

Développement Application Android

Introduction et Interface Utilisateur

Anthony Chomienne

CPE Lyon

 2020

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 INTRODUCTION DÉVELOPPEMENT MOBILE

- Des chiffres ...
- ... et des lettres
- Puis commencer

2 INTERFACE UTILISATEUR

3 DATA BINDING

4 MODEL VIEW VIEW-MODEL

5 ANDROIDMANIFEST.XML

6 GRADLE

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 INTRODUCTION DÉVELOPPEMENT MOBILE

- Des chiffres ...
- ... et des lettres
- Puis commencer

2 INTERFACE UTILISATEUR

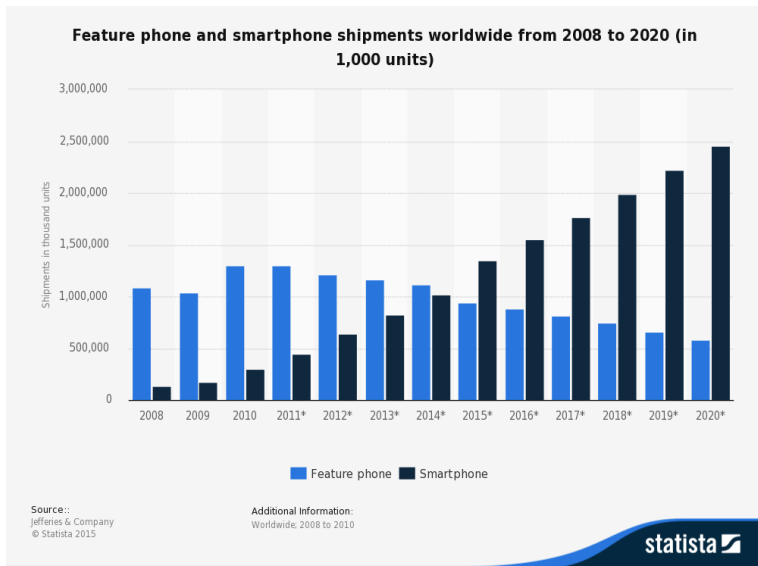
3 DATA BINDING

4 MODEL VIEW VIEW-MODEL

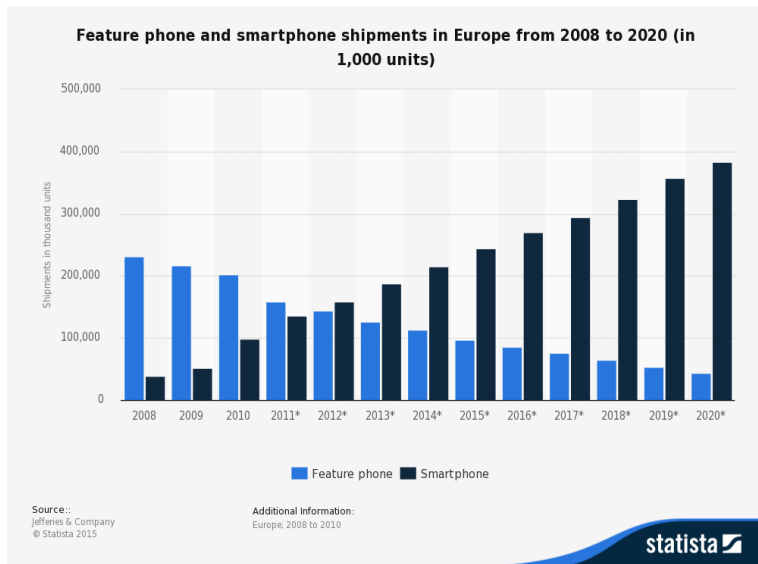
5 ANDROIDMANIFEST.XML

6 GRADLE

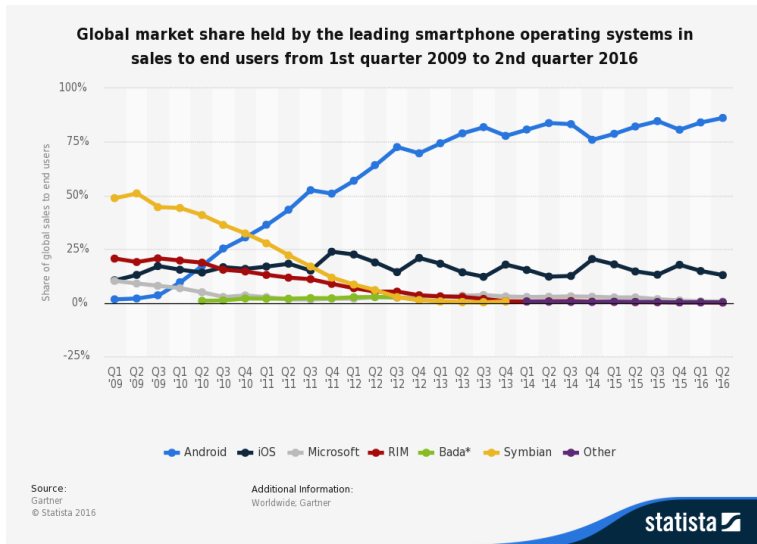
RÉPARTITION DES SMARTPHONES DANS LE MONDE



