

ListView : afficher une liste d'éléments

(voir tuto : <http://tutos-android-france.com/listview-afficher-une-liste-delements/>)

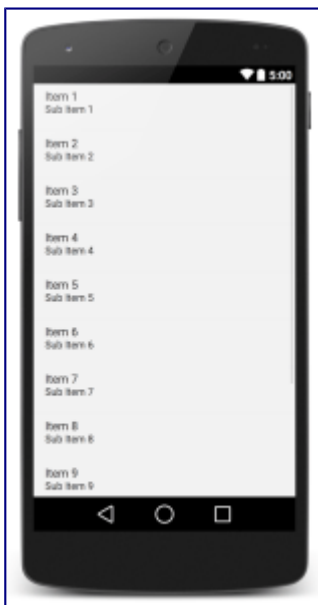
Android vous propose dans son SDK une suite d'objets permettant l'affichage de liste ou de grilles d'éléments de façon simple et performant, tout en gardant le contrôle total sur l'apparence des vues que l'on souhaite afficher.

C'est pourquoi dans la liste des vues disponible apparait l'objet **ListView**, Créez un nouveau projet, puis ajouter une ListView dans votre fichier **res/layout/activity_main.xml** :

Remplacer le fichier créé par défaut par le suivant :

```
01<RelativeLayout
02xmlns:android="http://schemas.android.com/apk/res/android"
03    android:layout_width="match_parent"
04    android:layout_height="match_parent">
05
06    <ListView
07        android:id="@+id/listView"
08        android:layout_width="wrap_content"
09        android:layout_height="wrap_content" />
10</RelativeLayout>
```

Votre preview devrait vous afficher l'écran suivant :



Passons maintenant au code, nous allons commencer par afficher une liste de pseudos afin de comprendre les concepts de base de cette ListView.

Ouvrez donc le fichier **MainActivity.java** et récupérez un pointeur vers la ListView défini dans activity_main.xml

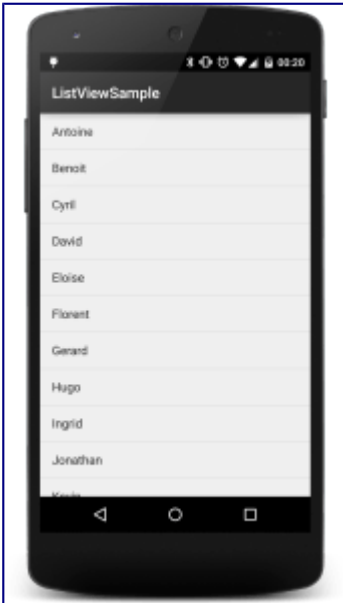
```
01 public class MainActivity extends ActionBarActivity {
02
03     ListView mListView;
04
05     @Override
06     protected void onCreate(Bundle savedInstanceState) {
07         super.onCreate(savedInstanceState);
08         setContentView(R.layout.activity_main);
09
10         mListView = (ListView) findViewById(R.id.listView);
11     }
12 }
```

Essayons d'afficher simplement la liste suivante : « Antoine », « Benoit », « Cyril », « David », « Eloise », « Florent », « Gerard », « Hugo », « Ingrid », « Jonathan », « Kevin », « Logan », « Mathieu », « Noemie », « Olivia », « Philippe », « Quentin », « Romain », « Sophie », « Tristan », « Ulric », « Vincent », « Willy », « Xavier », « Yann », « Zoé »

Afin de remplir une ListView, il existe un objet nommé **Adapter**, qui va prendre en entrée une liste d'objets et un layout XML et va générer pour chaque entrée une cellule formatée. Afin de mieux comprendre, mettons nous dans le code, créez un objet ArrayAdapter comme ceci, et affecter le à votre ListView :

```
01 public class MainActivity extends ActionBarActivity {
02
03     ListView mListView;
04     String[] prenom = new String[]{
05         "Antoine", "Benoit", "Cyril", "David", "Eloise", "Florent",
06         "Gerard", "Hugo", "Ingrid", "Jonathan", "Kevin", "Logan",
07         "Mathieu", "Noemie", "Olivia", "Philippe", "Quentin", "Romain", "Sophie",
08         "Tristan", "Ulric", "Vincent", "Willy", "Xavier", "Yann", "Zoé"};
09
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14
15         mListView = (ListView) findViewById(R.id.listView);
16
17         //android.R.layout.simple_list_item_1 est une vue disponible de base dans le SDK android,
18         //Contenant une TextView avec comme identifiant "@android:id/text1"
19         ArrayAdapter<String> adapter = new ArrayAdapter<String>(MainActivity.this,
20             android.R.layout.simple_list_item_1, prenom);
21         mListView.setAdapter(adapter);
22     }
23 }
```

Compilez maintenant l'application et lancez la sur votre appareil Android, le résultat devrait être le suivant :



Nous allons mettre un listener sur nos items, le principe est identique au bouton (`onClickListener`) mais l'interface à implémenter est différente : `AdapterView.OnItemClickListener`

On ajoute un listener à la liste :

```
mListView.setOnItemClickListener(this);
```

La classe Main implémente l'interface `OnItemClickListener` , lorsqu'on implémente une interface , on doit implémenter toutes les méthodes de cette interface, dans ce cas la méthode :

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id)
{
}
```

position : représente la position de l'item sélectionné dans la liste

```
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id)
{
    Toast.makeText(getApplicationContext(), "L'index de cet élément est : " +
        position, Toast.LENGTH_SHORT).show();
}
```

ListView : afficher une liste d'objets

Importer dans votre projet la classe Personne, créez une liste de personnes et afficher cette liste sur Android.

Déclarer un :

```
ArrayList<Personne> lesPersonnes = new ArrayList<Personne>();
```

Ajouter à cette liste une dizaine de personne:

```
Personne p = new Personne("Anouk",22,1.6,45);  
lesPersonnes.add(p);
```

etc

Afficher cette liste dans l'objet Adapter :

```
ArrayAdapter<Personne> adapter = new ArrayAdapter<Personne>(MainActivity.this,  
    android.R.layout.simple_list_item_1, lesPersonnes);
```

Afficher dans le Toast le nom et l'imc de la personne.