



# Android-Java

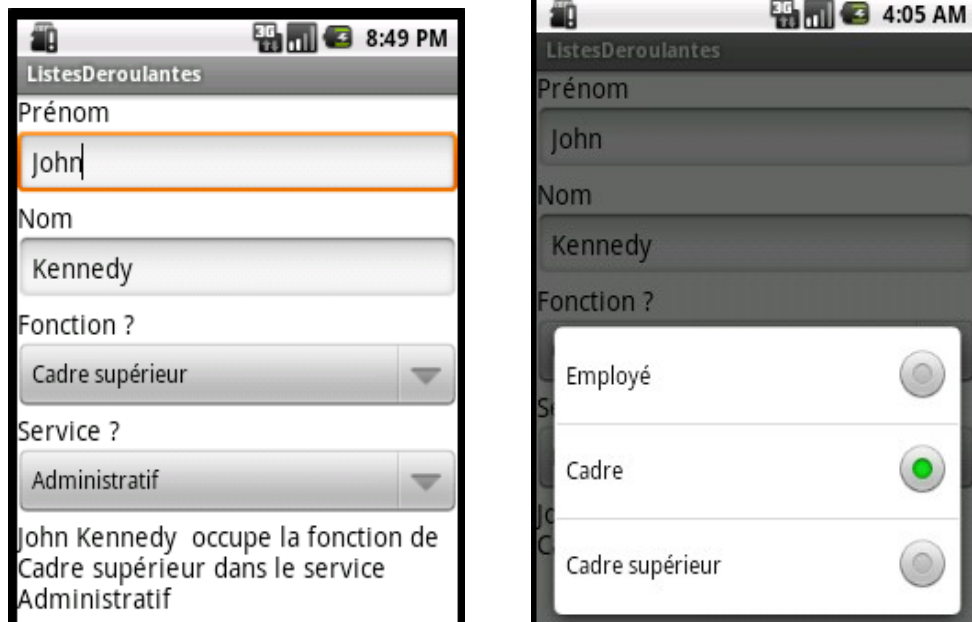
## Les Spinners

# Sommaire

1.1 - Les listes déroulantes : Spinner.....	3
1.1.1 - L'interface (IHM).....	3
1.1.2 - Diagramme de classes.....	5
1.1.3 - Syntaxes côté layout.....	6
1.1.4 - Représentation graphique.....	7
1.1.5 - Syntaxes côté Activity.....	8
1.1.5.1 - Pour un Spinner.....	8
1.1.5.2 - Le remplissage de la liste avec du code java.....	9
1.1.5.3 - La récupération d'un item sélectionné.....	10
1.1.5.4 - La récupération d'un item sélectionné hors d'une gestion événementielle onItemSelected.....	11
1.1.6 - Codes.....	12
1.1.7 - Exercice : une nouvelle liste déroulante.....	16
1.1.8 - Mettre à jour un spinner.....	17
1.1.8.1 - Toute ou une grande partie de la liste est modifiée.....	17
1.1.8.2 - Un seul item est ajouté.....	17
1.1.8.3 - Supprimer un item.....	17
1.1.9 - Pré-sélectionner un élément d'un spinner.....	18
1.2 - Spinner personnalisé.....	19
1.3 - Spinner avec des images.....	20
1.3.1 - Objectif.....	20
1.3.2 - IHM.....	20
1.3.3 - Démarche.....	21
1.3.4 - La classe TexteEtImage.java.....	22
1.3.5 - Les layouts.....	23
1.3.5.1 - Layout principal.....	23
1.3.5.2 - ligne_spinner_texte.xml.....	25
1.3.5.3 - ligne_spinner_image.xml.....	26
1.3.5.4 - ligne_spinner_image_et_texte.xml.....	27
1.3.6 - Les adapters.....	28
1.3.6.1 - ArrayAdapterTexte.java.....	28
1.3.6.2 - ArrayAdapterImage.java.....	29
1.3.6.3 - ArrayAdapterImageEtTexte.java.....	31
1.3.7 - L'activité.....	33
1.4 - Annexes.....	36
1.4.1 - Tableau des jours de la semaine.....	36
1.4.2 - Tableau des mois de l'année.....	36

## 1.1 - LES LISTES DÉROULANTES : SPINNER

### 1.1.1 - L'interface (IHM)



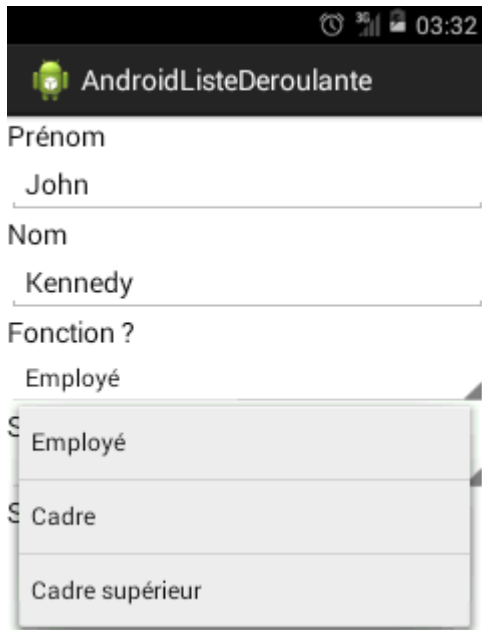
L'interface visuelle est constituée de 2 zones de saisie (EditText), de 2 listes déroulantes (Spinner) et de 5 étiquettes (TextView).