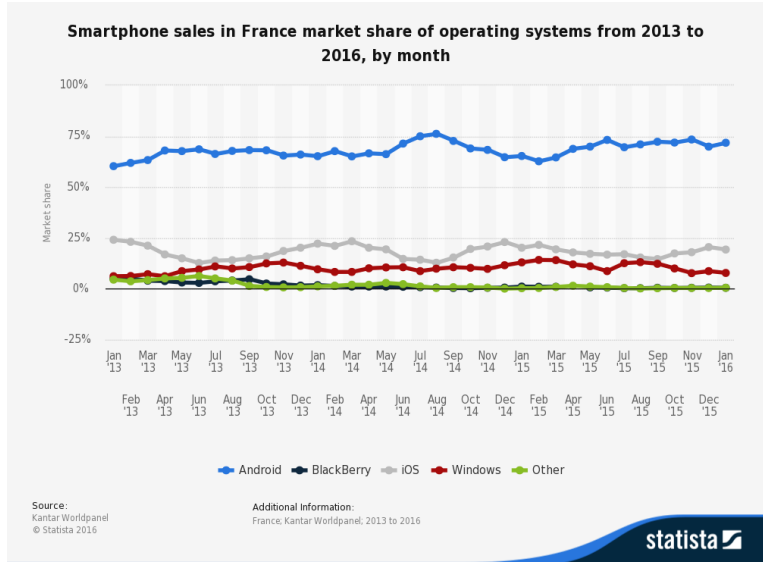
RÉPARTITION DES SMARTPHONES EN EUROPE



RÉPARTITION DES OS DANS LE MONDE



RÉPARTITION DES OS EN FRANCE



DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 INTRODUCTION DÉVELOPPEMENT MOBILE

- Des chiffres ...
- ... et des lettres
- Puis commencer

2 INTERFACE UTILISATEUR

3 DATA BINDING

4 MODEL VIEW VIEW-MODEL

5 ANDROIDMANIFEST.XML

6 GRADLE

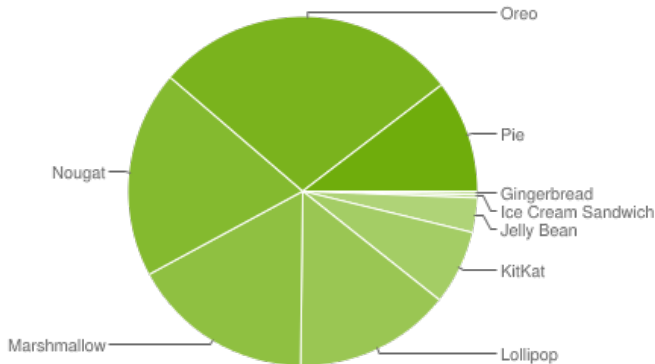
LES DIFFÉRENTES VERSIONS D'ANDROID



RÉPARTITION DES VERSIONS

Version	Codename	API	Distribution
2.3.3-2.3.7	Gingerbread	10	0.3%
4.0.3-4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.0		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

Data collected during a 7-day period ending on May 7, 2019.



DÉVELOPPEMENT APPLICATION ANDROID — PLAN

1 INTRODUCTION DÉVELOPPEMENT MOBILE

- Des chiffres ...
- ... et des lettres
- Puis commencer

2 INTERFACE UTILISATEUR

3 DATA BINDING

4 MODEL VIEW VIEW-MODEL

5 ANDROIDMANIFEST.XML

6 GRADLE

CONTRAINTES DU DÉVELOPPEMENT MOBILE

- Résolution et Taille d'écran

CONTRAINTES DU DÉVELOPPEMENT MOBILE

- Résolution et Taille d'écran
- Fragmentation des versions

CONTRAINTES DU DÉVELOPPEMENT MOBILE

- Résolution et Taille d'écran
- Fragmentation des versions
- Orientation

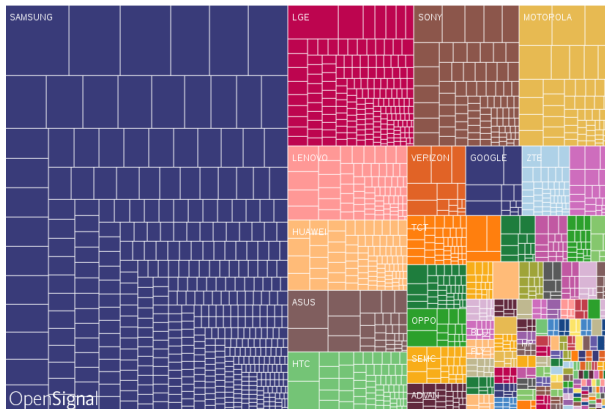
CONTRAINTES DU DÉVELOPPEMENT MOBILE

- Résolution et Taille d'écran
- Fragmentation des versions
- Orientation
- Usage de la Batterie

