



Affichage automatisé de données

Plan

1 ListView

Plan

1 ListView

Principe général



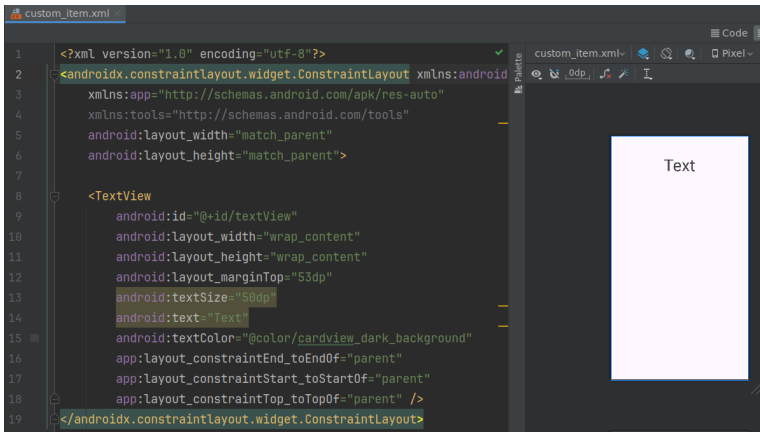
1. android.widget.ListView

Version la plus simple, avec une mise en forme prédéfinie :
`android.R.layout.simple_list_item_1`

```
public class SimpleListView extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_simple_list_view);  
  
        String[] departments = {"Info", "GEA", "Tech2Co"};  
  
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<>(  
            context: this,  
            android.R.layout.simple_list_item_1,  
            departments);  
  
        ListView listView = findViewById(R.id.listView);  
  
        listView.setAdapter(dataAdapter);  
    }  
}
```

2. android.widget.ListView

Version avec une mise en forme personnalisée : custom_item.xml



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <TextView
9         android:id="@+id/textView"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:layout_marginTop="53dp"
13        android:textSize="50dp"
14        android:text="Text"
15        android:textColor="@color/cardview_dark_background"
16        app:layout_constraintEnd_toEndOf="parent"
17        app:layout_constraintStart_toStartOf="parent"
18        app:layout_constraintTop_toTopOf="parent" />
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

The screenshot shows the XML code in the left pane and a visual preview in the right pane. The preview shows a white rectangular box with the word "Text" centered inside it.

2. android.widget.ListView

Utilisation de la mise en page personnalisée dans la définition de l'ArrayAdapter

```
public class CustomListView extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_custom_list_view);  
  
        String[] departments = {"Info", "GEA", "Tech2C0"};  
  
        ListView listView = findViewById(R.id.listView2);  
  
        ArrayAdapter<String> dataAdapter = new ArrayAdapter<>(  
            context: this,  
            R.layout.custom_item,  
            R.id.textView,  
            departments);  
  
        listView.setAdapter(dataAdapter);  
    }  
}
```

3. Avec un ArrayAdapter customisé

Utilisation de plusieurs types de données à passer à l'Adapter customisé

```
public class CustomAdapterExample extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_custom_adapter);  
  
        String[] departments = {"Info", "GEA", "Tech2C0"};  
  
        int[] imageID = {  
            R.drawable.iut,  
            R.drawable.iut,  
            R.drawable.iut  
        };  
  
        ListView listView = findViewById(R.id.listView2);  
  
        CustomArrayAdapter dataAdapter = new CustomArrayAdapter(  
            context: this,  
            departments,  
            imageID);  
  
        listView.setAdapter(dataAdapter);  
    }  
}
```


3. Avec un ArrayAdapter customisé

Définition de l'Adapter customisé : renvoie la construction d'une vue

```
public class CustomArrayAdapter extends ArrayAdapter<String> {  
    2 usages  
    private Activity context;  
    2 usages  
    private String[] title;  
    2 usages  
    private int[] imageID;  
  
    1 usage  
    public CustomArrayAdapter(@NonNull Activity context, String[] title, int[] imageID) {  
        super(context,R.layout.custom_item,title);  
        this.context = context;  
        this.title = title;  
        this.imageID = imageID;  
    }  
    @NonNull  
    @Override  
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
        LayoutInflater inflater = context.getLayoutInflater();  
        View v = inflater.inflate(R.layout.custom_item2, root: null, attachToRoot: true);  
  
        TextView tv = v.findViewById(R.id.textView);  
        ImageView iv = v.findViewById(R.id.thumbnail);  
  
        tv.setText(title[position]);  
        iv.setImageResource(imageID[position]);  
        return v;  
    }  
}
```