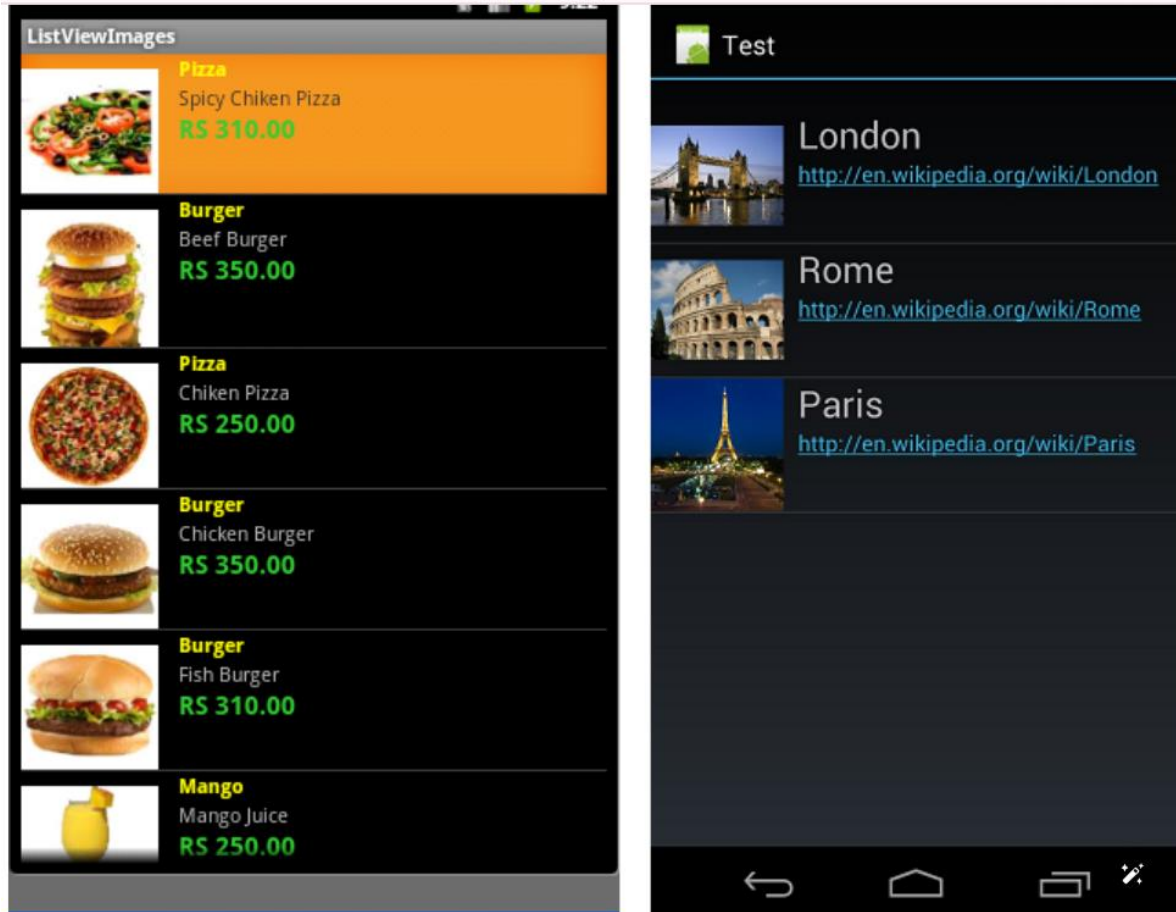
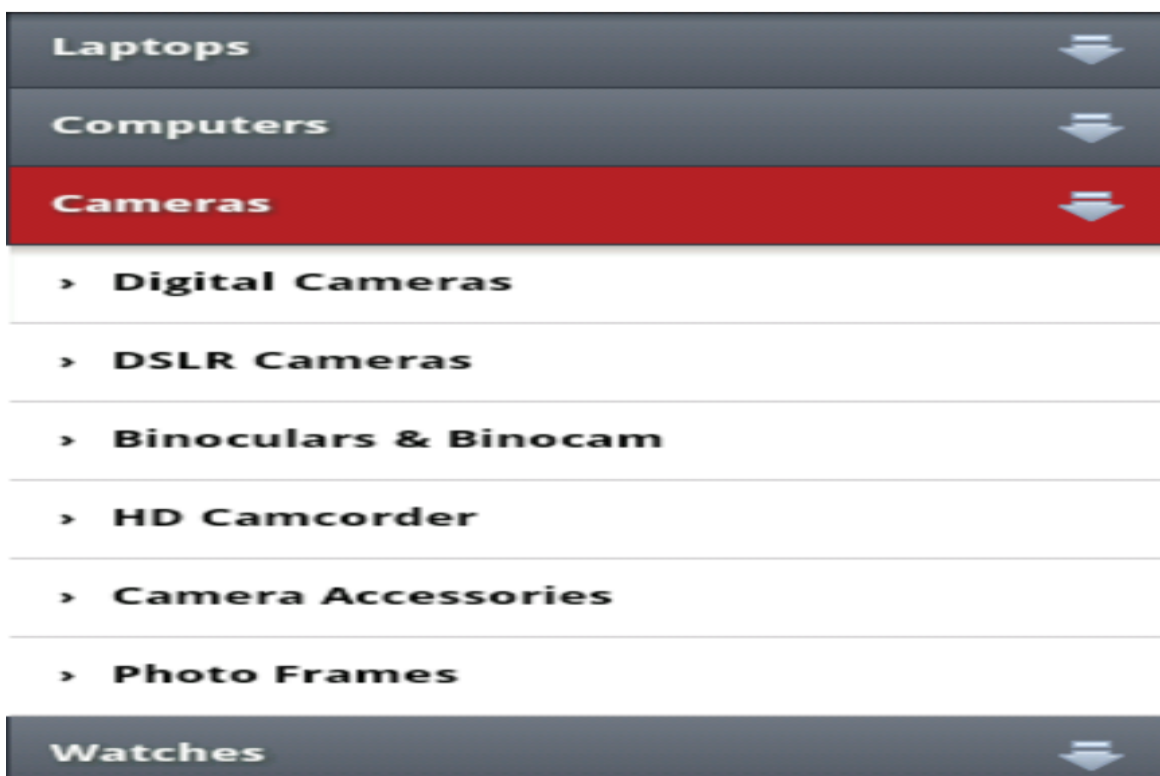


ListView est un view group, affiche des éléments (elements) selon une liste et peut être déplacé verticalement. ListView est une vue importante et est largement utilisé dans les applications Android.

Un simple exemple de ListView est votre carnet de contacts, où vous avez une liste de vos contacts affichés dans un ListView.



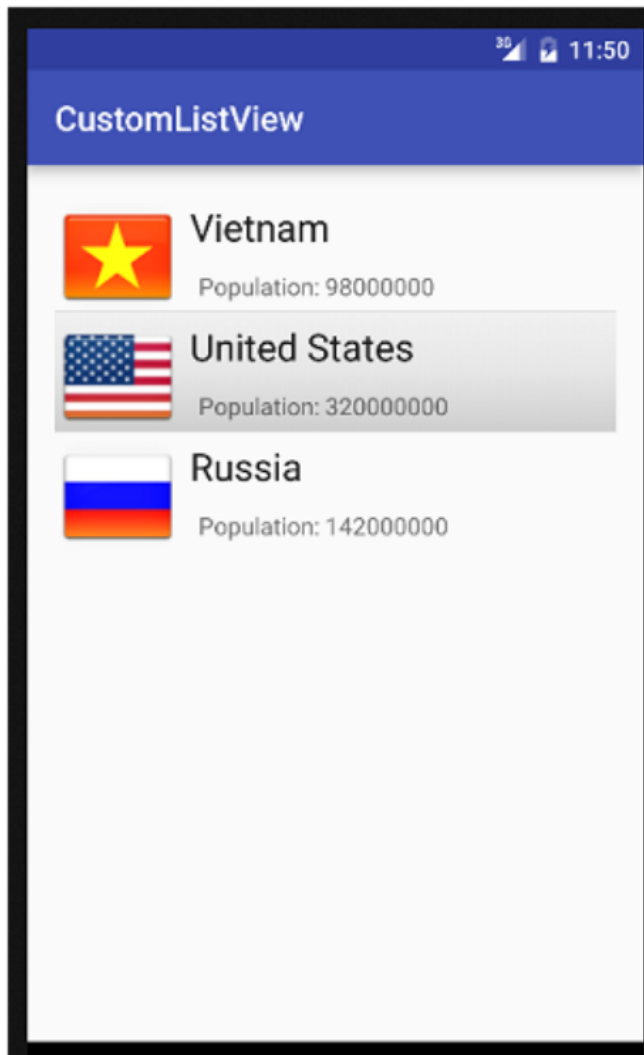
En plus de ListView, Android vous fournit un autre view similaire qui est ExpandableListView.



