



Université Mohamed Boudiaf de M'Sila  
Faculté des Mathématique et de l'Informatique  
Département d'Informatique

# Les Ressources

---

Samir Akhrouf

Année universitaire  
2019-2020

## Définition

- ❑ Ce sont des fichiers externes ne contenant pas d'instructions qui sont utilisés par le code. Ils contiennent des informations nécessaires au bon déroulement de l'application:
  - des fichiers de paramètres,
  - des données, des images JPEG et PNG,
  - des textes
  - des fichiers XML, ...

## Utilisation

- ❑ Disponibles au moment de l'écriture des codes.
- ❑ Ils restent inchangés tout au long de l'exécution.
- ❑ Dépendent de l'utilisateur et de son appareil
- ❑ Les ressources de l'application sont déposées dans le *répertoire* **res** du projet.
  - Android crée *une classe nommée R* utilisée pour référer aux ressources dans le code.
- ❑ Toutes les ressources sont placées, converties ou non, dans un fichier de type **APK** qui constituera le programme distribuable de l'application.

# Types de Ressources

Type de ressources	Répertoire associé	Description
Valeurs simples	res/values	Définitions en XML de valeurs : chaînes, tableaux, valeurs numériques
Drawables	res/drawable	Des ressources images
Layouts	res/layout	Description en XML des interfaces
Animations	res/anim	Description en XML d'animations
Ressources XML	res/xml	Fichier XML qui peuvent être lus et convertis à l'exécution par la méthode <code>ressources.getXML</code>
Ressources brutes	res/raw	Tous les autres types de ressources : fichiers texte, vidéo, son. Fichiers à ajouter sous leurs formats d'origine.

# Création de ressources

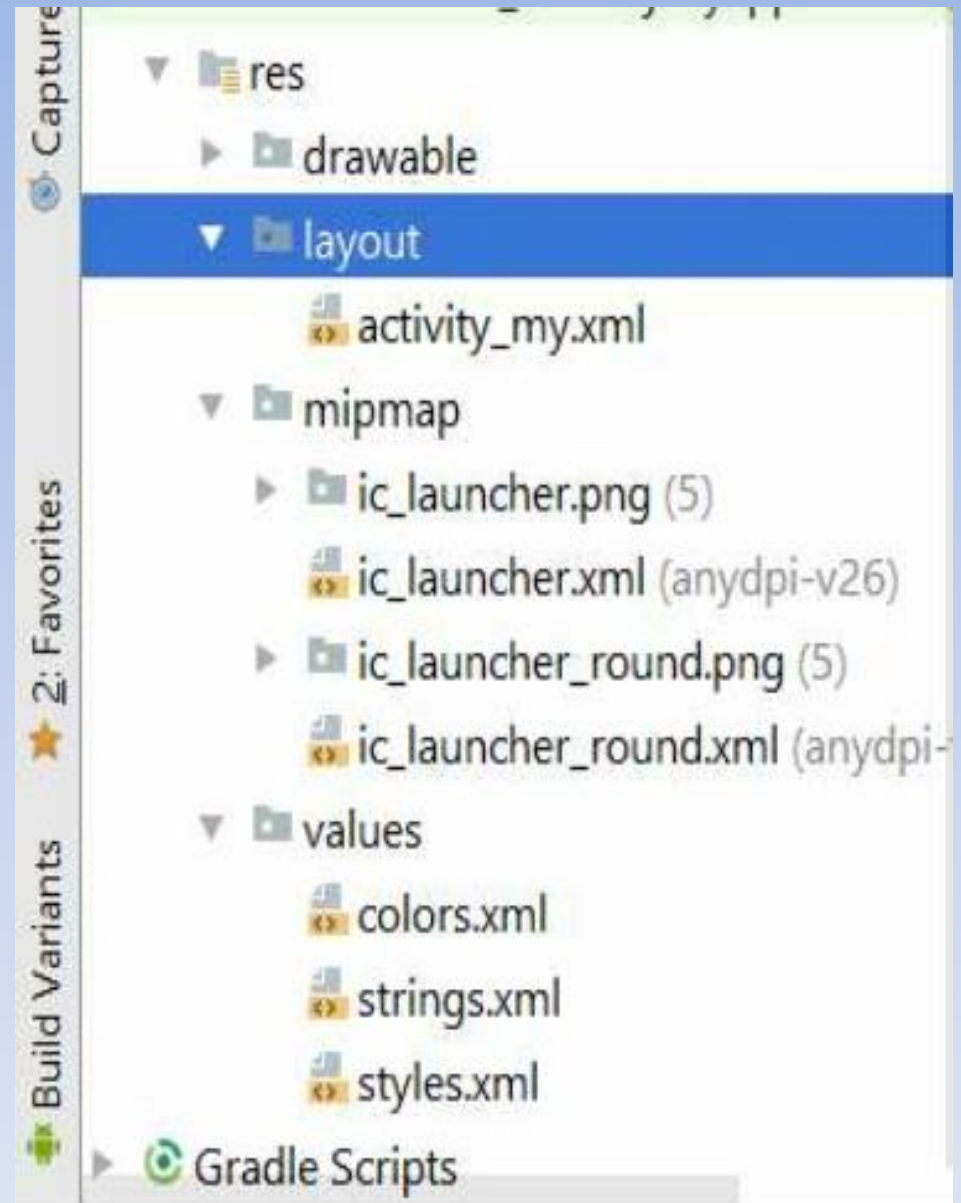
- ❑ Les ressources de type valeur (entiers, booléens, chaînes de caractères, etc. et des tableaux) peuvent être décrites dans des fichiers xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<color name="coulfond" > #AA7B03 </color>
<integer name="limite" > 567 </integer>
<integer-array name="codes_postaux">
    <item>34100</item>
    ...
    <item>30000</item>
</integer-array>
<string name "nom_de_mon_application" > mon premier exemple Android </string>
<string-array name="planetes">
    <item>Mercure</item>

