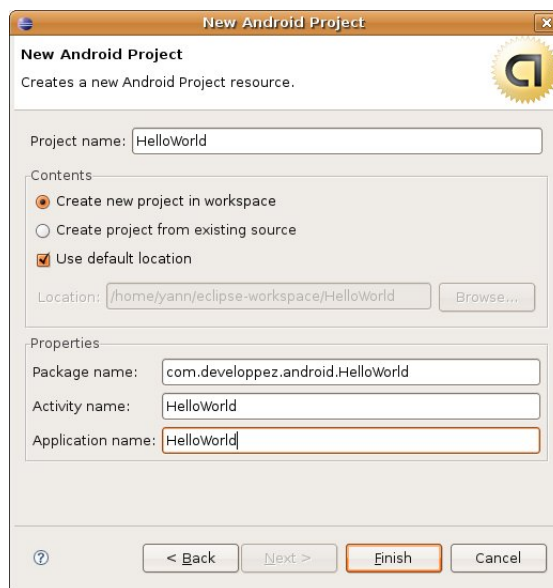
Une fois ceci fait, sélectionnez le menu "Applications" dans l'émulateur puis lancez l'application HelloWorld :





III-B - Développement avec le plugin Eclipse

Voici les étapes pour développer la même application avec Eclipse et le plugin *Android Development Tools*. Commencez par sélectionner le menu **File > New > Project...** puis **Android > Android project**. Indiquez les propriétés du projet puis appuyez sur le bouton **Finish** :



Sélectionnez le fichier HelloWorld.java dans l'arborescence de gauche qui contient le code suivant :

```

HelloWorld.java
package com.developpez.android;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorld extends Activity
{
    /** Called with the activity is first created. */

```

HelloWorld.java

```
@Override
public void onCreate(Bundle icicle)
{
    super.onCreate(icicle);
    setContentView(R.layout.main);
}
```

Modifier ce code comme suit :

HelloWorld.java

```
package com.developpez.android;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloWorld extends Activity
{
    /** Called with the activity is first created. */
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        TextView textView = new TextView(this);
        textView.setText("Hello world !");
        setContentView(textView);
    }
}
```

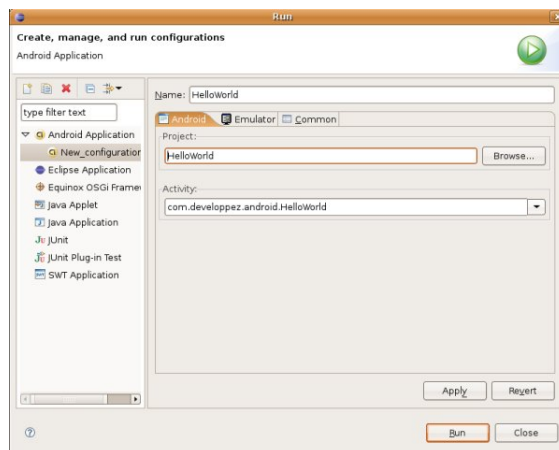


Les explications du code sont données dans le chapitre précédent.

Pour exécuter l'application, sélectionnez le menu **Run > Open Run Dialog...** (ou **Run > Run...**).

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 :



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



IV - Remerciements

Je voudrais remercier RideKick, keulkeul et Gfx pour leur contribution et leurs corrections.

V - Téléchargements

- **SDK Android**
- Projet Ant : **FTP (lien principal)**, **HTTP (lien de secours)**
- Projet Eclipse : **FTP (lien principal)**, **HTTP (lien de secours)**

VI - Liens

-  **Le site d'Android**
-  **Le site officiel d'Android pour les développeurs**