La liste déroulante (le Spinner) peut être remplie :

- ✓ soit dans le layout à partir d'un `<string-array>`, stocké dans les ressources,
- ✓ soit dynamiquement dans le code de l'activité à partir d'un tableau de String ou une `List<String>` (le tableau ou la List étant rempli à partir de données SQL ou XML ou JSON, etc) via un `ArrayAdapter`,

Dans cet exemple la liste des fonctions est remplie via un tableau de strings stocké dans les ressources et la liste des services est remplie via un tableau statique de strings dans du code java.

Ou ça



AndroidListeDeroulante

Prénom  
John

Nom  
Kennedy

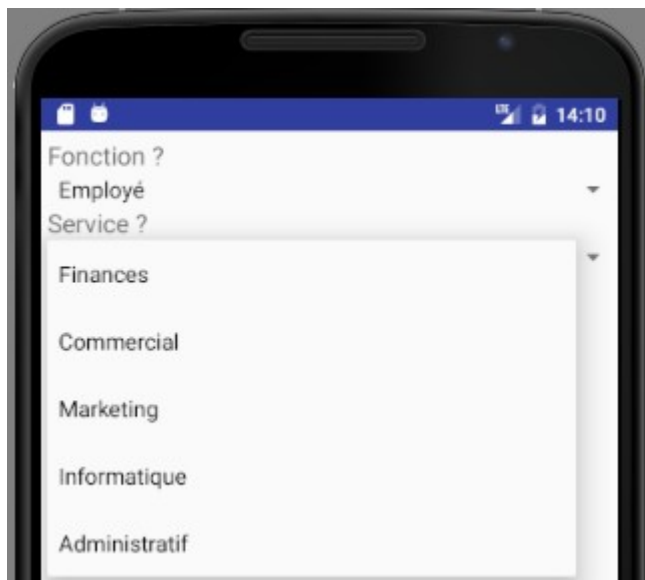
Fonction ?  
Employé

S  
Employé

S  
Cadre

Cadre supérieur

Le code présenté plus bas simplifie cette interface.



Fonction ?  
Employé

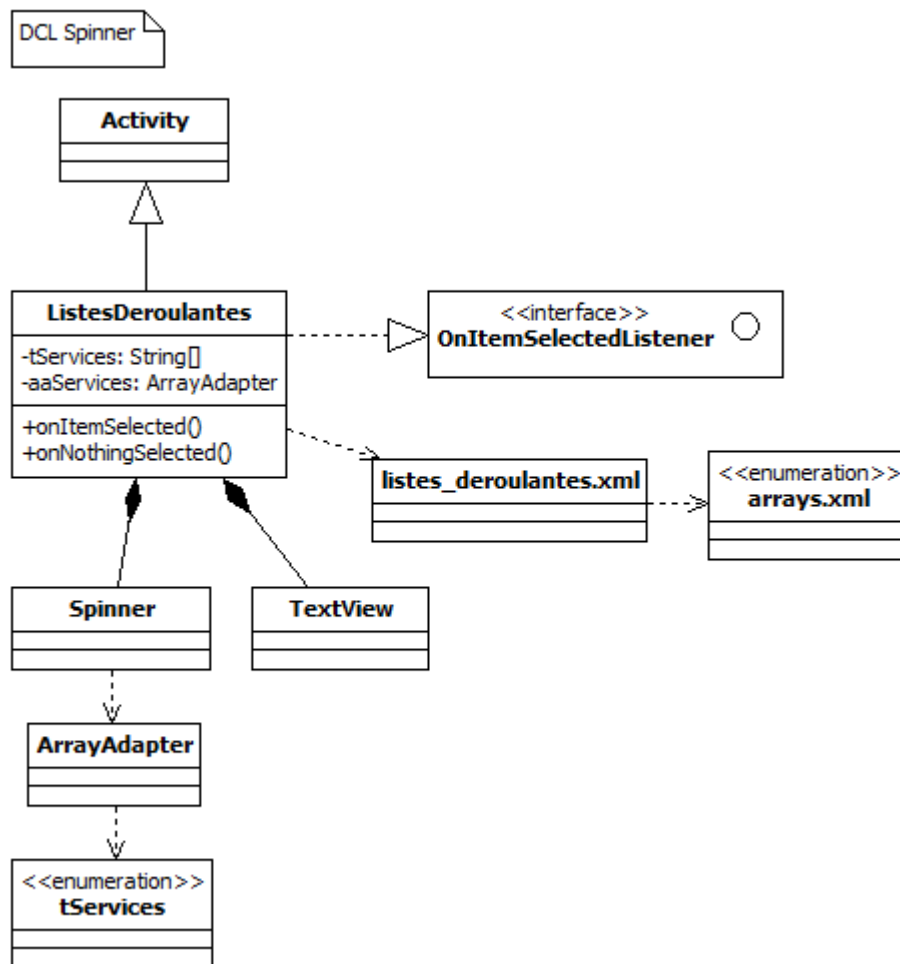
Service ?  
Finances  
Commercial  
Marketing  
Informatique  
Administratif