CONTRAINTES DU DÉVELOPPEMENT MOBILE

- Résolution et Taille d'écran
- Fragmentation des versions
- Orientation
- Usage de la Batterie
- Interactivité

RÉSOLUTION ET TAILLE D'ÉCRAN

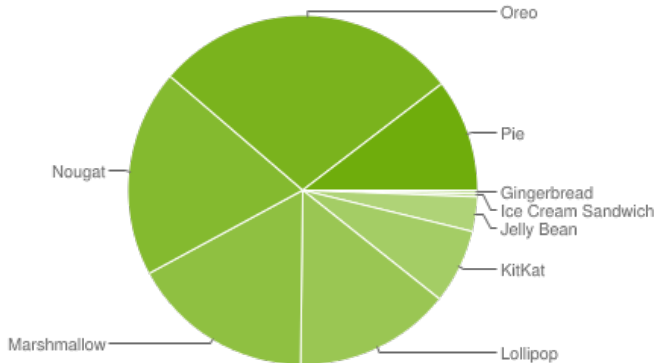


En 2015, il y avait: 1294 constructeurs
24093 devices
Samsung 37.8% des devices

RÉPARTITION DES VERSIONS

Version	Codename	API	Distribution
2.3.3-2.3.7	Gingerbread	10	0.3%
4.0.3-4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.0		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

Data collected during a 7-day period ending on September 12, 2018.



- Portait Uniquement ?

ORIENTATION

- Portait Uniquement ?
- Paysage Uniquement ?

ORIENTATION

- Portait Uniquement ?
- Paysage Uniquement ?
- Les deux ?

ORIENTATION

- Portait Uniquement ?
- Paysage Uniquement ?
- Les deux ?
- Dépendant du support ?

- Utilisation intensive du GPS ?

- Utilisation intensive du GPS ?
- Accès Réseau régulier ?

BATTERIE

- Utilisation intensive du GPS ?
- Accès Réseau régulier ?
- Wifi ?

- Utilisation intensive du GPS ?
- Accès Réseau régulier ?
- Wifi ?
- 3G/4G uniquement ?

BATTERIE

- Utilisation intensive du GPS ?
- Accès Réseau régulier ?
- Wifi ?
- 3G/4G uniquement ?
- Écran maintenu activé ?

- Utilisation intensive du GPS ?
- Accès Réseau régulier ?
- Wifi ?
- 3G/4G uniquement ?
- Écran maintenu activé ?
- Services en arrière plan ?

- Simple Touch / Multi Touch ?

- Simple Touch / Multi Touch ?
- appui long, appui court ?

- Simple Touch / Multi Touch ?
- appui long, appui court ?
- Bouton Physique ?

- Simple Touch / Multi Touch ?
- appui long, appui court ?
- Bouton Physique ?
- Clavier Physique ?

- Java et Kotlin

- Java et Kotlin
- Émulateur

SDK

- Java et Kotlin
- Émulateur
- Outils de bases

- Java et Kotlin
- Émulateur
- Outils de bases
- Nécessaire pour tout

- C et C++

- C et C++
- App à Usage CPU intensif

- C et C++
- App à Usage CPU intensif
- JNI

- C et C++
- App à Usage CPU intensif
- JNI
- Qt (apk pour un hello world \Rightarrow 8Mo)

- Cordova/Ionic

- Cordova/Ionic
 - Langage Web(html5, javascript,...)

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long

- Cordova/Ionic

- Langage Web(html5, javascript,...)
- Threading pas possible ou compliqué à mettre en place
- Traitement des données plus long
- Rarement Le look and feel du système

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin
 - C#

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin
 - C#
- Flutter

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin
 - C#
- Flutter
 - Dart

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin
 - C#
- Flutter
 - Dart
- Qt

- Cordova/Ionic
 - Langage Web(html5, javascript,...)
 - Threading pas possible ou compliqué à mettre en place
 - Traitement des données plus long
 - Rarement Le look and feel du système
 - Passe par des WebView
- Xamarin
 - C#
- Flutter
 - Dart
- Qt
 - C++ et QML

Quels sont vos critères ?

- Avoir la plus grande population ?

Quels sont vos critères ?

- Avoir la plus grande population ?
- Avoir moins de code à maintenir ?

Quels sont vos critères ?

- Avoir la plus grande population ?
- Avoir moins de code à maintenir ?
- Avoir un développement rapide ?

Quels sont vos critères ?

- Avoir la plus grande population ?
- Avoir moins de code à maintenir ?
- Avoir un développement rapide ?
- Faire un prototype ?

Quels sont vos critères ?

- Avoir la plus grande population ?
- Avoir moins de code à maintenir ?
- Avoir un développement rapide ?
- Faire un prototype ?
- ...