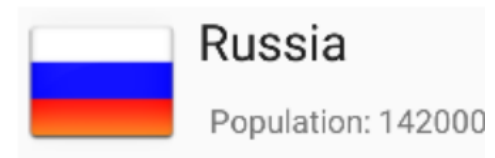
ListItem

Un ListView est fait à partir d'un groupe de ListItem. ListItem est une ligne (row) individuelle dans listview où les données seront affichées. Toutes les données dans listview sont affichées uniquement via listItem. Considérez ListView comme groupe défilable de ListItems.

ListView



ListItem



Un ListItem est une pièce de l'interface qui peut être créée par un nombre de View.

ListItem

Android construit certaine forme ListItem différente, appelées le Layout pré- défini, qui sera mentionné dans les exemples de ce document.

Adapter (L'adaptateur)

Android Adapter (L'adaptateur) est un pont entre des View (par exemple comme ListView) et les données sous-jacentes pour ce View. Un Adapter gère des données et adapte les données dans les lignes individuelles (ListItem) du view.

Vous pouvez lier des Adapter avec Android ListView via la méthode `setAdapter`. Maintenant, nous allons voir comment Adapter fonctionne à l'aide de l'image ci-dessous.

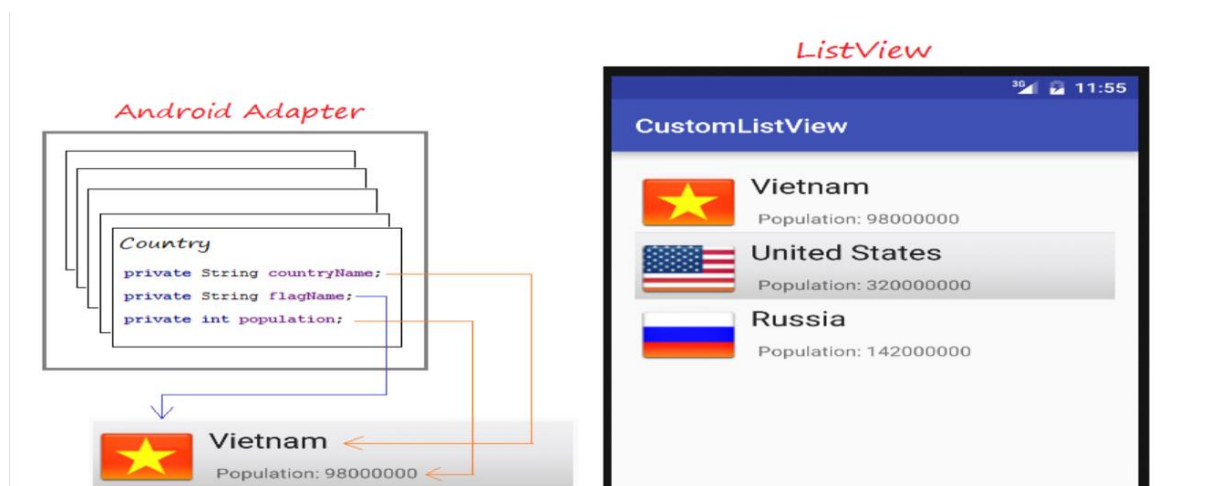


Example:

Array, Cursor,...

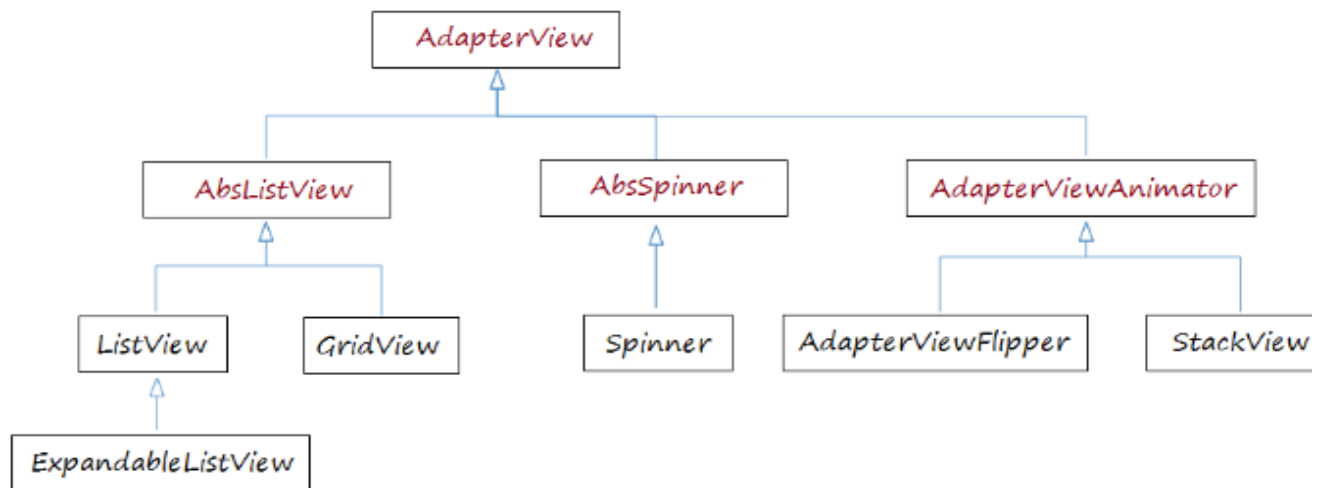
Example:

GridView, ListView,
Spinner, StackView, ...

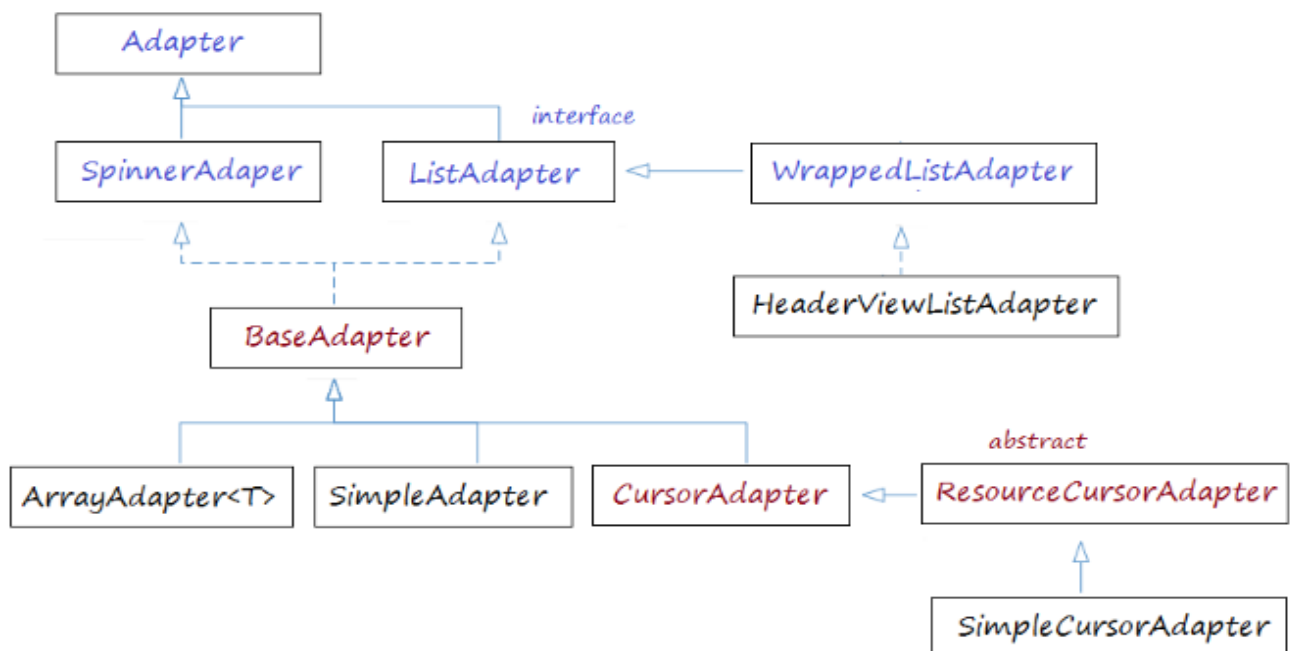


AdapterView

.Il existe plusieurs View ont eu besoin d'Android Adapter en vue de gérer des données affichées, ces View sont les sous-classes de la classe AdapterView, vous pouvez voir l'illustration suivante:

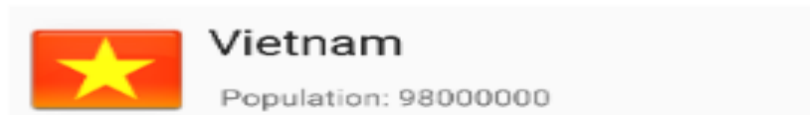


Android Adapter



ListView Selector

Pour rendre ListView plus beau, vous devez personnaliser les effets, tels que modifier la couleur d'arrière-plan de ListItem lorsque le curseur se déplace dessus ou modifier la couleur d'arrière-plan lorsqu'il est sélectionné. Vous pouvez voir un exemple pour personnaliser le ListView Selector à la fin de ce document.