Affichage suite à un appui.



Cadre supérieur

### 1.1.2 - Diagramme de classes



**Note :**

Le tableau `tServices` et l'`ArrayAdapter aaServices` peuvent ne pas être des attributs de classe et être des variables déclarées dans la méthode `initInterface()`.

### 1.1.3 - Syntaxes côté layout

**Spinner** : liste déroulante



Spinner  
Sub Item



```
<Spinner
    android:id="@+id/spinnerFonctions"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

**Note** : pour initialiser une liste déroulante dans le layout créez un tableau de strings dans un fichier nommé /res/values/arrays.xml.

Pour créer le tableau allez dans /res/values/, cliquez droit, sélectionnez New puis Values Resource File ... nommez-le arrays.xml.

```
<string-array name="nom_du_tableau">
    <item>Item 1</item>
    <item>Item 2</item>
</string-array>
```

Ajoutez cet attribut au Spinner

```
android:entries="@array/nom_du_tableau"
```

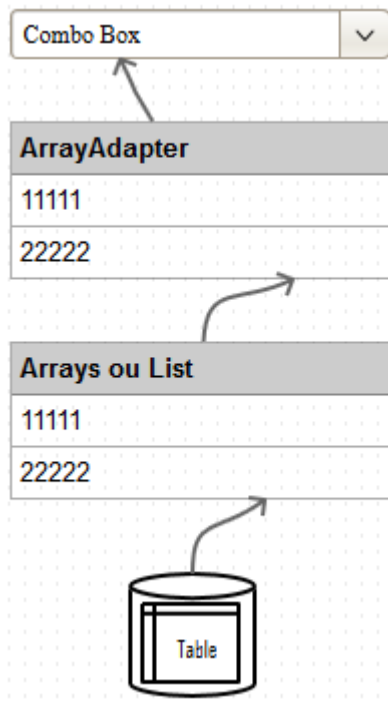
Par exemple :

```
<string-array name="fonctions">
    <item>Employé</item>
    <item>Cadre</item>
    <item>Cadre supérieur</item>
</string-array>
```

et

```
android:entries="@array/fonctions"
```

### 1.1.4 - Représentation graphique



### 1.1.5 - Syntaxes côté Activity

#### 1.1.5.1 - Pour un Spinner

La classe Activité **doit** implémenter l'interface **AdapterView.OnItemSelectedListener** pour gérer la gestion de la sélection d'un item par l'utilisateur :

```
public class Classe extends Activity implements OnItemSelectedListener
```

et donc importer l'interface **AdapterView.OnItemSelectedListener** :

```
import android.widget.AdapterView.OnItemSelectedListener;
```

et faire la liaison entre le widget et le listener :

```
spinnerXXX.setOnItemSelectedListener(this);
```



### 1.1.5.2 - Le remplissage de la liste avec du code java

Le Spinner est rempli à partir d'un ArrayAdapter.

Un ArrayAdapter est un pont entre un **Array** ou une **List** et une interface visuelle.

Cet ArrayAdapter est rempli à partir d'un tableau d'objets ou de String ou une List d'objets ou de Strings.

<http://developer.android.com/reference/android/widget/ArrayAdapter.html>

Déclaration d'un ArrayAdapter (cela peut être une variable locale)

```
private ArrayAdapter<String> aa;
```

Constructeur de l'ArrayAdapter

```
ArrayAdapter(contexte, identifiant de ressource de la vue, tableau)
```

L'identifiant de la ressource de la vue est **android.R.layout.simple\_spinner\_item** pour un Spinner. C'est un identifiant système (android.R.layout.id).

Cf <http://developer.android.com/reference/android/R.layout.html>

De plus vous devez préciser le type de liste pour lequel il est implémenté.