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR**
- 3 DATA BINDING
- 4 MODEL VIEW VIEW-MODEL
- 5 ANDROIDMANIFEST.XML
- 6 GRADLE
- 7 CONCLUSION

Architecture d'un projet

- AndroidManifest.xml

Architecture d'un projet

- AndroidManifest.xml
 - Description du projet

Architecture d'un projet

- AndroidManifest.xml
 - Description du projet
- java

Architecture d'un projet

- AndroidManifest.xml
 - Description du projet
- java
 - Code Java du projet

Architecture d'un projet

- AndroidManifest.xml
 - Description du projet
- java
 - Code Java du projet
- res

Architecture d'un projet

- AndroidManifest.xml
 - Description du projet
- java
 - Code Java du projet
- res
 - Fichier de ressources (layout, chaînes de caractères, images,...)

LAYOUT GÉNÉRIQUE

Layout vient du terme Layer (couche/strate)

- Lister des éléments

LAYOUT GÉNÉRIQUE

Layout vient du terme Layer (couche/strate)

- Lister des éléments
- Hiérarchiser des éléments

LAYOUT GÉNÉRIQUE

Layout vient du terme Layer (couche/strate)

- Lister des éléments
- Hiérarchiser des éléments
- Placer des éléments

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML

LAYOUT ANDROID

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML
- Peut être subdivisé en plusieurs fichiers

```
<include  
    layout="@layout/my_super_sub_layout"/>
```

LAYOUT ANDROID

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML
- Peut être subdivisé en plusieurs fichiers

```
<include  
    layout="@layout/my_super_sub_layout"/>
```

- Contient différentes View

LAYOUT ANDROID

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML
- Peut être subdivisé en plusieurs fichiers

```
<include  
    layout="@layout/my_super_sub_layout"/>
```

- Contient différentes View
- Décrit une Vue dans sa globalité

LAYOUT ANDROID

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML
- Peut être subdivisé en plusieurs fichiers

```
<include  
    layout="@layout/my_super_sub_layout"/>
```

- Contient différentes View
- Décrit une Vue dans sa globalité
- Peut être défini pour une orientation particulière

LAYOUT ANDROID

Se retrouve dans le dossier res/layout des projets Android

- Fichier XML
- Peut être subdivisé en plusieurs fichiers

```
<include  
    layout="@layout/my_super_sub_layout"/>
```

- Contient différentes View
- Décrit une Vue dans sa globalité
- Peut être défini pour une orientation particulière
- Peut être défini pour une taille d'écran particulière

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/
 - layout-land/ pour le mode paysage

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/
 - layout-land/ pour le mode paysage
 - layout-large/ pour les grands écrans

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/
 - layout-land/ pour le mode paysage
 - layout-large/ pour les grands écrans
 - layout-hdpi/ pour les écrans avec une densité élevée

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/
 - layout-land/ pour le mode paysage
 - layout-large/ pour les grands écrans
 - layout-hdpi/ pour les écrans avec une densité élevée
- pour voir l'ensemble des options fournies :

UNE GESTION POUR LES GOUVERNER TOUS

Pour avoir des layouts qui s'adaptent à l'orientation, la taille de l'écran il faut créer des sous répertoires.

- res/
 - layout/
 - layout-land/ pour le mode paysage
 - layout-large/ pour les grands écrans
 - layout-hdpi/ pour les écrans avec une densité élevée
- pour voir l'ensemble des options fournies :
- http://developer.android.com/guide/practices/screens_support.html#qualifiers

QUELQUES LAYOUTS: LINEARLAYOUT 1/4

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

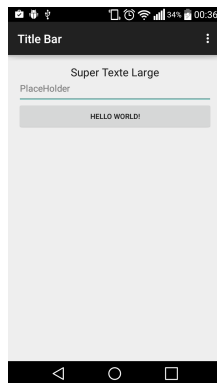
```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Super Texte Large"  
    android:layout_gravity=  
        "center_horizontal" />
```

```
<EditText
```

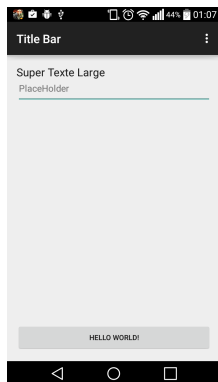
```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Placeholder" />
```

```
</LinearLayout>
```



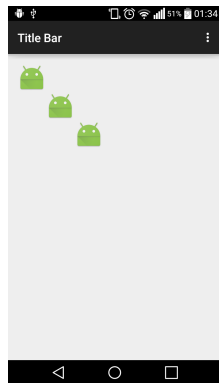
QUELQUES LAYOUTS: RELATIVELAYOUT 2/4

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/textview"
        android:layout_alignParentTop="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Super Texte Large"/>
    <EditText
        android:layout_below="@+id/textview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Placeholder" />
    <Button
        android:layout_alignParentBottom="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello World!"/>
</RelativeLayout>
```



QUELQUES LAYOUTS: GRIDLAYOUT 3/4

