

# Android

INTRODUCTION



# Présentation

---



Système d'exploitation open source Développé en 2007 par une startup rachetée par Google.

Caractéristique du Système d'exploitation Android:

- Multi-utilisateur, chaque application est considérée comme un utilisateur différent.
- Android attribue à chaque application un user id unique.
- Utilise une machine virtuelle java nommée Dalvic.

Langages de développement

- Java
- C

Environnement de développement

- Android Studio (basé sur IntelliJ IDEA)
- ADT: Android Development Tools (plugin Eclipse)
- IntelliJ
- Netbeans.

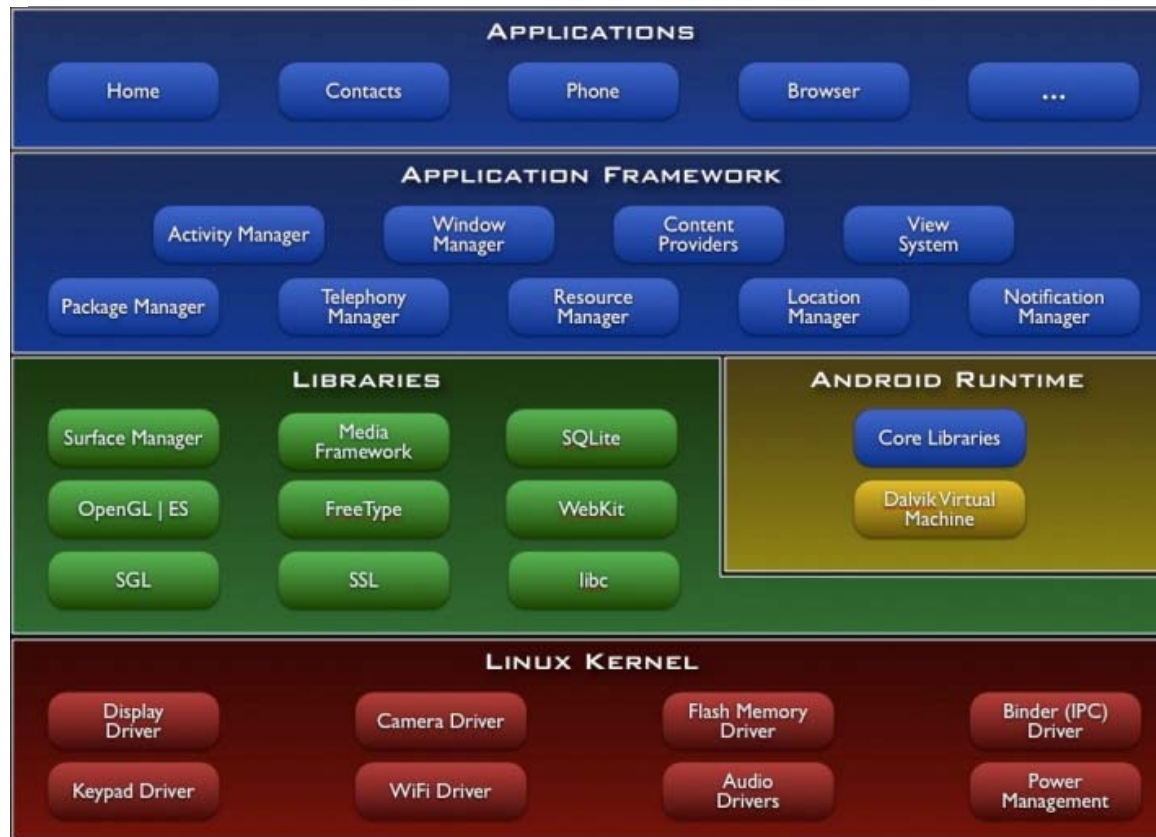
# Open Handset Alliance

Android est développé par le consortium OHA ( formé en 2009 à l'initiative de Google).

Le consortium regroupe 84 compagnies.



# Architecture Android



## Le noyau Linux:

- Gestion de la mémoire
- Gestion des pilotes
- Gestionnaire des processus
- Réseau

## Bibliothèques

- Bibliothèques C/C++ accessibles par des interfaces Java.