```
ArrayAdapter.setDropDownViewResource(id);
```

L'id est **android.R.layout.simple\_spinner\_dropdown\_item**.

Ou android.R.layout.simple\_list\_item\_checked (pour avoir une case à cocher à droite).

Ou android.R.layout.simple\_list\_item\_single\_choice (pour avoir un bouton radio à droite).

Remplissage du Spinner

```
Spinner.setAdapter(ArrayAdapter);
```

Ajout d'un gestionnaire d'événement pour la sélection

```
Spinner.setOnItemSelectedListener(this);
```

Pour personnaliser un Spinner (Texte + image) :

<http://abhiandroid.com/ui/custom-spinner-examples.html>

### 1.1.5.3 - La récupération d'un item sélectionné

La classe **doit** implémenter les deux méthodes abstraites suivantes :

Lorsque qu'un item est sélectionné :

```
onItemSelected(AdapterView<?> parent, View v, int position, long id)
```

parent	The AdapterView where the selection happened – Ici le spinner
view	The view within the AdapterView that was clicked – Ici le TextView sous-jacent
position	The position of the view in the adapter – La position
id	The row id of the item that is selected - La position

Lorsque qu'aucun item n'est sélectionné :

```
onNothingSelected(AdapterView<?> arg0)
```

Si le Spinner a été rempli avec un tableau de strings la récupération du texte de l'item sélectionné pourrait se faire ainsi :

```
@Override
public void onItemSelected(AdapterView<?> parent, View vue, int position, long id) {
    String lsSelection = getResources().getStringArray(R.array.tableau)[position];
}
```

ou **mieux**, c'est indépendant du nom de la source à partir du paramètre AdapterView

```
String lsSelection = parent.getItemAtPosition(position).toString();
```

ou

```
String lsSelection = parent.getAdapter().getItem(position).toString();
```

ou

```
String lsSelection = parent.getSelectedItem().toString();
```

ou à partir du paramètre View (TextView sous-jacent)

```
String lsSelection = ((TextView)vue).getText().toString();
```

#### 1.1.5.4 - La récupération d'un item sélectionné **hors** d'une gestion événementielle onItemSelectedListener

```
Spinner.getSelectedItem().toString()
```

#### Autres méthodes

Méthode	Description
View getView()	Renvoie un item de la liste déroulante
int getCount()	Renvoie le nombre d'items de la liste déroulante
void setSelection(int position)	Sélectionne un item de la liste déroulante

### 1.1.6 - Codes

Créez une nouvelle application Android nommée `ListesDeroulantes` avec une activité nommée `ListesDeroulantes` et un layout nommé `listes_deroulantes`.

Créez un fichier nommé `arrays.xml` dans le dossier `/res/values/` et ajoutez ce qui suit dans l'élément `<resources>`.

Pour créer le tableau allez dans `/res/values/`, cliquez droit, sélectionnez New et Values Resource File ... nommez-le `arrays.xml`.

```
<string-array name="fonctions">
  <item>Employé</item>
  <item>Cadre</item>
  <item>Cadre supérieur</item>
</string-array>
```

**Le layout : listes\_deroulantes.xml**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="5dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fonction ?"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/spinnerFonctions"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:entries="@array/fonctions"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Service ?"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <Spinner
        android:id="@+id/spinnerServices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/textViewMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Message" />

</LinearLayout>
```

**L'activité : ListesDeroulantes.java**

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

// -----
public class ListesDeroulantes extends Activity implements OnItemSelectedListener
{

    private Spinner spinnerFonctions;
    private Spinner spinnerServices;
    private TextView textViewMessage;

    @Override
    // -----
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.listes_deroulantes);

        initInterface();

    } // / onCreate

    // -----
    private void initInterface() {

        // --- Pour une liste deroulante a initialiser en code
        String[] tServices = { "Finances", "Commercial", "Marketing",
"Informatique", "Administratif" };
        ArrayAdapter<String> aaServices;

        /*
         * INITIALISATION DES COMPOSANTS
         */
        textViewMessage = findViewById(R.id.textViewMessage);
        spinnerFonctions = findViewById(R.id.spinnerFonctions);
        spinnerServices = findViewById(R.id.spinnerServices);

        /*
         * REMPLISSAGE DE LA LISTE DES SERVICES
         */
        // --- Le spinner avec les services
        aaServices = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, tServices);

        aaServices.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
);

        spinnerServices.setAdapter(aaServices);
    }
}
```

```
        /*
        * GESTION EVENEMENTIELLE
        */
        spinnerFonctions.setOnItemSelectedListener(this);
        spinnerServices.setOnItemSelectedListener(this);

    } // / initInterface

    @Override
    // --- Spinner, TextView, position, position
    public void onItemSelected(AdapterView<?> parent, View vue, int position, long
id) {
        /*
        * Le spinner spinnerFonctions est initialisé à partir d'un tableau
        * dans arrays.xml nomme fonctions
        */
        if (parent == spinnerFonctions) {
            String lsFonction = parent.getItemAtPosition(position).toString();
            Toast.makeText(this, lsFonction, Toast.LENGTH_SHORT).show();
        }

        if (parent == spinnerServices) {
            String lsService = parent.getSelectedItem().toString();
            Toast.makeText(this, lsService, Toast.LENGTH_SHORT).show();
        }