```
<GridLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="3"
    android:rowCount="3">
    <ImageView
        android:src="@mipmap/ic_launcher"
        android:layout_column="0"
        android:layout_row="0"/>
    <ImageView
        android:src="@mipmap/ic_launcher"
        android:layout_column="1"
        android:layout_row="1"/>
    <ImageView
        android:src="@mipmap/ic_launcher"
        android:layout_column="2"
        android:layout_row="2"/>
</GridLayout>
```



QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout
- GridLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout
- GridLayout
- FrameLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout
- GridLayout
- FrameLayout
- CoordinatorLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout
- GridLayout
- FrameLayout
- CoordinatorLayout
- ConstraintLayout

QUELQUES LAYOUTS: PETITE LISTE INCOMPLÈTE 4/4

- LinearLayout
- RelativeLayout
- GridLayout
- FrameLayout
- CoordinatorLayout
- ConstraintLayout
- ...

- Élément plus ou moins interactif

- Élément plus ou moins interactif
- Propriété commune : `android:layout_width` et `android:layout_height`
 - `match_parent`
 - `wrap_content`
 - taille en dp ex: "15dp" ou "0dp"

- Élément plus ou moins interactif
- Propriété commune : `android:layout_width` et `android:layout_height`
 - `match_parent`
 - `wrap_content`
 - taille en dp ex: "15dp" ou "0dp"
- Accessible depuis l'Activity ou le Fragment

- Élément plus ou moins interactif
- Propriété commune : `android:layout_width` et `android:layout_height`
 - `match_parent`
 - `wrap_content`
 - taille en dp ex: "15dp" ou "0dp"
- Accessible depuis l'Activity ou le Fragment
- Standard ou Personnalisé

- Élément plus ou moins interactif
- Propriété commune : `android:layout_width` et `android:layout_height`
 - `match_parent`
 - `wrap_content`
 - taille en dp ex: "15dp" ou "0dp"
- Accessible depuis l'Activity ou le Fragment
- Standard ou Personnalisé
- Différentes propriétés

QUELQUES VIEWS/WIDGETS UTILES

- Button, ImageButton, RadioButton, CheckBox
- ListView
- ProgressBar
- EditText, TextView
- Spinner
- DatePicker
- ...

UNE ACTIVITY...

- Élément central

UNE ACTIVITY...

- Élément central
- Presenter au sens MVP du terme

UNE ACTIVITY...

- Élément central
- Presenter au sens MVP du terme
- Accès au view du layout

```
public View findViewById(int);
```

UNE ACTIVITY...

- Élément central
- Presenter au sens MVP du terme
- Accès au view du layout

```
public View findViewById(int);
```

- Fait le lien avec les données

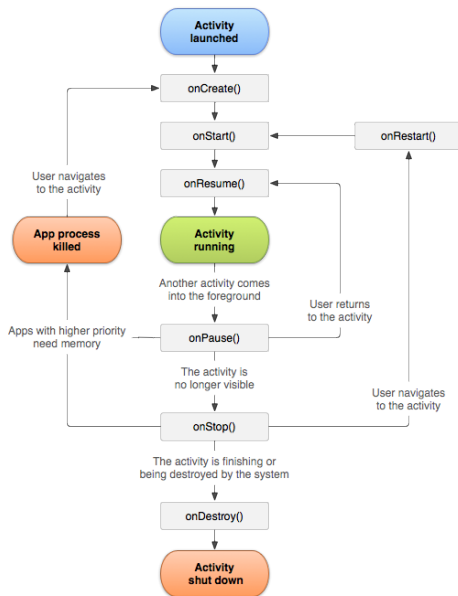
UNE ACTIVITY...

- Élément central
- Presenter au sens MVP du terme
- Accès au view du layout