    <item>Venus</item>
</string-array>
<bool name="actif"> true </bool>
<dimen name "taille"> 55px </dimen>
</resources>
```

# Exemples de ressources

- Les icônes de l'application
- Les animations
- Les fichiers de params
- Les strings ...



# Déclaration et utilisation des ressources

- ❑ Les ressources peuvent être prédéfinies par le SDK (répertoire SDK/platforms/android-\*/data/res/
- ❑ Ressources définies par l'utilisateur (dans le répertoire res de son application).
- ❑ Les ressources peuvent être utilisées dans les fichiers XML ou dans le code java
  - ❑ Réutilisation de *la classe statique R* automatiquement générée
- ❑ Utilisation des ressources dans le code Java
  - ❑ Les ressources peuvent être utilisées via leurs identifiants

```
android.R.type_ressource.nom_ressource
```

# Utilisation des ressources

```
public final class R {  
  
    public static final class string {  
        public static final int invitation = 0x7f040001;  
        public static final int texte_titre_ecran = 0x7f040002;  
    };  
  
    public static final class layout {  
        public static final int ecrain_de_demarrage= 0x7f030001;  
        public static final int ecran_principal= 0x7f030002;  
    };  
  
    public static final class drawable {  
        public static final int image_android =          0x7f020000;  
    };  
};
```

Utilisation de la  
ressource dans  
le code Java



android.R.string.invitation

# Ajouter des ressources

☐ Les ressources sont regroupées dans le répertoire res en fonction de leurs types :

The screenshot shows a file explorer view of the 'res' directory in an IDE. The 'layout' folder is selected. Red arrows point from text labels on the right to specific folders and files in the tree:

- Images**: Points to the 'drawable' folder.
- Ecrans (fenêtres)**: Points to the 'activity\_my.xml' file.
- Icones**: Points to the 'ic\_launcher.png' file.
- Strings & values**: Points to the 'values' folder.



## Ajouter des ressources

- ❑ Pour ajouter une ressource, il suffit de la placer dans le répertoire adéquat en terme de fichier direct ou **XML**.
- ❑ A chaque ressource incorporée, est associé un ID qui la référence.
- ❑ Tous les IDs sont définis dans la class « static » R
- ❑ La classe R est générée au fur et à mesure que l'utilisateur manipule l'interface et les ressources.
- ❑ R est générée en utilisant le « aapt » [Android Asset Packaging Tool](#)

## Accéder aux ressources

- ❑ Pour chaque type de ressources, aapt génère une sous classe dans R
  - 🔍Exemple: **R.drawable; R.String**, ...
- ❑ A chaque ressource est associé un id dans la sous classes correspondante.
- ❑ *C'est cet id qu'on utilise pour accéder à la ressource.*
  - 🔍Exemple: **R.drawable.myPhoto**

# Utilisation des ressources

- ❑ Les ressources peuvent être utilisées dans XML ou dans le code java
- ❑ Utilisation des ressources dans le code Java
  - les ressources peuvent être utilisées via leurs identifiants : utilisation de *la classe statique R* automatiquement générée
  - Les ressources peuvent être utilisées en récupérant l'instance de la ressource en utilisant la classe **Resources**