When ListItem in normal state



When ListItem in selected state



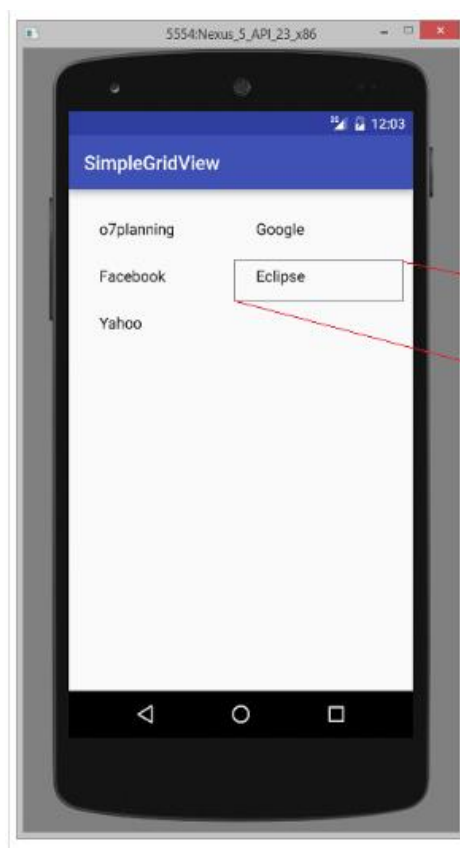
When ListItem in Pressed state

2. ListView basic utilisant ArrayAdapter

ArrayAdapter

ArrayAdapter a été utilisé à afficher le ListView avec des ListItem simple, ListItem peut être fabriqué à partir seulement un TextView, CheckedTextView, EditText,...

Au cas où vous voulez avoir un ListView avec ListItem plus compliqué, vous pouvez créer manuellement un Adapter personnalisé.

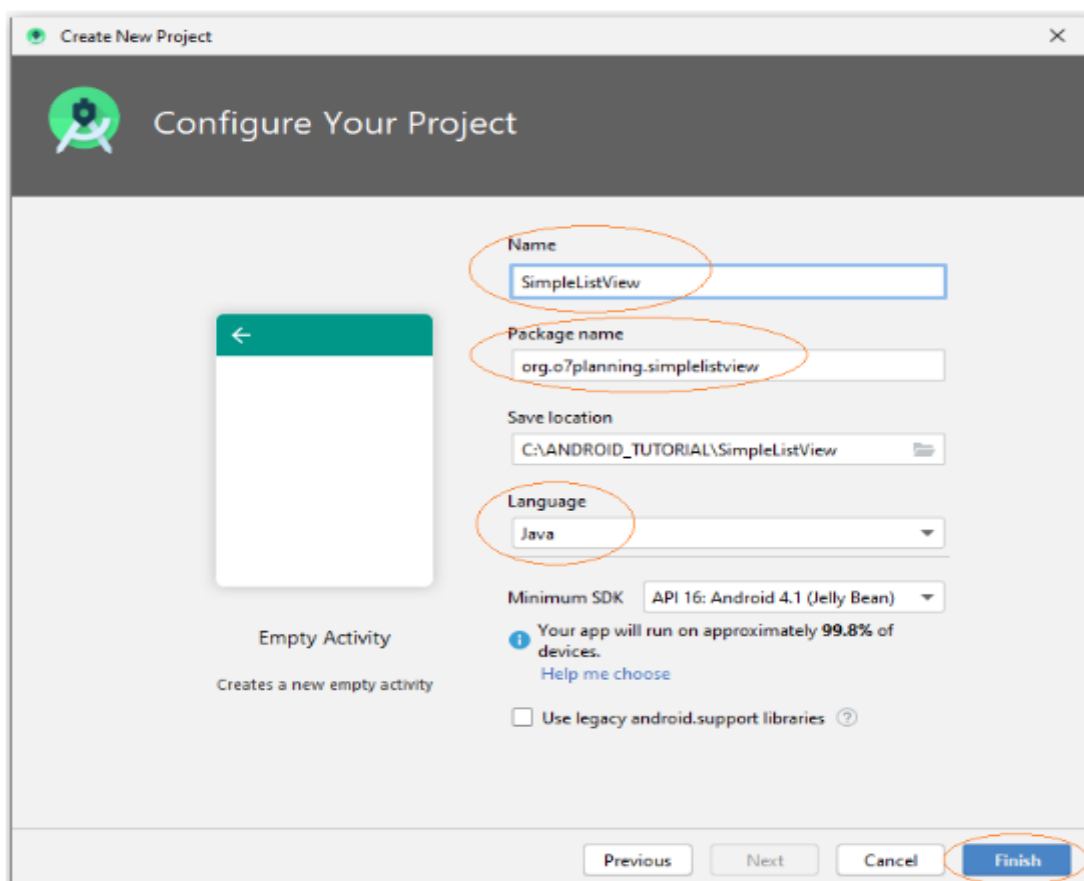
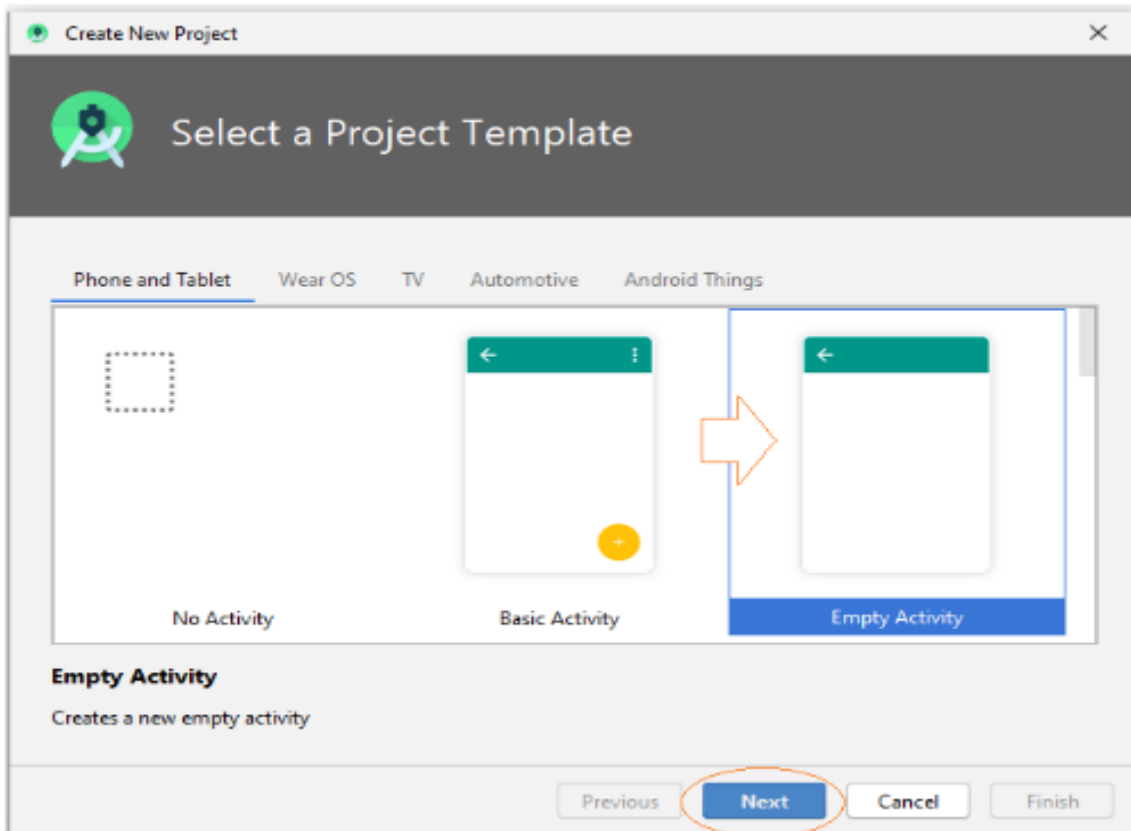


*ArrayAdapter used to create simple
GridView, ListView, ...
With items made from a TextView*

