

TP : ListView & RecyclerView

Etape 1 : La listView de Base

RESSOURCES

Aller dans le fichier res>layout>activity_list.xml

Créer une liste view :

```
<ListView
    android:id="@+id/my_simple_list"
    android:layout_height="wrap_content"
    android:layout_width="match_parent">
</ListView>
```

NB : Il vous est possible de modifier la couleur, le padding de la liste ou autres via le style

JAVA

Aller dans ListActivity.java

Définir votre listview

```
ListView listView = (ListView) findViewById(R.id.my_simple_list);
```

Définir les valeurs de votre liste

```
String[] values = new String[] { "Item n°1",
    "Item n°2",
    "Item n°3",
    "Item n°4",
    "Item n°5",
    "Item n°6",
    "Item n°7",
    "Item n°8",
};
```

Définir l'adaptateur

```
final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, android.R.id.text1, values);
```

NB : ceci est l'adapter par défaut d'une liste

Lier l'adaptateur à la liste

```
listView.setAdapter(adapter);
```

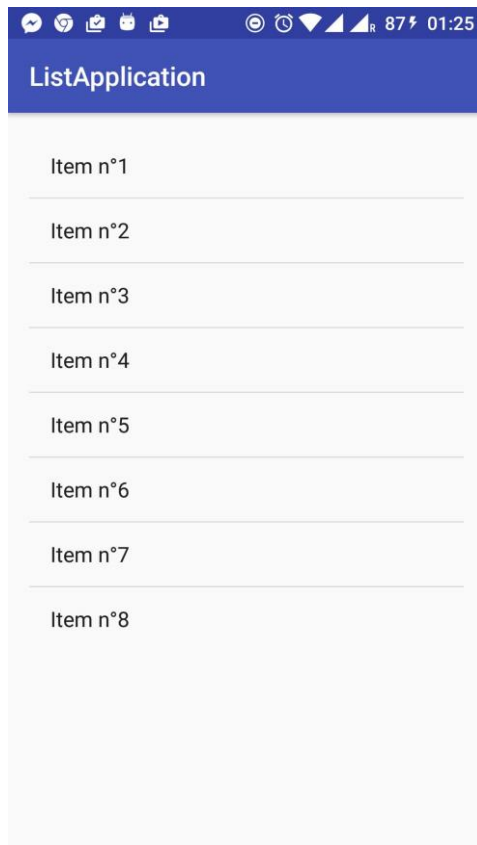
Rendre la vue clickable

```
listView.setClickable(true);
```

Définir le click

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i,  
long l) {  
        // Création d'un toast  
        Toast.makeText(getApplicationContext(), "Titre : " + adapterView.getItemAtPosition(i).toString(),  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

RENDU FINAL



Etape 2 : La listView avec adapter

RESSOURCES

Aller dans le layout de l'activité

```
res>layout>activity_list_adapter.xml
```

Ajouter la listView

```
<ListView
    android:id="@+id/my_adapter_list"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    >
</ListView>
```

Ajouter une nouvelle ressource

```
res>layout>row_item.xml
```

Créer le layout souhaité pour votre ligne

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

    <TextView
        android:id="@+id/numero"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Numéro" />

    <TextView
        android:id="@+id/titre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Titre" />

</LinearLayout>
```

JAVA

Création de l'objet de la ligne

Ici > Item.java

Création de l'adapter pour la liste

ItemAdapter.java

Override la méthode getView() dans l'adapter

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    // Récupère l'item à une position précise

    Item item = getItem(position);

    // Vérifie s'il existe une view, si oui récupère la vue de la ligne
    pour l'item
    if (convertView == null) {
        convertView = LayoutInflater.from(getContext()).inflate(R.layout.row_item, parent, false);
    }

    // Déclare les différents champs de la vue
    TextView itemNumero = (TextView) convertView.findViewById(R.id.numero);
    final TextView itemTitre = (TextView) convertView.findViewById(R.id.titre);

    // Met les éléments demandés dans les champs (NB : cela doit être une
    string)
    itemNumero.setText(String.valueOf(item.numero));
    itemTitre.setText(String.valueOf(item.titre));

    // Ajouter un click
    convertView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast.makeText(getContext(), "Titre : " + itemTitre.getText(),
                Toast.LENGTH_SHORT).show();
        }
    });

    // Retourne la vue de la ligne
    return convertView;
}
```

Pour finir dans ListAdapterActivity.java

Récupérer les items

```
ArrayList<Item> items = Item.getItems();
```

Initialiser notre adapter

```
ItemAdapter adapter = new ItemAdapter(this, items);
```

