

# Android - Les fondamentaux

Jérémie S. Cochoy

INRIA Paris-Saclay | jeremy.cochoy@gmail.com

Mars 2017

1 Applications

2 Le manifest

3 L'hello world Android

4 Conclusion

# La documentation

Votre nouveau livre de chevet.

*<https://developer.android.com/guide/index.html>*

# Qu'est-ce qu'une application ?

- Les applications android sont écrite en Java
- Le SDK créer un fichier APK (Android Package)

# Sandbox

- Un système multi-utilisateur, un user par app.
- Les fichiers de l'app ne sont accessibles que par cet user.
- Chaque processus a sa propre VM.

Pour accéder à d'autres fichiers, une app requiert des priviléges.

# Les composants

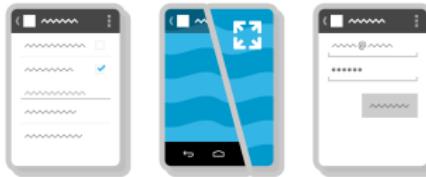


# Les composants

Les composants sont les blocks élémentaires. Certains sont les entrypoint de l'application. Il y à 4 type de composants :

- Activities
- Services
- Content providers
- Broadcast receivers

# Activities



Une activité est un écran avec une interface utilisateur.

Ex : liste des mails, affichage d'un e-mail, etc.

Une app peut lancer l'activité d'une autre app.

Ex : appareil photo.

Une activité est implémenté comme une sous classe d'Activity.

# Services



Un service est un composant qui s'exécute en arrière plan.  
Ex : musique, facebook messenger, etc.

Un service est une instance d'une sous classe de Service.

# Content providers



Gère un ensemble de données partagé entre des applications. FS, SQLite, Cloud...

Ex : Les contacts de l'utilisateur.

Un fournisseur de contenu est implémenté comme une sous classe de ContentProvider. Cette classe doit implémenter une API.

# Broadcast receiver

Un Broadcast receiver est un composant qui répond aux messages émis par le système, à l'intention de toute les applications. Une application peut aussi émettre un message.

Ex : Batterie faible, écran en veille, photo prise...

En général, un broadcast receiver est un composant léger dont le seul but est de lancer une autre tâche qui s'occupera du traitement (service, ou activité).

Un broadcast receiver est implémenté comme sous classe de BroadcastReceiver. Chaque message est délivré sous la forme d'un objet Intent.

# Appeler un composant



- Une app peut appeler le composant d'une autre app.
- Chaque composant s'exécute dans l'app à laquelle il appartient.
- Il n'existe donc pas de `main()` comme dans d'autres applications.

# A quoi sert le fichier manifest ?

- Liste les permissions requise pour exécuter l'application (liste de contactes, internet, appareille photo, ...)
- Déclare l'API minimal sous la quel l'application peux s'exécuter
- Déclare les fonctionnalités matériel requise/utilisé par l'application (bluetouth, multitouch, ...)
- Bibliothèques utilisés (ex : Google Maps library)
- Liste les composants de l'application
- et encore d'autres choses...

# Un manifest

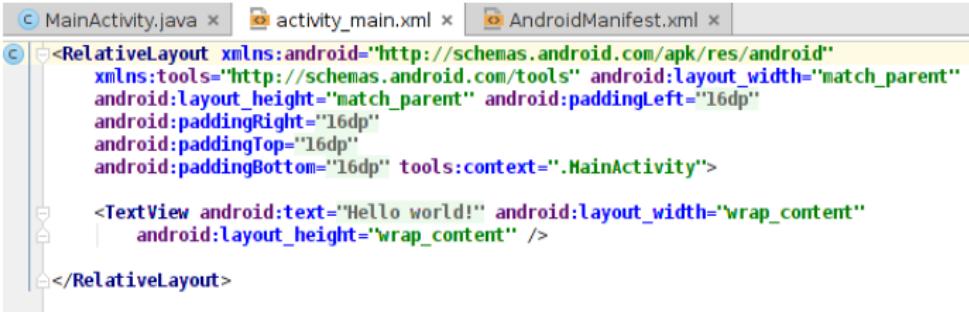
```
<?xml version='1.0' encoding='utf-8'?>
<manifest xmlns:android='http://schemas.android.com/apk/res/android'
    package='com.example.geometrica2.myapplication' >

    <application
        android:allowBackup='true'
        android:icon="@mipmap/ic_launcher"
        android:label='My Application'
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label='My Application' >
            <intent-filter>
                <action android:name='android.intent.action.MAIN' />

                <category android:name='android.intent.category.LAUNCHER' />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

# Positionnement des composants



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

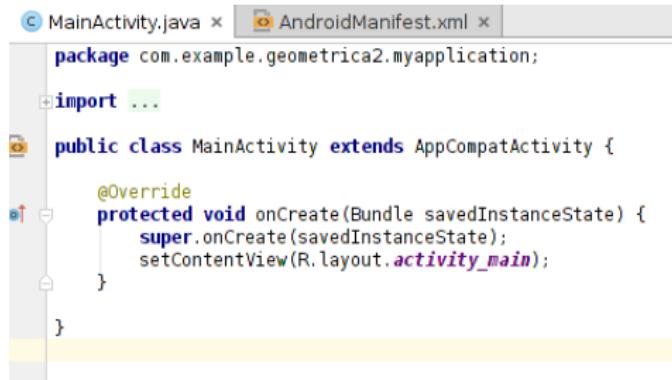
    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

La construction de l'UI se fait via un document XML

Les composants s'agencent "automatiquement" (Vertical / Horizontal Layout, gravity, match\_parent et wrap\_content).

# Une activité



The screenshot shows a code editor with two tabs: 'MainActivity.java' and 'AndroidManifest.xml'. The 'MainActivity.java' tab is active, displaying the following Java code:

```
package com.example.geometrica2.myapplication;

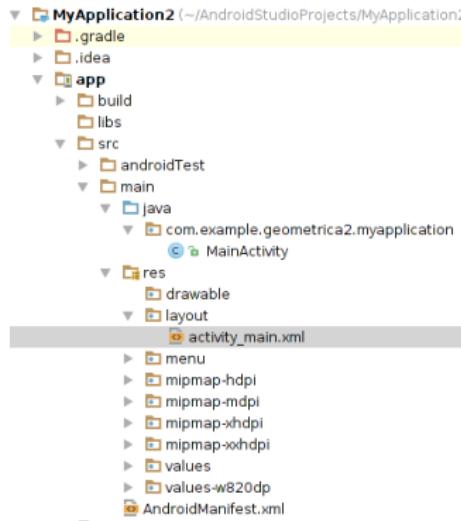
import ...

public class MainActivity extends AppCompatActivity {

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

Ici, l'activité "charge" l'interface depuis le fichier XML à sa création.

# Arborescence



On trouve le fichier manifest.xml, l'implémentation de l'activité.

Les ressources contiennent les textes, les interfaces (layout), les images (drawable)...

Assez parler...

Maintenant codons !

Pour commencer...

<https://developer.android.com/training/basics/firstapp/building-ui.html>

Pour me contacter : jeremy.cochoy@gmail.com, merci et à bientôt.