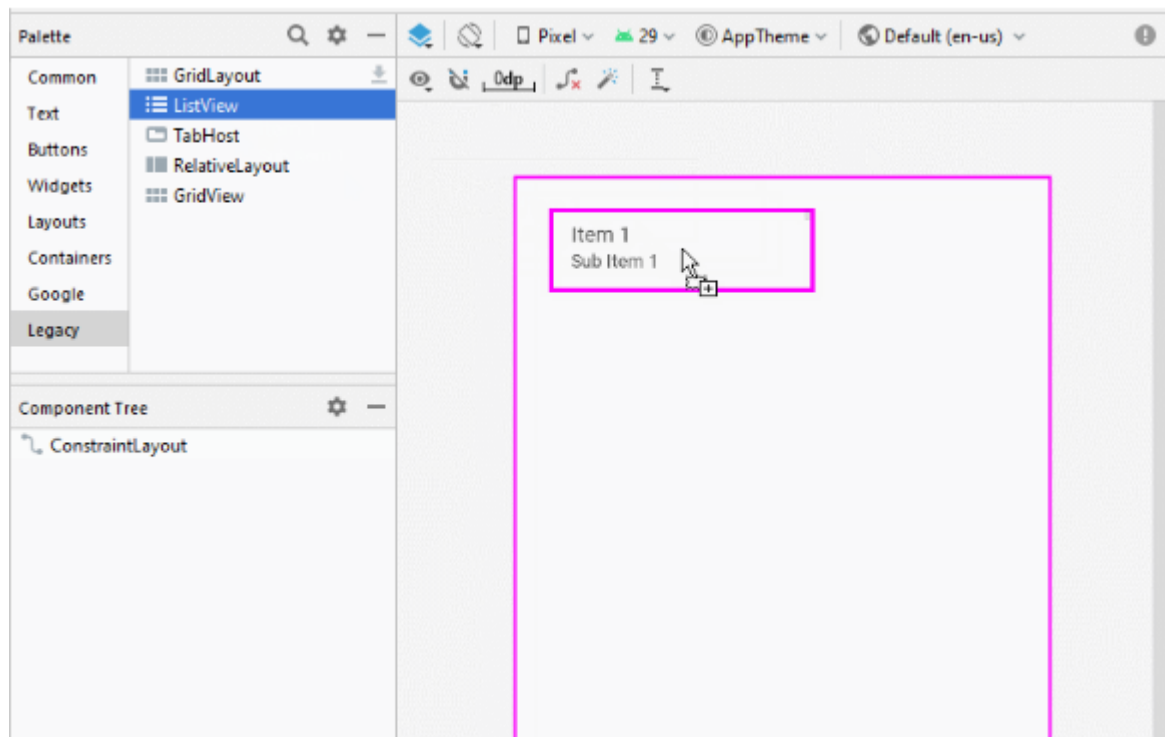
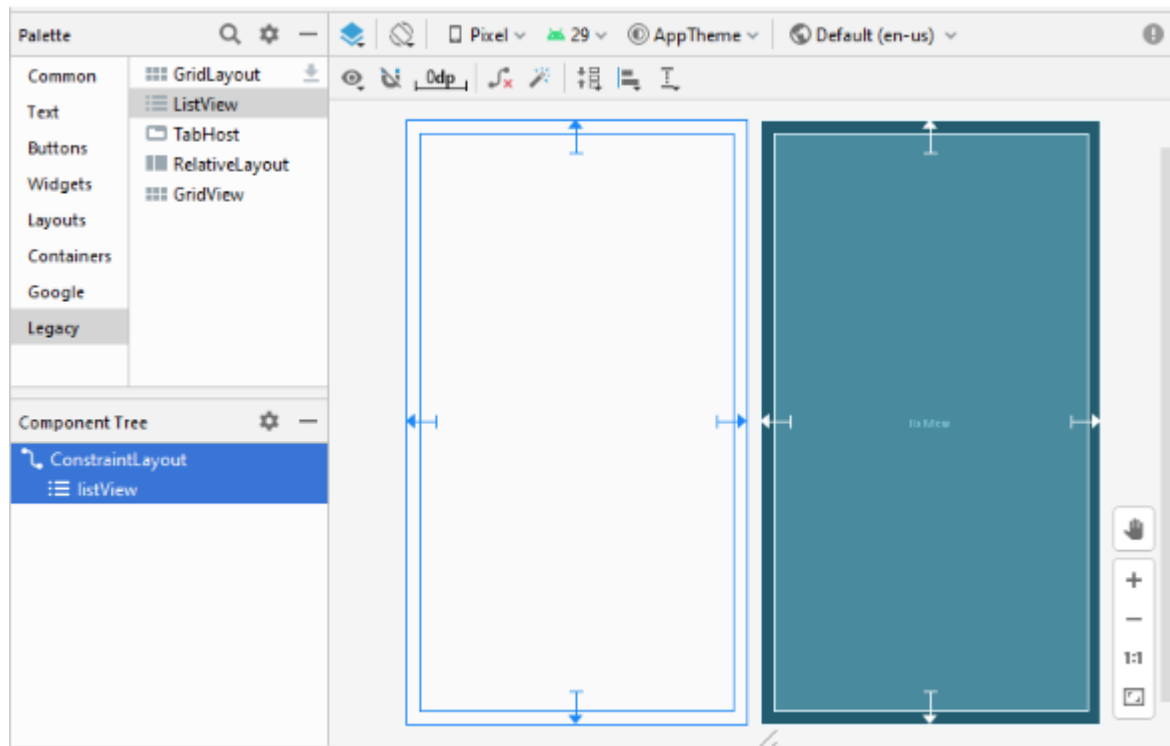
TextView

L'exemple de ListView et ArrayAdapter

Créer un nouveau projet nommé SimpleListView.



The interface of the application:



activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/listView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

```

UserAccount.java

```

package org.o7planning.simplelistview;

import java.io.Serializable;

public class UserAccount implements Serializable {

```



```
private String userName;

private String userType;

private boolean active;

public UserAccount(String userName, String userType) {
    this.userName= userName;
    this.userType = userType;
    this.active= true;
}

public UserAccount(String userName, String userType, boolean active) {
    this.userName= userName;
    this.userType = userType;
    this.active= active;
}

public String getUserType() {
    return userType;
}

public void setUserType(String userType) {
    this.userType = userType;
}

public String getUserName() {
    return userName;
}

public void setUserName(String userName) {
```

```
        this.userName = userName;
    }

    public boolean isActive() {
        return active;
    }

    public void setActive(boolean active) {
        this.active = active;
    }

    @Override
    public String toString() {
        return this.userName + " (" + this.userType + ")";
    }
}
```

MainActivity.java

```
package org.o7planning.simplelistview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {

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

```
setContentView(R.layout.activity_main);

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

//

UserAccount tom = new UserAccount("Tom","admin");
UserAccount jerry = new UserAccount("Jerry","user");
UserAccount donald = new UserAccount("Donald","guest", false);

UserAccount[] users = new UserAccount[]{tom,jerry, donald};

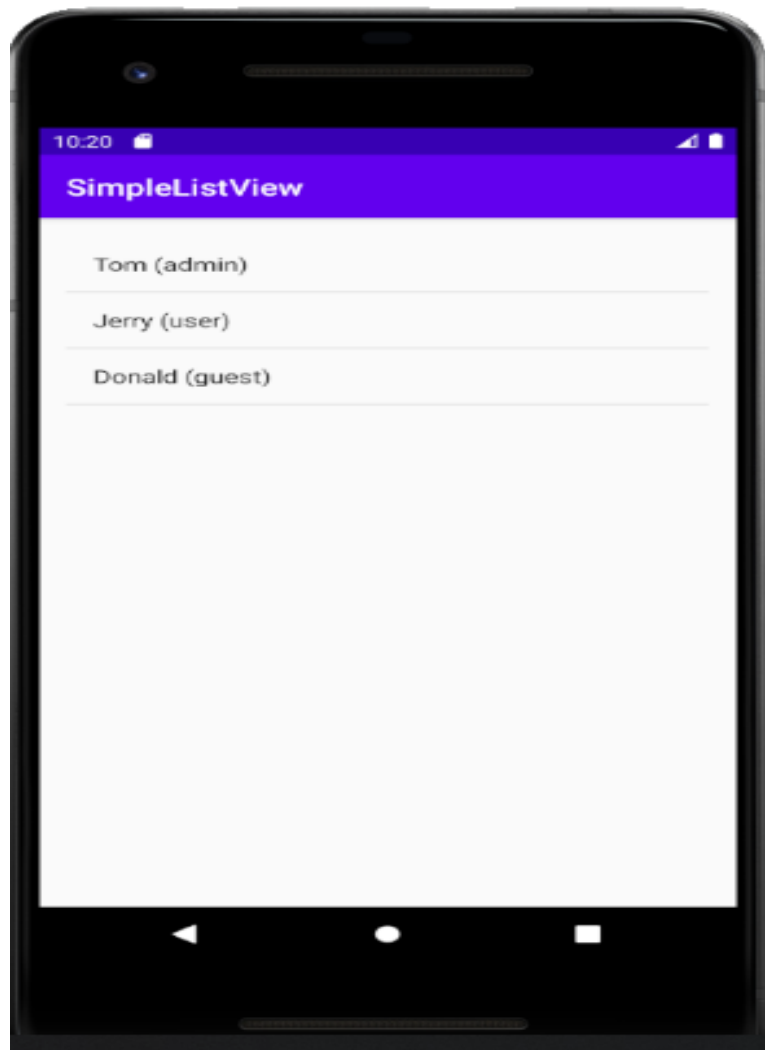
// android.R.layout.simple_list_item_1 is a constant predefined layout of Android.
// used to create a ListView with simple ListItem (Only one TextView).

ArrayAdapter<UserAccount> arrayAdapter
    = new ArrayAdapter<UserAccount>(this, android.R.layout.simple_list_item_1 , users);

listView.setAdapter(arrayAdapter);
}

}
```