    } // / onItemSelected

    @Override
    // -----
    public void onNothingSelected(AdapterView<?> arg0) {
    } // / onNothingSelected

} // / class ListesDeroulantes
```

### 1.1.7 - Exercice : une nouvelle liste déroulante

Ajoutez une liste déroulante dans le layout acteur\_xxx.xml pour la nationalité de l'acteur.  
Modifiez l'activité ActeurXXX.java en conséquence.

Vous avez 2 possibilités pour le remplissage de la liste :

- ✓ la plus facile en passant par un string-array,
- ✓ la plus difficile en passant par un ArrayAdapter.

Choisissez la plus difficile !

Vous avez 2 possibilités pour l'affichage dans le TextView :

- ✓ soit après la sélection dans la liste,
- ✓ soit sur l'appui sur le bouton.

Choisissez !

Prénom : Julia

Nom : fr

☐ Vivant(e)

☐ Homme

☐ Femme

Naissance : 1990

Nationalité : française

Valider

Julia est de nationalité française

4:16 AM

**Note :**

étant donnée la petite taille de l'écran, il faut peut-être changer la taille de la police (android:textSize) ou bien avoir vu le ScrollView ou avoir vu les layouts horizontaux dans des layouts verticaux et les poids (android:layout\_weight).



### 1.1.8 - Mettre à jour un spinner

#### 1.1.8.1 - Toute ou une grande partie de la liste est modifiée

En ré-instanciant l'adapter et en le réaffectant au Spinner.

```
private void remplirSpinner(List<String> listeElements) {  
    aaSpinner = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,  
    listeElements);  
  
    aaSpinner.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)  
    ;  
    spinner1.setAdapter(aaSpinner);  
} // / remplirSpinner
```

#### 1.1.8.2 - Un seul item est ajouté

```
this.arrayAdapter.add("valeur");
```

**Note** : il faut que l'ArrayAdapter ait été alimenté via une List et pas un tableau statique.

#### 1.1.8.3 - Supprimer un item

```
this.arrayAdapter.remove("valeur");
```

**Note** : il faut que l'ArrayAdapter ait été alimenté via une List et pas un tableau statique.

### 1.1.9 - Pré-sélectionner un élément d'un spinner

Boucler sur l'Adapter et sélectionner dans le Spinner.

```
String lsJour = "Mercredi";
String lsItem;
SpinnerAdapter sa;
int liCount;

sa = spinnerJours.getAdapter();
liCount = sa.getCount();
for (int i = 0; i < liCount; i++) {
    lsItem = sa.getItem(i).toString();
    if (lsItem.equals(lsJour)) {
        spinnerJours.setSelection(i);
    }
}
```

## 1.2 - SPINNER PERSONNALISÉ

Si vous voulez personnaliser un spinner (fonte, couleur, ...) vous devez créer un layout pour les items du Spinner.



### Code du layout de ligne du spinner

/res/layout/spinner\_ligne\_perso.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FF0000"
    android:paddingBottom="10dp"
    android:paddingLeft="3dp"
    android:paddingTop="10dp"
    android:textColor="#FFFFFF"
    android:textSize="20sp" />
```

### Code de l'activité

```
String[] tNationalites = {"Française", "Italienne", "Espagnole", "Marocaine",
    "Tunisienne"};
ArrayAdapter<String> aaNationalites = new ArrayAdapter<String>( this,
R.layout.spinner_ligne_perso, tNationalites);
aaNationalites.setDropDownViewResource(R.layout.spinner_ligne_perso);
```