```
public View findViewById(int);
```

- Fait le lien avec les données
- Géré par le thread graphique

..., SON CYCLE DE VIE...



..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`
- `protected void onStart();`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`
- `protected void onStart();`
- `protected void onResume();`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`
- `protected void onStart();`
- `protected void onResume();`
- `protected void onPause();`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`
- `protected void onStart();`
- `protected void onResume();`
- `protected void onPause();`
- `protected void onStop();`

..., SON CYCLE DE VIE...

Plusieurs États :

- `protected void onCreate(Bundle savedInstanceState);`
- `protected void onRestart();`
- `protected void onStart();`
- `protected void onResume();`
- `protected void onPause();`
- `protected void onStop();`
- `protected void onDestroy();`

... SES INTERACTIONS...

- Interaction avec les différentes Views au travers de plein de listeners :
- OnClickListener
- OnItemClickListener
- OnLongClickListener
- onTouchListener
- ...

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`
- `setResult(int result, [Intent data]);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`
- `setResult(int result, [Intent data]);`
- `finish();`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`
- `setResult(int result, [Intent data]);`
- `finish();`
- `finishActivity(int resultCode);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`
- `setResult(int result, [Intent data]);`
- `finish();`
- `finishActivity(int resultCode);`
- `onSaveInstanceState(Bundle bundle);`

... VERS D'AUTRES ACTIVITY.

- `startActivity(Intent intentToLaunch, [Bundle options]);`
- `startActivityForResult(Intent intentToLaunch, int requestCode, [Bundle options]);`
- `onActivityResult(int requestCode, int resultCode, Intent data);`
- `setResult(int result, [Intent data]);`
- `finish();`
- `finishActivity(int resultCode);`
- `onSaveInstanceState(Bundle bundle);`
- `onRestoreInstanceState(Bundle bundle);`

FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface

FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface
- Plus de fonctionnalité que la View

FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface
- Plus de fonctionnalité que la View
- Prend une partie du rôle de l'Activity

FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface
- Plus de fonctionnalité que la View
- Prend une partie du rôle de l'Activity
- Un ou plusieurs Fragments par Vue (MVP)

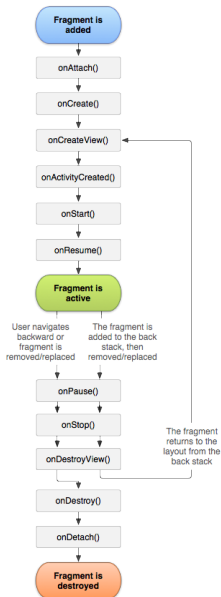
FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface
- Plus de fonctionnalité que la View
- Prend une partie du rôle de l'Activity
- Un ou plusieurs Fragments par Vue (MVP)
- Layout différent en fonction du support pour un même code

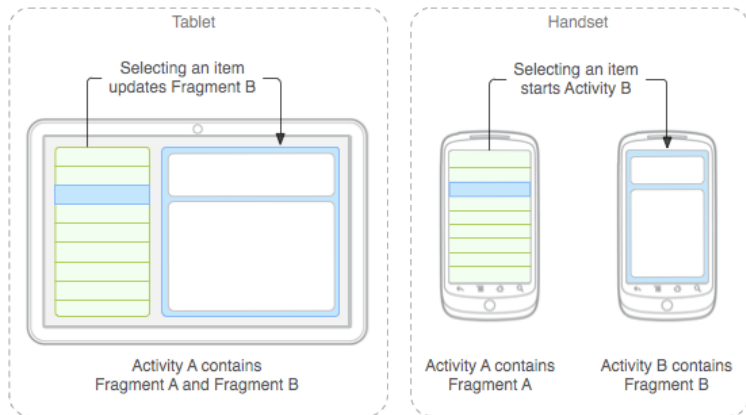
FRAGMENT : UTILITÉ ET UTILISATION

- Un sous ensemble de l'interface
- Plus de fonctionnalité que la View
- Prend une partie du rôle de l'Activity
- Un ou plusieurs Fragments par Vue (MVP)
- Layout différent en fonction du support pour un même code
- Cycle de vie différent de celui de l'Activity

FRAGMENT : CYCLE DE VIE



FRAGMENT : CONCRETEMENT



FRAGMENT : EXEMPLE (1/3)

```
public class MonFragment extends Fragment {  
    public MonFragment() { }  
  
    public static MonFragment newInstance(int  
        param) {  
        MonFragment fragment = new MonFragment();  
        Bundle args = new Bundle();  
        args.putInt("ParamKey", param);  
        fragment.setArguments(args);  
        return fragment;  
    }  
}
```

FRAGMENT : EXEMPLE (2/3)

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}
```

```
@Override
public View onCreateView(LayoutInflater inflater,
    ViewGroup container, Bundle
    savedInstanceState) {
    View view = inflater.inflate(R.layout.notes,
        container, false);
    setParam(getArguments().getInt(
        "ParamKey", 0));
    return view;
}
```


FRAGMENT : EXEMPLE (3/3)

```
public interface MyListener {  
    void onItemClick(Note item);  
}
```

FRAGMENT : DES TRANSACTIONS

```
FragmentManager manager =  
    getSupportFragmentManager();  
FragmentTransaction transaction =  
    manager.beginTransaction();  
Fragment frag = MonFragmentFragment  
    .newInstance(param1);  
//Add  
//transaction.add(R.id.frag_container, frag);  
//Or Replace  
transaction.replace(R.id.frag_container, frag);  
transaction.commit();  
Fragment frag2 =  
manager.findFragmentById(R.id.frag_container);
```

FRAGMENT : ET LE XML ASSOCIÉ

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".MainActivity">
    <FrameLayout
        app:layout_behavior=
"@string/appbar_scrolling_view_behavior"
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
    </LinearLayout>
```

FRAGMENT : OU DU XML