L'exécution de l'exemple:



Les mises en page sont disponibles pour fonctionner avec ArrayAdapter

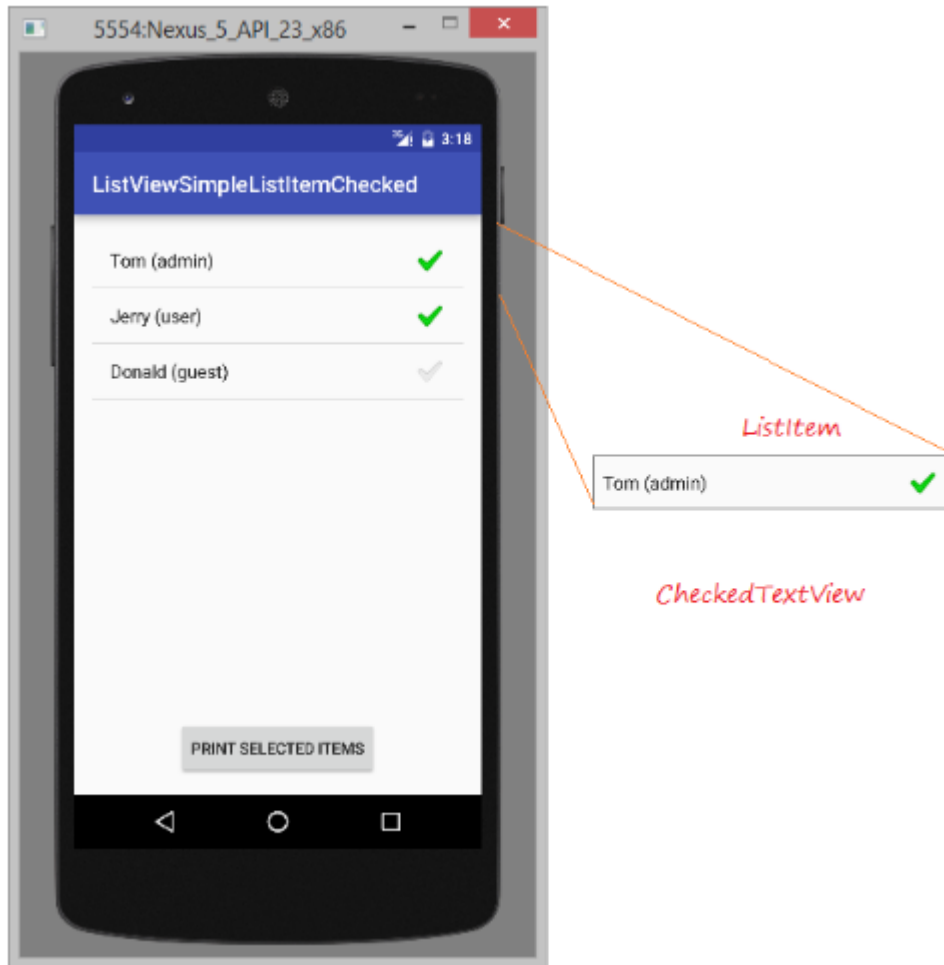
Android construit certains Layout (pour ListItem) qui peuvent fonctionner avec ArrayAdapter.

`android.R.layout.simple_list_item_1`

C'est un Layout simple de ListItem, créé par un seul TextView (Vous pouvez voir des exemples ci-dessus).

`android.R.layout.simple_list_item_checked` & `android.R.layout.simple_list_item_multiple_choice`

Deux Layout ci-dessus sont les mises en page simples pour créer un ListView avec ListItem créé par un seul checkbox.



Vous pouvez voir cet exemple ici:

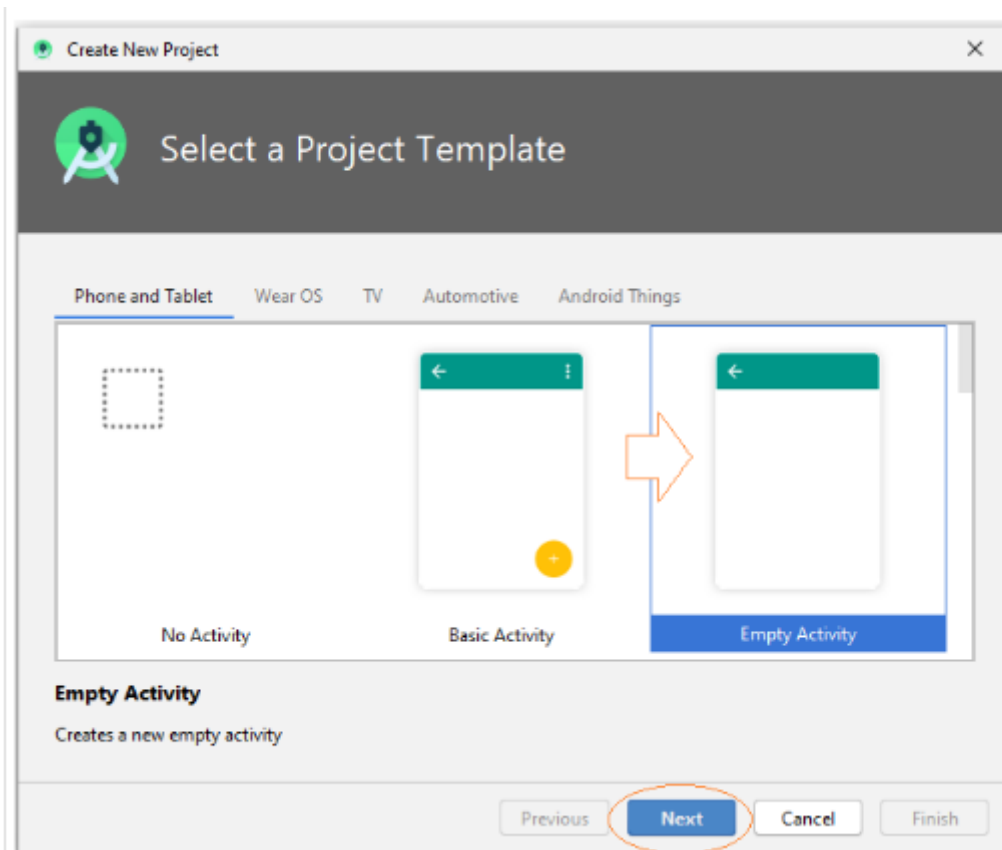
Android ListView avec Checkbox en utilisant ArrayAdapter