```
Resources res = getResources();  
String hw = res.getString(R.string.hello);  
XXX o = res.getXXX(id);
```

- ❑ Une méthode spécifique pour les objets graphiques permet de les récupérer à partir de leur id:

```
TextView texte = (TextView) findViewById(R.id.le_texte);  
texte.setText("Here we go !");
```

## Accéder aux ressources

- ❑ Dans le code java : Utiliser R avec le id correspondant :  
`myEditText.setText(R.string.salem);`
- ❑ Dans un fichier **XML** : syntaxe XML spéciale en utilisant l'id dans R:  
`@string/salem`

## Exemples : Ressources drawable

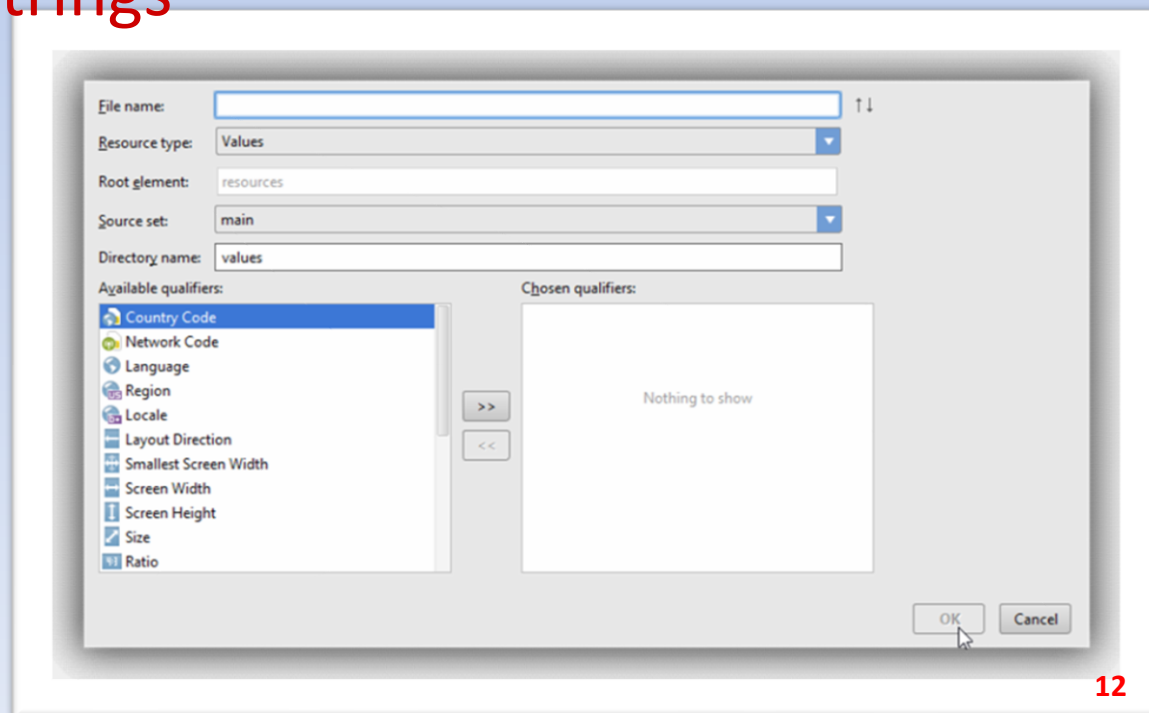
- ❑ Ajout d'une image :
  1. bouton droit sur Res
  2. New image asset
  3. Pour **Asset type**, choisir **Action Bar and Tab Icons**
  4. Choisir le chemin de l'image
  5. Donner un nom à l'image
  6. Finish

# Exemples : Ressources drawable

- ❑ En fonction de la taille de l'écran on a plusieurs cas à gérer :
  - ❑ ldpi : environ 120 dpi ;
  - ❑ mdpi : environ 160 dpi ;
  - ❑ hdpi : environ 240 dpi ;
  - ❑ xhdpi : environ 320 dpi
  - ❑ nodpi : pour ne pas redimensionner les images matricielles

# Exemples : Ressources strings

- ❑ Spécifier la langue:
  - ❑ ar-dz : arabic algeria
  - ❑ en : english
  - ❑ ...



# Exemples : Ressources écran