# Versions Android



Apple pie 1.1 (2007)



Bananas split 1.2  
HTC G1 (Dream 2008)



Cupcake 1.5



Donut 1.6



Eclair 2.0 2.1



Froyo 2.2.3



Gingerbread 2.3.7



Honeycomb 3.2



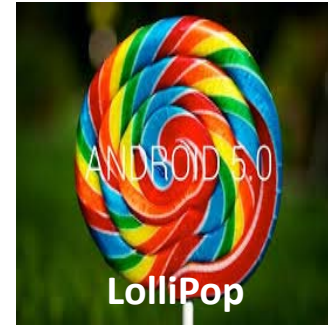
Ice Cream Sandwich  
4.0.4



Jelly Bean 4.3



KitKat 4.4



Lollipop

# Structure d'un projet Android

---

▲ a1	Le manifest
▲ src	
▲ com.example.a1	Les ressources
▶ Actvit2.java	
▶ MainActivity.java	Les activités
▶ gen [Generated Java Files]	
▶ Android 4.4	Les vues
▶ Android Private Libraries	
▶ assets	
▶ bin	
▶ libs	
▶ res	
▶ AndroidManifest.xml	
▶ ic_launcher-web.png	
▶ proguard-project.txt	
▶ project.properties	

# Le manifest

## Android Manifest

### Manifest General Attributes

Defines general information about the AndroidManifest.xml

Package	<input type="text" value="com.example.a1"/>
Version code	<input type="text" value="1"/>
Version name	<input type="text" value="1.0"/>
Shared user id	<input type="text"/>
Shared user label	<input type="text"/>
Install location	<input type="text"/>

### Manifest Extras

U S P U C U P S L O Az

<input checked="" type="checkbox"/> Uses Sdk
--

Add...  
Remove...  
Up  
Down

### Attributes for Uses Sdk

The [uses-sdk](#) tag describes the minimum SDK version containing package manifest correctly.

Min SDK version	<input type="text" value="8"/>
Target SDK version	<input type="text" value="18"/>
Max SDK version	<input type="text"/>

- Informations techniques :package de l'application, sa version, la version du SDK requise pour son exécution
- Informations à destination de l'utilisateur:l'icône, le nom de l'application, le thème utilisé
- les activités
- Informations utiles au système, telles que les messages auxquels peut répondre l'application (*Intent*), les permissions requises pour autoriser l'application à s'exécuter (accès réseau, accès aux fichiers, etc.)

# Le manifest

---

```
<?xml version="1.0" encoding="utf-8"?>

<manifest
xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.examples.exemple0"

    android:versionCode="1"

    android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8"

    android:targetSdkVersion="18" />

<application android:icon="@drawable/ic_launcher"

    android:label="@string/app_name" >

    <activity

android:name="com.examples.exemple0/MainActivity"

        android:label="@string/app_name" >

</activity> </application> </manifest>
```

- Le namespace est appliqué uniquement aux attributs et non aux éléments.
- L'attribut package n'est pas préfixé par android.
- Support plusieurs tailles d'affichage

```
<supports-screens

    android:smallScreens="false" <!-- Supérieure à 3" -->
    android:normalScreens="true" <!-- entre 3" et 4.5" -->
    android:largeScreens="true" <!-- entre 4.5" et 10" -->
    android:xlargeScreens="true" <!-- > 10" --> />
```

# Les ressources

---

Les ressources constituent tout ce qui est externe au code de l'application: textes, images, vidéos, chaînes de caractères, codes couleurs, menus, composants graphiques

- **Values:** le fichier strings.xml contient les valeurs textuelles, comme des messages utilisateurs, les intitulés des composants graphiques (<string>) ou des valeurs de couleur (<color>).

Dans le code source, nous accédons aux ressources à l'aide de la classe R générée automatiquement par le SDK.

- **Drawable:** contient les images (png,gif,jpg)
- **Layout:** contient les éléments de l'interface utilisateurs (Vues).

# Les activités

---

Une activité est la composante principale pour une application Android, elle représente un écran de l'application.

Une application doit comporter au moins une activité.



# Exemple

---

```
public class MainActivity extends
Activity {

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

        Button b1= (Button)
findViewById(R.id.b1);

        //View.OnClickListener est une
interface qui contient la méthode
onClick
```

```
        b1.setOnClickListener(new
View.OnClickListener() {

    @Override

    public void onClick(View arg0) {

        EditText t1=(EditText)
findViewById(R.id.t1);

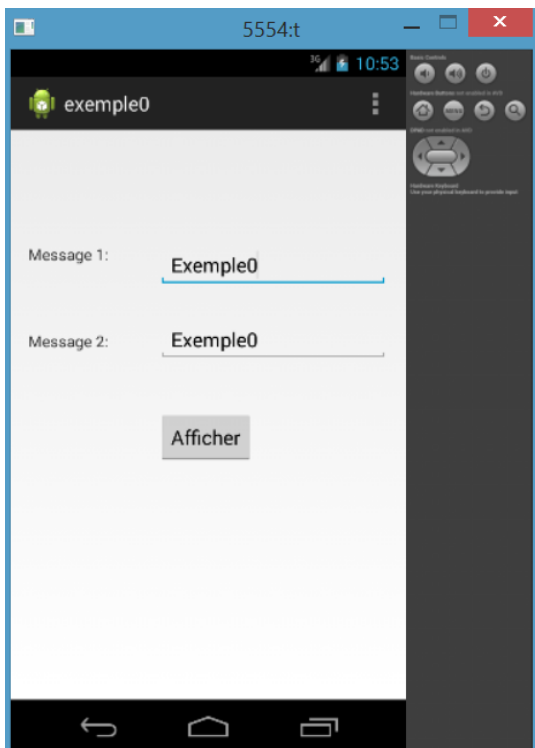
        EditText t2=(EditText)
findViewById(R.id.t2);

        t2.setText(t1.getEditableText());

    }

});
```