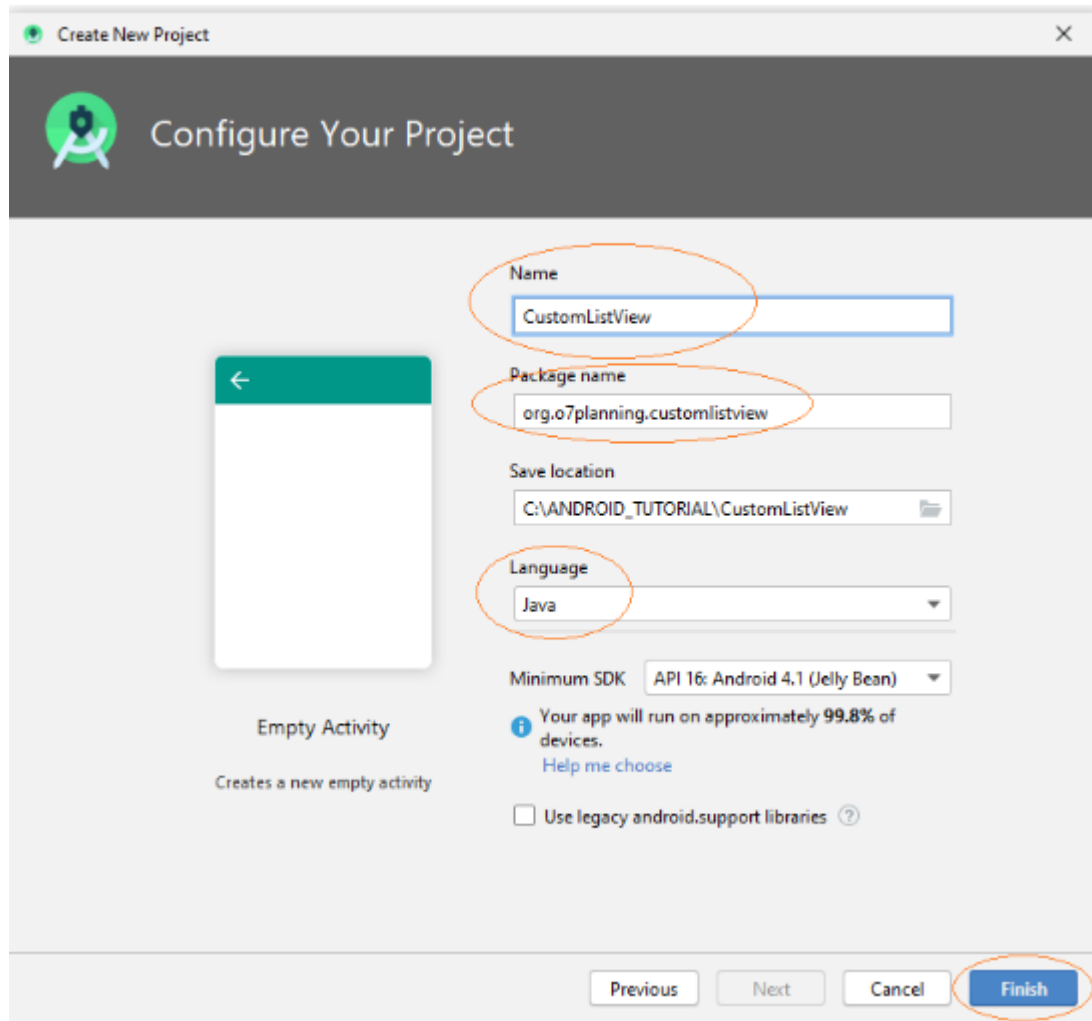
3. La personnalisation de ListView utilisant BaseAdapter

Vous pouvez personnaliser un ListView. Votre Adapter devrait étendre à partir de la classe BaseAdapter.

L'exemple de la personnalisation de ListView

Créer un projet "Empty Activity" nommé CustomListView.



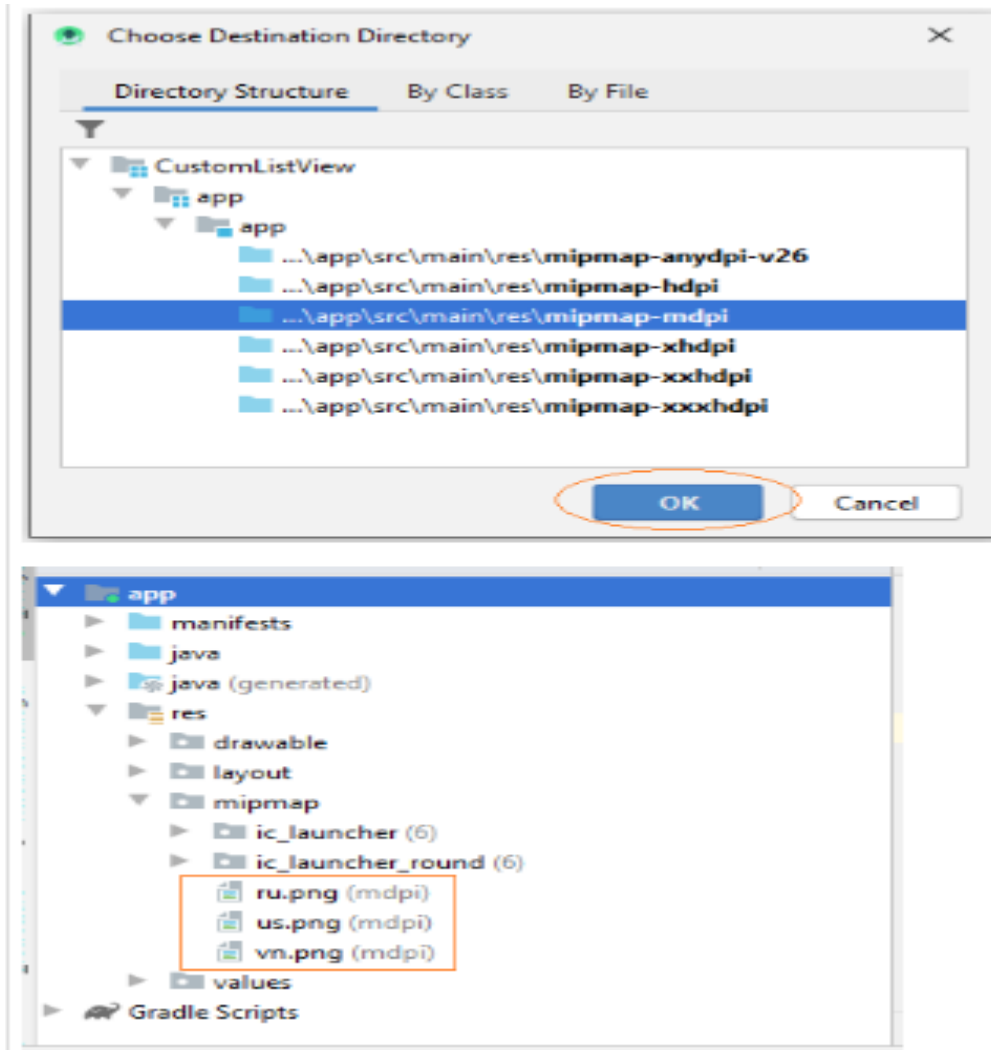


Voici sont l'aperçu de l'application qu'on fera:



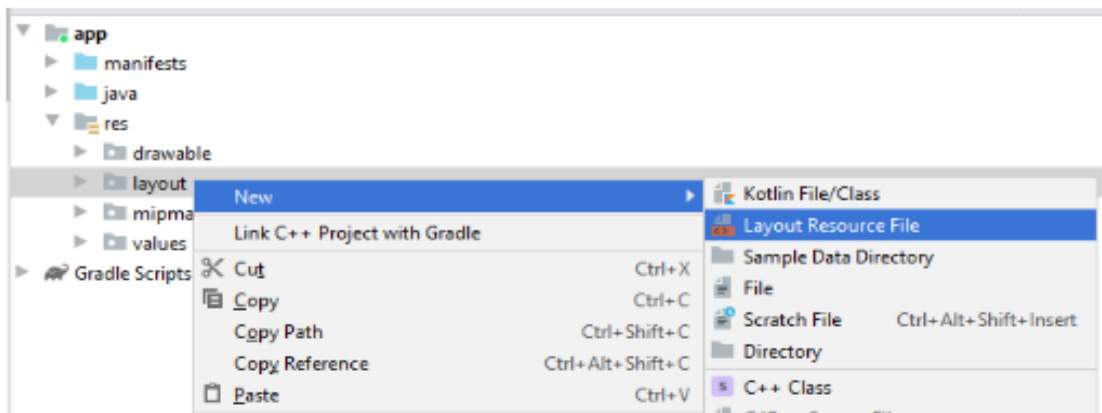
Tout d'abord, vous devez préparer quelques images:

Copiez et collez le fichier de l'image dans le dossier mipmap:



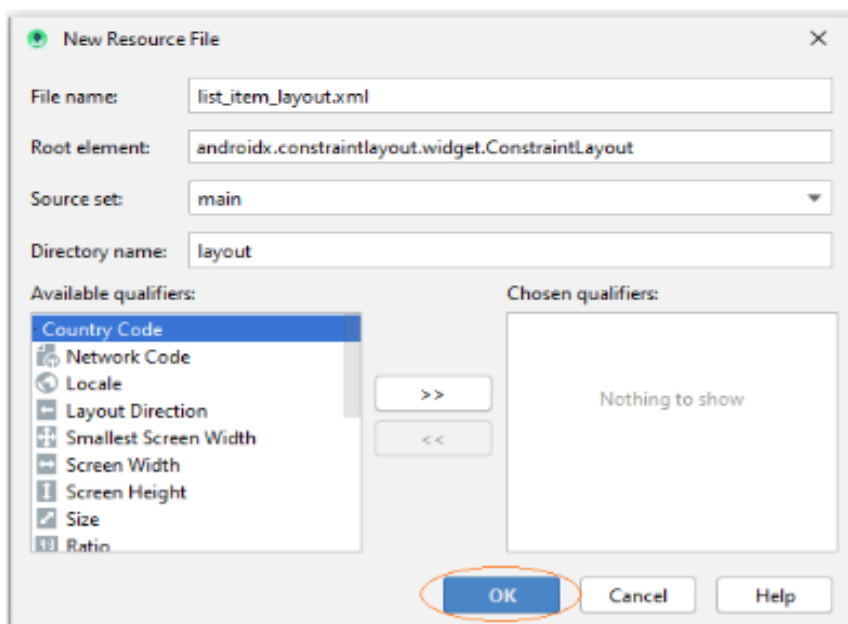
Vous devriez créer un layout pour Listitem. Sur Android Studio, cliquez sur le bouton droit de res/layout et sélectionnez:

New/Layout resource file

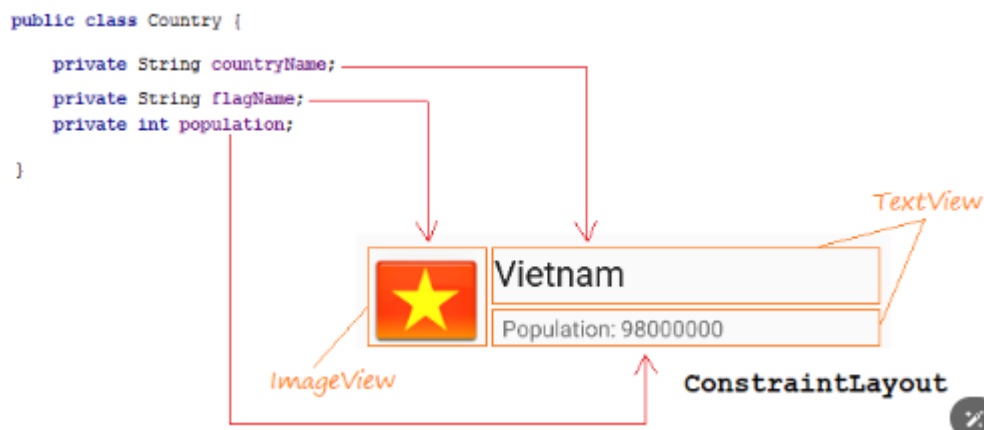


Saisissez:

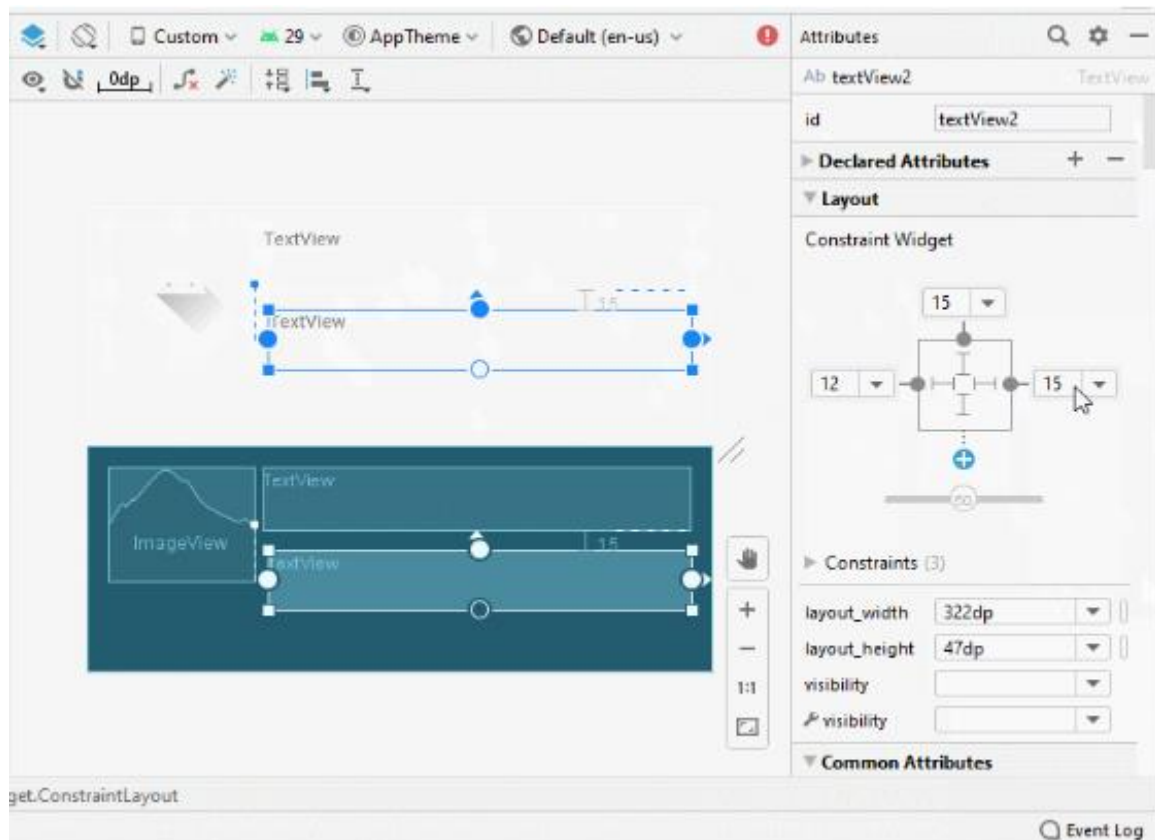
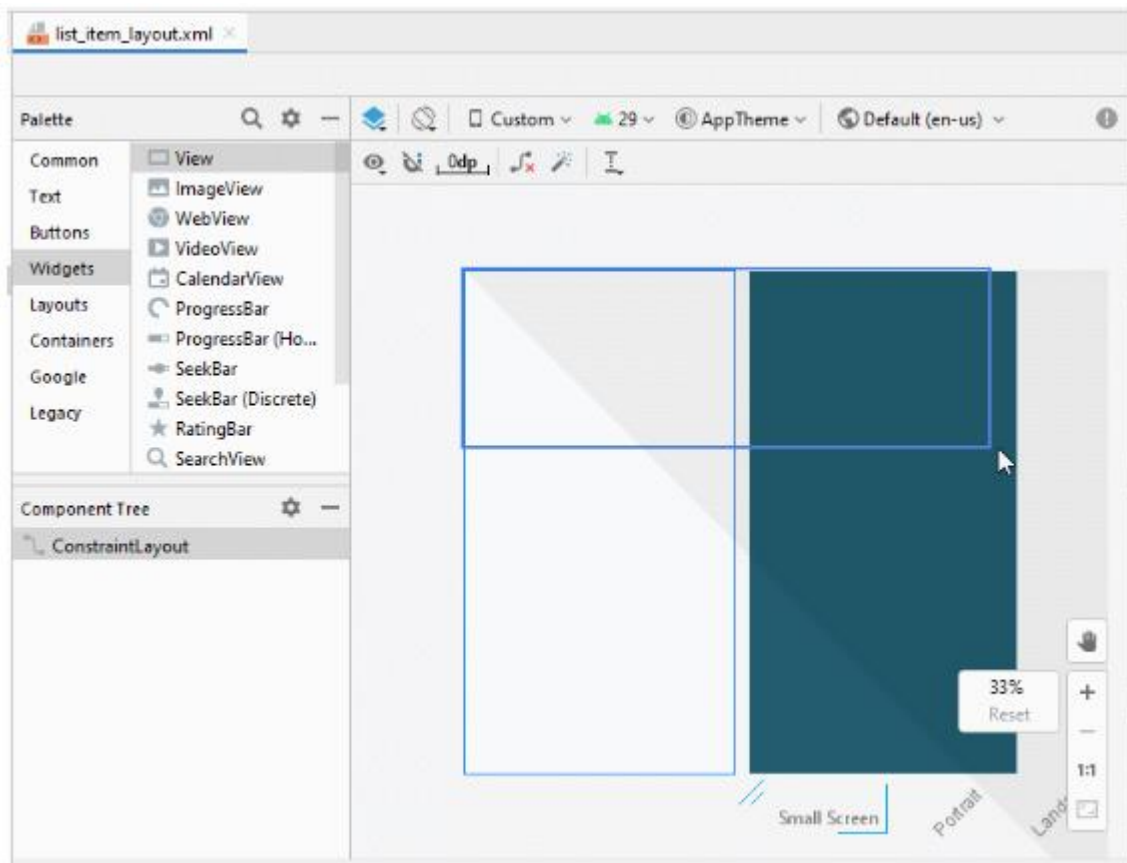
- **File name:** list_item_layout.xml
- **Root element:** androidx.constraintlayout.widget.ConstraintLayout



La conception de l'interface pour List Item.