```
<?xml version="1.0" encoding="utf-8"?>
<fr.cpe.fragment.MonFragment
    xmlns:android=
        "http://schemas.android.com/apk/res/android"
    android:id="@+id/my_super_fragment"
    android:layout_height="match_parent"
    android:layout_width="match_parent" />
```

FRAGMENT : INTERACTION DEPUIS LE FRAGMENT

- Utilisation d'un listener

FRAGMENT : INTERACTION DEPUIS LE FRAGMENT

- Utilisation d'un listener
- `MyListener()`;

FRAGMENT : INTERACTION DEPUIS LE FRAGMENT

- Utilisation d'un listener
- `MyListener();`
- `onAttach(Context context)/onDetach();`

FRAGMENT : INTERACTION DEPUIS L'ACTIVITY

- `FragmentManager.findFragmentById();`

FRAGMENT : INTERACTION DEPUIS L'ACTIVITY

- `FragmentManager.findFragmentById();`
- Transactions

FRAGMENT : INTERACTION DEPUIS L'ACTIVITY

- `FragmentManager.findFragmentById();`
- Transactions
- Mutateurs

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR
- 3 DATA BINDING**
- 4 MODEL VIEW VIEW-MODEL
- 5 ANDROIDMANIFEST.XML
- 6 GRADLE
- 7 CONCLUSION

- Permet de faire le lien entre layout et Activity/Fragment

- Permet de faire le lien entre layout et Activity/Fragment
- Évite d'utiliser findViewById

- Permet de faire le lien entre layout et Activity/Fragment
- Évite d'utiliser findViewById
- Natif

- Permet de faire le lien entre layout et Activity/Fragment
- Évite d'utiliser findViewById
- Natif
- Déclaration dans gradle

- Permet de faire le lien entre layout et Activity/Fragment
- Évite d'utiliser findViewById
- Natif
- Déclaration dans gradle
- Déclaration d'une balise <data> dans le layout

- Permet de faire le lien entre layout et Activity/Fragment
- Évite d'utiliser findViewById
- Natif
- Déclaration dans gradle
- Déclaration d'une balise <data> dans le layout
- Création d'une nouvelle classe au build

AU NIVEAU XML

```
<layout>
  <data/>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
      android:text="Super Texte Large"
      android:id="@+id/name_view" />

    <EditText
      android:id="@+id/edit_view"/>
  </LinearLayout>
</layout>
```

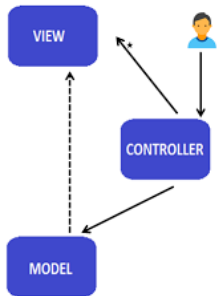
ACTIVITY/FRAGMENT

```
//Dans l'activity  
XmlNameBinding binding = DataBindingUtil.  
    setContentView(this,R.layout.xml_name);  
//Dans le fragment  
XmlNameBinding binding = DataBindingUtil.  
    inflate(inflater,R.layout.xml_name,  
        container,false);  
//dans les deux cas  
binding.nameView.setText("New Text");  
binding.editView.setText("A Text");
```

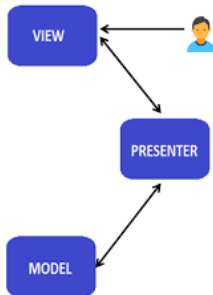
DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR
- 3 DATA BINDING
- 4 MODEL VIEW VIEW-MODEL**
- 5 ANDROIDMANIFEST.XML
- 6 GRADLE
- 7 CONCLUSION

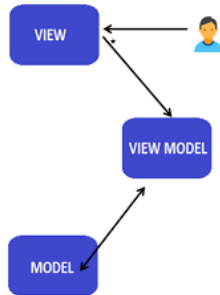
MVC-MVP-MVVM



MVC



MVP



MVVM

MVC-MVP-MVVM

- View -> Layout XML

MVC-MVP-MVVM

- View -> Layout XML
- Model -> Class Java contenant les données

MVC-MVP-MVVM

- View -> Layout XML
- Model -> Class Java contenant les données
- ViewModel -> Class Java héritant de BaseObservable

VIEW

```
<layout
  <data>
    <variable name="my_view_model"
      type="fr.cpe.myapp.MyViewModel">
    </data>
  <LinearLayout>
    <EditText
      android:id="@+id/edit"
      android:text="@{my_view_model.texte}"/>
    <Button
      android:onClick={ () -> my_view_model.save(
        edit.getText().toString()) }"/>
  </LinearLayout>
</layout>
```

MODEL

```
public class MyModel{  
    private String texte;  
    public MyModel(String texte) {  
        this.texte=texte;  
    }  
    public String getTexte() {  
        return texte;  
    }  
    public void setTexte(String texte) {  
        this.texte=texte;  
    }  
}
```

VIEWMODEL

```
public class MyViewModel extends BaseObservable{
    private MyModel myModel;
    public MyViewModel(MyModel myModel) {
        this.myModel = myModel;
    }
    @Bindable
    public String getTexte() {
        return myModel.getTexte();
    }
    public void setTexte(String texte) {
        myModel.setTexte(texte);
        notifyPropertyChanged(BR.texte);
    }
    public void save(String texte) {
        setTexte(texte);
    }
}
```

ACTIVITY/FRAGMENT