## 1.3 - SPINNER AVEC DES IMAGES

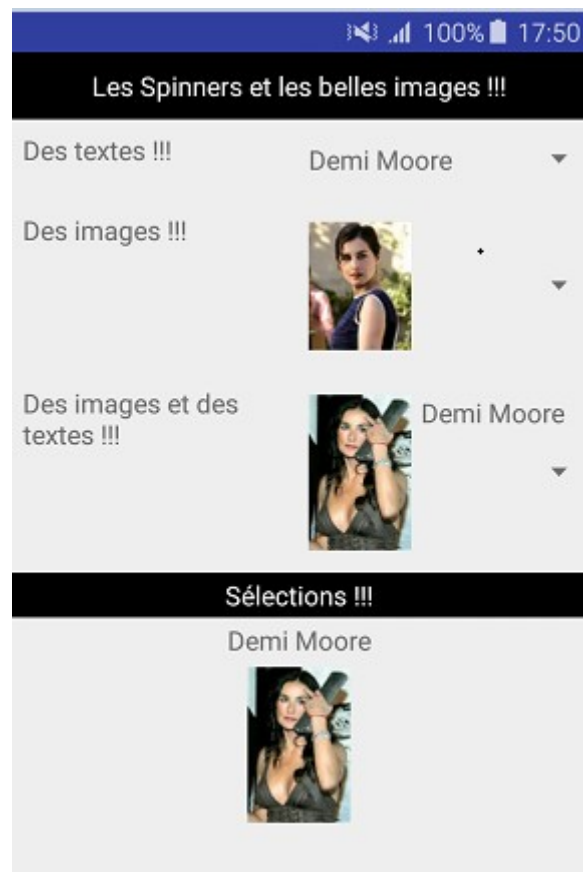
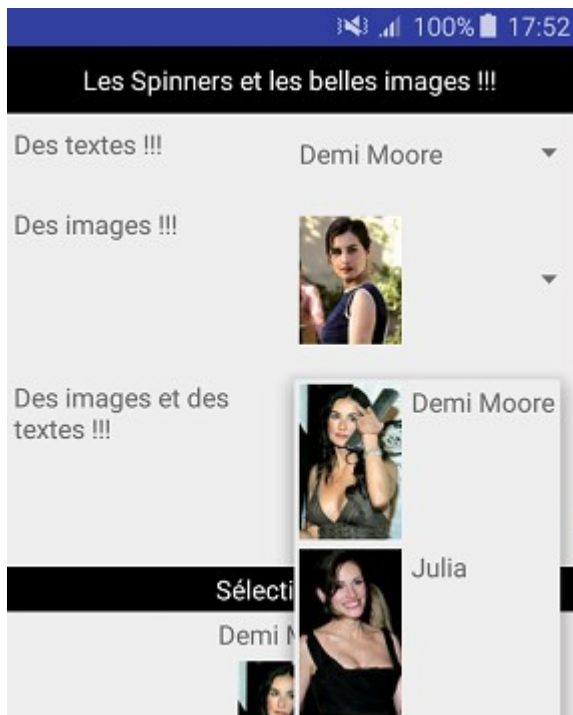
### 1.3.1 - Objectif

Afficher des listes déroulantes (spinner) avec des images ou des textes et des images.

### 1.3.2 - IHM

Il y a 3 spinners :

- ✓ un avec du texte,
- ✓ un avec des images,
- ✓ un avec du texte et une image.



### 1.3.3 - Démarche

Les images sont stockées dans les ressources !!!

Création d'une classe POJO, nommée TexteEtImage, avec un attribut de type String pour les textes et un attribut de type int pour les références vers les images.

Création d'un layout principal avec des TextView et les 3 Spinners.

Création de 3 layouts de ligne ;

- un avec un TextView,
- un avec une ImageView,
- un avec un TextView et une ImageView.

Création de 3 classes pour les Adapters :

- une héritant de ArrayAdapter<String> ,
- une héritant de BaseAdapter ,
- une héritant de ArrayAdapter<TexteEtImage> .

Ces classes implémentent différentes méthodes en fonction de la classe parent. Cf leur diagramme de classes.

Création d'une activité :

Les ArrayAdapter sont instanciés avec des listes et les différents layouts de ligne.

### 1.3.4 - La classe TexteEtImage.java

Une classe « POJO ».

```
package pb.fr.spinneravecimagesetttextes;

public class TexteEtImage {

    /*
    ATTRIBUTS
    */
    private String texte;
    private int image;

    /*
    CONSTRUCTEURS
    */
    public TexteEtImage() {
    } /// constructeur

    public TexteEtImage(String texte, int image) {
        this.texte = texte;
        this.image = image;
    } /// constructeur

    /*
    GETTERS ET SETTERS
    */
    public String getTexte() {
        return texte;
    }

    public void setTexte(String texte) {
        this.texte = texte;
    }

    public int getImage() {
        return image;
    }

    public void setImage(int image) {
        this.image = image;
    }
} /// class
```

### 1.3.5 - Les layouts

#### 1.3.5.1 - Layout principal

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#000"
        android:gravity="center"
        android:padding="10dp"
        android:text="Les Spinners et les belles images !!!"
        android:textColor="#FFF" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="5dp">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="50"
            android:padding="3dp"
            android:text="Des textes !!!" />

        <Spinner
            android:id="@+id/spinnerTextes"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="3dp"
            android:layout_weight="50"
            android:padding="3dp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="5dp">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="50"
            android:padding="3dp"
            android:text="Des images !!!" />

        <Spinner
```

```
        android:id="@+id/spinnerImages"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="3dp"
        android:layout_weight="50"
        android:padding="3dp" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="50"
        android:padding="3dp"
        android:text="Des images et des textes !!!" />