Interface design:



Set ID, Text for components on the interface.

ImageView

ID: imageView_flag

TextView 1:

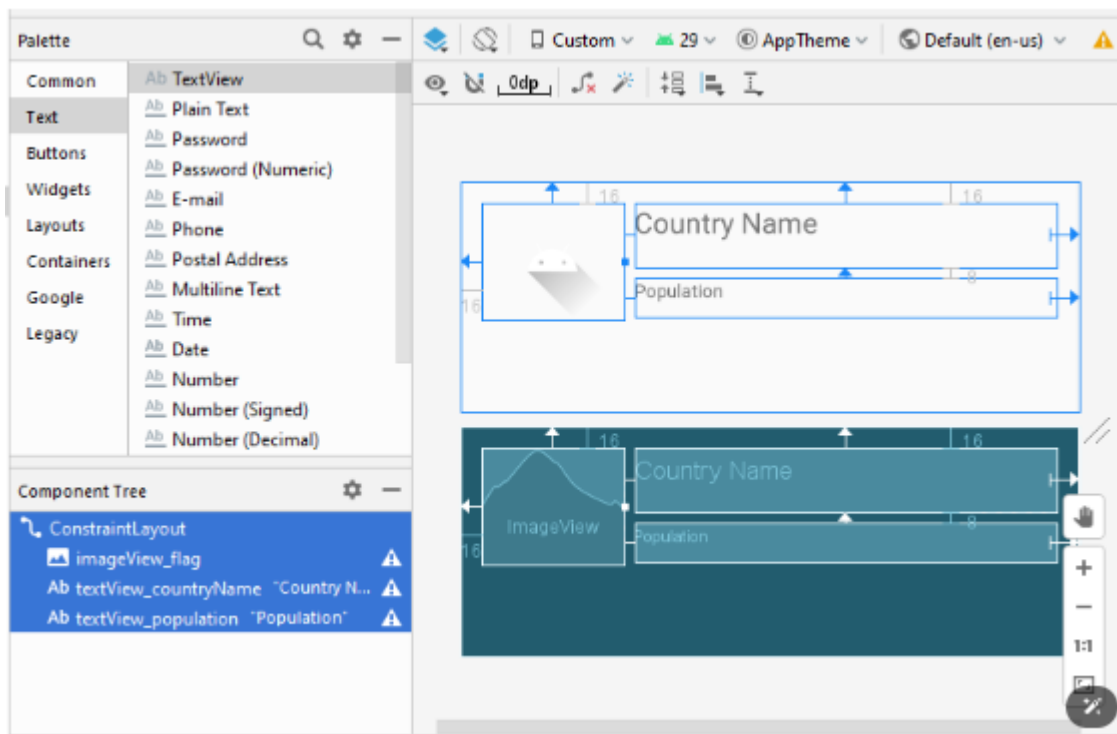
ID: textView_countryName

Text: Country Name

TextView 2:

ID: textView_population

Text: Population



list_item_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

<ImageView

```
    android:id="@+id/imageView_flag"
    android:layout_width="110sp"
    android:layout_height="90sp"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_launcher_foreground"
    tools:ignore="VectorDrawableCompat" />
```

<TextView

```
    android:id="@+id/textView_countryName"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:text="Country Name"
    android:textSize="22sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/imageView_flag"
    app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
    android:id="@+id/textView_population"
    android:layout_width="0dp"
    android:layout_height="30dp"
```

```

        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:text="Population"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageView_flag"
        app:layout_constraintTop_toBottomOf="@+id/textView_countryName" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Country.java

```

package org.o7planning.customlistview;

public class Country {

    private String countryName;

    // Image name (Without extension)
    private String flagName;
    private int population;

    public Country(String countryName, String flagName, int population) {
        this.countryName= countryName;
        this.flagName= flagName;
        this.population= population;
    }

    public int getPopulation() {
        return population;
    }

    public void setPopulation(int population) {

```

```

        this.population = population;
    }

    public String getCountryName() {
        return countryName;
    }

    public void setCountryName(String countryName) {
        this.countryName = countryName;
    }

    public String getFlagName() {
        return flagName;
    }

    public void setFlagName(String flagName) {
        this.flagName = flagName;
    }

    @Override
    public String toString() {
        return this.countryName+" (Population: "+ this.population+" )";
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

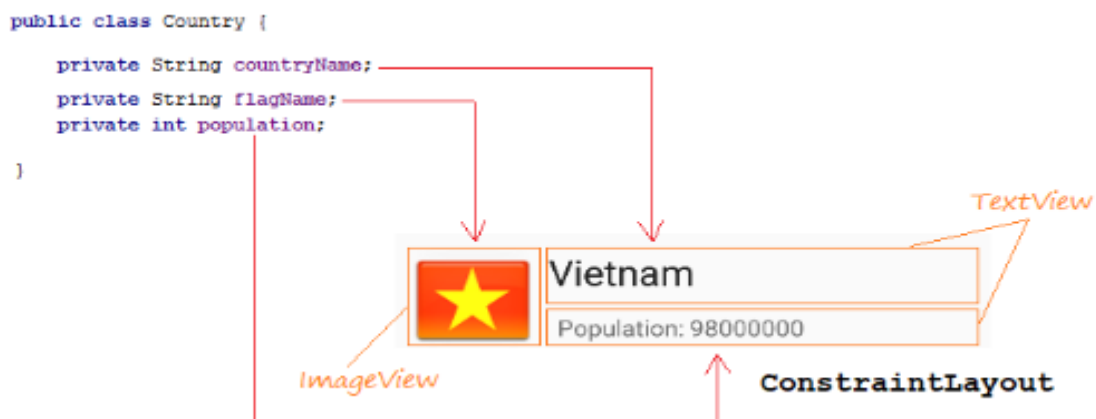
```

android:layout_height="match_parent"
tools:context=".MainActivity">

<ListView
    android:id="@+id/listView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

CustomListAdapter est une classe étendue de BaseAdapter, sert à afficher des données sur des List Item.



CustomListAdapter.java

```
package org.o7planning.customlistview;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import java.util.List;

public class CustomListAdapter extends BaseAdapter {

    private List<Country> listData;
    private LayoutInflater layoutInflater;
    private Context context;

    public CustomListAdapter(Context aContext, List<Country> listData) {
        this.context = aContext;
        this.listData = listData;
        layoutInflater = LayoutInflater.from(aContext);
    }

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

    @Override
```



```
public Object getItem(int position) {
    return listData.get(position);
}

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

public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;

    if (convertView == null) {
        convertView = inflater.inflate(R.layout.list_item_layout, null);
        holder = new ViewHolder();
        holder.flagView = (ImageView) convertView.findViewById(R.id.imageView_flag);
        holder.countryNameView = (TextView)
convertView.findViewById(R.id.textView_countryName);
        holder.populationView = (TextView) convertView.findViewById(R.id.textView_population);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    Country country = this.listData.get(position);
    holder.countryNameView.setText(country.getCountryName());
    holder.populationView.setText("Population: " + country.getPopulation());

    int imageId = this.getMipmapResIdByName(country.getFlagName());

    holder.flagView.setImageResource(imageId);
}
```

```

        return convertView;
    }

    // Find Image ID corresponding to the name of the image (in the directory mipmap).
    public int getMipmapResIdByName(String resName) {
        String pkgName = context.getPackageName();

        // Return 0 if not found.

        int resID = context.getResources().getIdentifier(resName, "mipmap", pkgName);

        Log.i("CustomListView", "Res Name: "+ resName+"==> Res ID = "+ resID);

        return resID;
    }

    static class ViewHolder {
        ImageView flagView;
        TextView countryNameView;
        TextView populationView;
    }
}

```

ActivityMain.java

```

package org.o7planning.customlistview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.List;

```

```

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);


        List<Country> image_details = getListData();

        final ListView listView = (ListView) findViewById(R.id.listView);

        listView.setAdapter(new CustomListAdapter(this, image_details));


        // When the user clicks on the ListItem

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override

            public void onItemClick(AdapterView<?> a, View v, int position, long id) {

                Object o = listView.getItemAtPosition(position);

                Country country = (Country) o;

                Toast.makeText(MainActivity.this, "Selected :" + " " + country, Toast.LENGTH_LONG).show();

            }

        });

    }


    private List<Country> getListData() {

        List<Country> list = new ArrayList<Country>();

        Country vietnam = new Country("Vietnam", "vn", 980000000);

        Country usa = new Country("United States", "us", 320000000);

        Country russia = new Country("Russia", "ru", 142000000);
    }
}

```

```
list.add(vietnam);  
  
list.add(usa);  
  
list.add(russia);  
  
return list;  
}  
  
}
```