# Exemple



L'écran contient deux TextView, deux EditText et un Button, le clic sur le bouton button1 copie le contenu de la zone de texte textEdit1 dans textEdit2.

# La classe R

---

```
package com.example.exemple0;

public final class R {

...

public static final class id {

    public static final int TextView01=0x7f080002;

    public static final int button1=0x7f080004;

    public static final int editText1=0x7f080000;

    public static final int editText2=0x7f080003;

    public static final int textView1=0x7f080001;

}

public static final class layout {

    public static final int activity_main=0x7f030000;

}

...}
```

Générée et mise à jour automatique dans le dossier src/gen.

La classe permet l'accès aux ressources à partir du code java. Pour chaque ressource ajoutée, l'assistant génère un identifiant lui correspondant dans la classe R.

# la classe Activity

---

```
...  
  
/*Pour Button et EditText, les composants  
 * de l'interface utilisateur sont dans l'espace de nom  
 * android.Widget */  
  
import android.widget.*;  
  
/*L'interface View contient la méthode onClick*/  
  
import android.view.View;  
  
public class MainActivity extends Activity implements  
View.OnClickListener {  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
        // Récupérer le bouton d'après son id
```

```
        Button b = (Button) findViewById(R.id.button1);  
  
        /* associer la méthode onClick à l'événement click sur le bouton */  
  
        b.setOnClickListener(this);  
  
        @Override  
  
        public void onClick(View v) {  
  
            EditText t1 = (EditText) findViewById(R.id.t1);  
            EditText t2 = (EditText) findViewById(R.id.t2);  
            t2.setText(t1.getText());  
  
            @Override  
  
            public boolean onCreateOptionsMenu(Menu menu) {  
  
                // Inflate the menu; this adds items to the action bar if it is  
                present.  
  
                getMenuInflater().inflate(R.menu.main, menu);  
  
                return true;}}
```

- Activite2.java
    - MainActivity.java
  - gen [Generated Java Files]
  - Android 4.4
  - Android Private Libraries
  - assets
  - bin
  - libs
  - res
    - drawable-hdpi
    - drawable-ldpi
    - drawable-mdpi
    - drawable-xhdpi
    - drawable-xxhdpi
    - layout
      - activity\_activite2.xml
      - activity\_main.xml
    - menu
    - values
      - dimens.xml
      - strings.xml
      - styles.xml
    - values-sw600dp

**Resources Elements**

S
C
D
D
S
I
P
Az

- Ⓢ app\_name (String)
- Ⓢ action\_settings (String)
- Ⓢ hello\_world (String)
- Ⓢ title\_activity\_activite2 (String)
- Ⓢ valeurs (String Array)
    - Ⓛ Item
    - Ⓛ Item
    - Ⓛ Item
    - Ⓛ Item
    - Ⓛ Item
    - Ⓛ Item

Add...

Remove...

Up

Down

Ⓛ A string value to use

Value\*

# Les intentions (Intents)

---

Les objets de type **Intent** représentent des messages (*intentions*) émis par des applications, à destination des trois composants **Activity**, **Service** et **BroadcastReceiver**. C'est ce mécanisme qui est mis en œuvre pour naviguer au sein d'une application, pour appeler d'autres applications ou pour envoyer ou recevoir des messages système.

attributs:

- **ComponentName** : le nom du composant à qui est destiné directement le message. la méthode **setClassName** initialise cette valeur avec le nom de la classe destinataire complètement qualifié.
- **Action** : une chaîne de caractère qui décrit une action à exécuter, par exemple `android.intent.action.DIAL` (émettre un appel), `android.intent.action.WEB_SEARCH` (lancer une recherche google), etc
- **Data** : données qui accompagnent l'action à exécuter, sous la forme d'une URI et d'un type. Si l'action est `DIAL`, nous devons par exemple passer le numéro de téléphone à joindre; pour `WEB_SEARCH`, la chaîne de caractères à rechercher.
- **Extras** : paires de clés/valeurs transmises avec le message, sans format particulier.