    <Spinner
        android:id="@+id/spinnerImagesEtTextes"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="3dp"
        android:layout_weight="50"
        android:padding="3dp" />
</LinearLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#000"
    android:gravity="center"
    android:padding="3dp"
    android:text="Sélections !!!"
    android:textColor="#FFF" />

<TextView
    android:id="@+id/textViewDescriptionLogo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="3dp"
    android:gravity="center"
    android:text="Texte" />

<ImageView
    android:id="@+id/imageViewLogo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="3dp"
    android:src="@drawable/spinner_image" />
</LinearLayout>
```



## 1.3.5.2 - ligne\_spinner\_texte.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textViewSpinnerTextes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="3dp"
        android:text="Texte" />
</LinearLayout>
```

## 1.3.5.3 - ligne\_spinner\_image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/imageViewSpinnerImages"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="3dp"
        android:src="@drawable/spinner_image" />
</LinearLayout>
```

## 1.3.5.4 - ligne\_spinner\_image\_et\_texte.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/imageViewSpinnerImagesEtTextes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="3dp"
        android:src="@drawable/spinner_image" />
    <TextView
        android:id="@+id/textViewSpinnerImagesEtTextes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="3dp"
        android:text="Texte" />
</LinearLayout>
```

### 1.3.6 - Les adapters

#### 1.3.6.1 - ArrayAdapterTexte.java

```
package pb.fr.spinneravecimagesettexes;

import android.content.Context;
import android.view.LayoutInflater;
import android.widget.ArrayAdapter;

import java.util.List;

public class ArrayAdapterTexte extends ArrayAdapter<String> {
    private LayoutInflater inflater;
    private Context contexte;

    public ArrayAdapterTexte(Context contexte, int ressource, int textViewResourceId,
List<String> liste) {
        // activité, layout de ligne, widget, liste de "valeurs"
        super(contexte, ressource, textViewResourceId, liste);
        /***** Layout inflater to call external xml layout () *****/
        inflater = (LayoutInflater)
contexte.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    } /// constructeur
} /// class ArrayAdapterTexte
```

## 1.3.6.2 - ArrayAdapterImage.java

```
package pb.fr.spinneravecimagesettexes;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.ImageView;

import java.util.ArrayList;
import java.util.List;

public class ArrayAdapterImage extends BaseAdapter {
    private LayoutInflater inflater;
    private Context contexte;
    private List<Integer> listImages;

    public ArrayAdapterImage() {
        listImages = null;
    }

    public ArrayAdapterImage(Context contexte, List<Integer> listImages) {
        this.listImages = listImages;
        inflater = (LayoutInflater)
contexte.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    public ArrayAdapterImage(Context contexte, int ressource, int textViewResourceId,
List<Integer> liste) {
        /***** Layout inflater to call external xml layout () *****/
        inflater = (LayoutInflater)
contexte.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    } /// constructeur

    @Override
    public int getCount() {
        return listImages.size();
    }

    @Override
    public Object getItem(int i) {
        Object objet = null;
        return objet;
    }

    @Override
    public long getItemId(int i) {
        return i;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
```

```
        View row = inflater.inflate(R.layout.ligne_spinner_image, viewGroup, false);
        ImageView imageView = (ImageView)
row.findViewById(R.id.imageViewSpinnerImages);
        imageView.setImageResource(listeImages.get(i));
        return row;
    }

    private static class ViewHolder {
        ImageView imageView;
    }
} // class ArrayAdapterImage
```

## 1.3.6.3 - ArrayAdapterImageEtTexte.java

```
package pb.fr.spinneravecimagesettexes;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

public class ArrayAdapterImageEtTexte extends ArrayAdapter<TexteEtImage> {

    private LayoutInflater inflater;
    private Context contexte;
    private List<String> tableau;
    private List<TexteEtImage> listeObjets;

    public ArrayAdapterImageEtTexte(Context contexte, int ressource, int
    textViewResourceId, List<TexteEtImage> listeObjets) {

        // activité, layout de ligne, widget, liste de "valeurs"
        super(contexte, ressource, textViewResourceId, listeObjets);
        this.listeObjets = listeObjets;
        /***** Layout inflater to call external xml layout () *****/
        inflater = (LayoutInflater)
        contexte.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    } /// constructeur

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        return getCustomView(position, convertView, parent);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        return getCustomView(position, convertView, parent);
    }

    // This function called for each row ( Called data.size() times )
    public View getCustomView(int position, View convertView, ViewGroup parent) {

        /***** Inflate spinner_rows.xml file for each row ( Defined below ) *****/
        View row = inflater.inflate(R.layout.ligne_spinner_image_et_texte, parent, false);

        ImageView imageView = (ImageView)
        row.findViewById(R.id.imageViewSpinnerImagesEtTextes);
        imageView.setImageResource(listeObjets.get(position).getImage());

        TextView textView = (TextView)
        row.findViewById(R.id.textViewSpinnerImagesEtTextes);
        textView.setText(listeObjets.get(position).getTexte());
    }
}
```

```
        return row;
    } /// getView
} /// class ArrayAdapterImageEtTexte
```