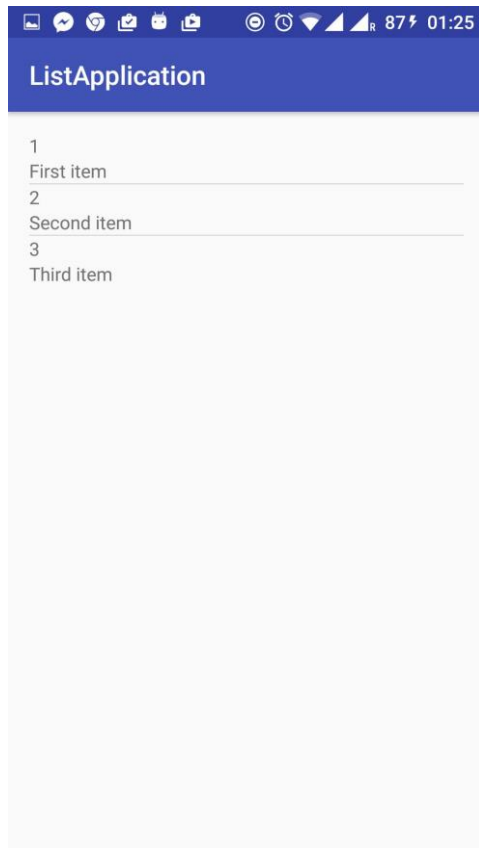
Déclarer la listview

```
ListView listView = (ListView) findViewById(R.id.my_adapter_list);
```

Lier notre liste à l'adapter

```
listView.setAdapter(adapter);
```

RENDU FINAL



Etape 3 : Le RecyclerView

RESSOURCES

Aller dans res>layout>list_item.xml && créer le layout pour la ligne

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/numero_recycler"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?attr/textAppearanceListItem" />

    <TextView
        android:id="@+id/titre_recycler"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?attr/textAppearanceListItem" />
</LinearLayout>
```

Aller dans res>layout>activity_recycler.xml et ajouter votre recyclerView

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_recycler"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="fr.univ_lille1.android.listapplication.RecyclerActiv-
ity">

    <android.support.v7.widget.RecyclerView xmlns:android="http://schemas.an-
droid.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/list"
        android:name="fr.univ_lille1.android.recycler.ItemFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        app:layoutManager="LinearLayoutManager"
        tools:context="fr.univ_lille1.android.recycler.ListFragment"
        tools:listitem="@layout/list_item" />

</RelativeLayout>
```

NB : tools :listitem => indique le layout que vous allez utiliser pour vos lignes

JAVA

Creation d'un adapter ItemAdapterRecycler.java

Creation d'un view holder qui sera votre ligne

```
public class MyViewHolder extends RecyclerView.ViewHolder implements Recy-
cyclerView.OnClickListener{
    public TextView numero, titre ;

    public MyViewHolder(View view) {
        super(view);
        numero = (TextView) view.findViewById(R.id.numero_recycler);
        titre = (TextView) view.findViewById(R.id.titre_recycler);
        view.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {

        Toast.makeText(context, "titre : " + titre.getText() ,
Toast.LENGTH_SHORT).show();

    }
}
```

NB : sur ce view holder vous pouvez définir différents listeners pour cette même ligne. Du fait que le recycler view utilise des itemAnimations.

Utilisation des methodes onCreate et onBind

```
@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.list_item, parent, false);

    return new MyViewHolder(itemView);
}

@Override
public void onBindViewHolder(MyViewHolder holder, int position) {
    Item item = itemList.get(position);
    holder.numero.setText(String.valueOf(item.numero));
    holder.titre.setText(String.valueOf(item.titre));
}
```

La première méthode sert à créer le view holder la seconde à attribuer une valeur au champ de la vue.

RecyclerView.java

Contient 3 objets :

```
private RecyclerView mRecyclerView;
private RecyclerView.Adapter mAdapter;
private RecyclerView.LayoutManager mLayoutManager;
```

Récupérer la liste d'item

```
List<Item> myItems = Item.getItems();
```

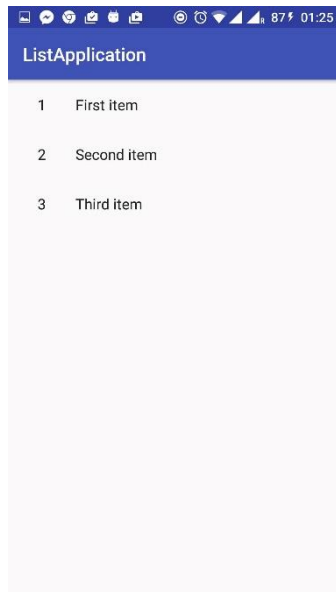
Définir le recycler view et le layout manager

```
mRecyclerView = (RecyclerView) findViewById(R.id.list);
mLayoutManager = new LinearLayoutManager(this);
mRecyclerView.setLayoutManager(mLayoutManager);
```

Adapter pour la liste d'items

```
mAdapter = new ItemAdapterRecycler(myItems, getApplicationContext());
mRecyclerView.setAdapter(mAdapter);
```

RENDU FINAL



SOURCES

<http://www.vogella.com/tutorials/AndroidListView/article.html>

<https://developer.android.com/training/material/lists-cards.html>

<http://feanorin.developpez.com/tutoriels/android/composant-graphique-recyclerview/>