```
MyModel myModel = new MyModel("Data");  
MyXMLBinding binding =  
    DataBindingUtil.setContentView(this,  
        R.layout.my_xml);  
binding.setMyViewModel(new  
    MyViewModel(myModel));
```

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR
- 3 DATA BINDING
- 4 MODEL VIEW VIEW-MODEL
- 5 ANDROIDMANIFEST.XML**
- 6 GRADLE
- 7 CONCLUSION

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application
- Contient les permissions

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application
- Contient les permissions
- Contient les activités

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application
- Contient les permissions
- Contient les activités
- Contient les services

ANDROIDMANIFEST.XML

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application
- Contient les permissions
- Contient les activités
- Contient les services
- Contient les intent-filters

- Fichier XML décrivant le projet
- Contient le numéro de version de l'application
- Contient les permissions
- Contient les activités
- Contient les services
- Contient les intent-filters
- Contient souvent les clefs d'utilisation d'APIs externes

CONSTRUCTION 1/5

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
        package="fr.cpe.myapp"
        android:versionCode="5"
        android:versionName="2.1">
    <!-- Déclaration des versions du sdk -->
    <!-- Déclaration des permissions -->
    <!-- Déclaration de l'application -->
</manifest>
```

CONSTRUCTION 2/5

```
<uses-sdk android:minSdkVersion="11"  
          android:targetSdkVersion="23"/>
```

CONSTRUCTION 3/5

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />
```

CONSTRUCTION 4/5

```
<application android:icon="@drawable/icon"  
            android:label="@string/app_name"  
            android:theme="@style/AppTheme">  
    <!-- Déclaration des Activity -->  
</application>
```


CONSTRUCTION 5/5

```
<activity
    android:name="fr.cpe.activity.MainActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category
            android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
<activity android:name=".activity.AboutActivity"/>
```

FICHER FINAL 1/2

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.cpe.myapp"
    android:versionCode="5"
    android:versionName="2.1">
  <!-- Déclaration des versions du sdk -->
  <uses-sdk android:minSdkVersion="11"
    android:targetSdkVersion="23"/>
  <!-- Déclaration des permissions -->
  <uses-permission
    android:name="android.permission.INTERNET" />
  <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"
    />
  <!-- Déclaration de l'application -->
  ...
```

FICHER FINAL 2/2

...

```
<!-- Déclaration de l'application -->
<application
    android:icon="@drawable/icon"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <!-- Déclaration des Activity -->
    <activity
        android:name="fr.cpe.activity.MainActivity"
        android:label="@string/app_name">
        <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category
            android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".activity.AboutActivity"/>
</application>
</manifest>
```

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR
- 3 DATA BINDING
- 4 MODEL VIEW VIEW-MODEL
- 5 ANDROIDMANIFEST.XML
- 6 GRADLE**
- 7 CONCLUSION

- Outil servant aux builds et relativement configurable

- Outil servant aux builds et relativement configurable
- Remplaçant de Ant

- Outil servant aux builds et relativement configurable
- Remplaçant de Ant
- Gestion automatique JCenter/Maven

- Outil servant aux builds et relativement configurable
- Remplaçant de Ant
- Gestion automatique JCenter/Maven
- Basé sur des fichiers de configuration

- Outil servant aux builds et relativement configurable
- Remplaçant de Ant
- Gestion automatique JCenter/Maven
- Basé sur des fichiers de configuration
- Beaucoup d'éléments du manifest ont basculé

- Outil servant aux builds et relativement configurable
- Remplaçant de Ant
- Gestion automatique JCenter/Maven
- Basé sur des fichiers de configuration
- Beaucoup d'éléments du manifest ont basculé
- Déclaration des dépendances

BUILD.GRADLE 1/2

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"

    defaultConfig {
        applicationId "fr.cpe.myapp"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 7
        versionName "2.4"
    }
}
```

BUILD.GRADLE 2/2

```
buildTypes {
    release {
        minifyEnabled false
        proguardFiles
            getDefaultProguardFile('proguard-android.txt'),
            'proguard-rules.pro'
    }
}

dataBinding { enabled = true }

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.android.support:design:25.1.0'
}
```

DÉVELOPPEMENT APPLICATION ANDROID — PLAN

- 1 INTRODUCTION DÉVELOPPEMENT MOBILE
- 2 INTERFACE UTILISATEUR
- 3 DATA BINDING
- 4 MODEL VIEW VIEW-MODEL
- 5 ANDROIDMANIFEST.XML
- 6 GRADLE
- 7 CONCLUSION

SOURCES ET BIBLIO

- <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- **Part de marché Mobile en France Janvier 2015**
- <http://opensignal.com/reports/2015/08/android-fragmentation/>
- <http://developer.android.com/>
- <http://www.statista.com/>
- <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad>

Licence : 