### 1.3.7 - L'activité

```
package pb.fr.spinneravecimagesettextes;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

public class ListeDeroulanteImagesEtTextes extends Activity implements
    AdapterView.OnItemClickListener {

    private Spinner spinnerTextes;
    private Spinner spinnerImages;
    private Spinner spinnerImagesEtTextes;

    private TextView textViewDescriptionLogo;
    private ImageView imageViewLogo;

    List<Integer> listeImages;
    List<TexteEtImage> listeTextesImages;

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

        spinnerTextes = findViewById(R.id.spinnerTextes);
        spinnerImages = findViewById(R.id.spinnerImages);
        spinnerImagesEtTextes = findViewById(R.id.spinnerImagesEtTextes);

        spinnerTextes.setOnItemClickListener(this);
        spinnerImages.setOnItemClickListener(this);
        spinnerImagesEtTextes.setOnItemClickListener(this);

        textViewDescriptionLogo = findViewById(R.id.textViewDescriptionLogo);
        imageViewLogo = findViewById(R.id.imageViewLogo);

        /*
        TEXTES
        */
        List<String> listeTextes = new ArrayList();
        listeTextes.add("Demi Moore");
        listeTextes.add("Julia");
        listeTextes.add("Pamela");
        listeTextes.add("Rene Russo");
        listeTextes.add("Amira Casar");

        // activité, layout de ligne, widget, liste de "valeurs"
```

```
        ArrayAdapterTexte aaTextes = new ArrayAdapterTexte(this,
R.layout.ligne_spinner_texte, R.id.textViewSpinnerTextes, listeTextes);
        spinnerTextes.setAdapter(aaTextes);

        /*
        IMAGES
        */
        listeImages = new ArrayList();
        listeImages.add(R.drawable.amira_casar_ico);
        listeImages.add(R.drawable.demi_moore_ico);
        listeImages.add(R.drawable.julia_roberts_ico);
        listeImages.add(R.drawable.logo_google_ico);
        listeImages.add(R.drawable.rene_russo_ico);

        // activité, layout de ligne, widget, liste de "valeurs"
        ArrayAdapterImage aalimages = new ArrayAdapterImage(this, listeImages);
        spinnerImages.setAdapter(aalimages);

        /*
        IMAGES ET TEXTES
        */
        listeTextesImages = new ArrayList();
        TexteEtImage ti;
        ti = new TexteEtImage("Demi Moore", R.drawable.demi_moore_ico);
        listeTextesImages.add(ti);
        ti = new TexteEtImage("Julia", R.drawable.julia_roberts_ico);
        listeTextesImages.add(ti);
        ti = new TexteEtImage("Pamela", R.drawable.logo_google_ico);
        listeTextesImages.add(ti);
        ti = new TexteEtImage("Rene Russo", R.drawable.rene_russo_ico);
        listeTextesImages.add(ti);
        ti = new TexteEtImage("Amira Casar", R.drawable.amira_casar_ico);
        listeTextesImages.add(ti);

        // activité, layout de ligne, widget, liste de "valeurs"
        ArrayAdapterImageEtTexte aalimagesEtTextes = new
ArrayAdapterImageEtTexte(this, R.layout.ligne_spinner_image_et_texte,
R.id.textViewSpinnerImagesEtTextes, listeTextesImages);
        spinnerImagesEtTextes.setAdapter(aalimagesEtTextes);

    } /// onCreate

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
    {

        textViewDescriptionLogo.setText("");
        imageViewLogo.setImageDrawable(null);

        if (parent == spinnerTextes) {
            textViewDescriptionLogo.setText(parent.getItemAtPosition(position).toString());
        }

        if (parent == spinnerImages) {
            // OK mais pas très bon, nécessite de déclarer le tableau d'images comme
            variable d'instance
            imageViewLogo.setImageResource(listeImages.get(position));
        }
        if (parent == spinnerImagesEtTextes) {
```

```
        imageViewLogo.setImageResource(listeTextesImages.get(position).getImage());
        textViewDescriptionLogo.setText(listeTextesImages.get(position).getTexte());
    }
} /// onItemSelected

@Override
public void onNothingSelected(AdapterView<?> adapterView) {
    } /// onNothingSelected
} /// class
```

## 1.4 - ANNEXES

### 1.4.1 - Tableau des jours de la semaine

```
String[] tJours = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"};
```

### 1.4.2 - Tableau des mois de l'année

```
String[] tMois =  
{ "Janvier", "Février", "Mars", "Avril", "Mai", "Juin", "Juillet", "Août", "Septembre", "Octobre", "Novembre", "Décembre"};
```