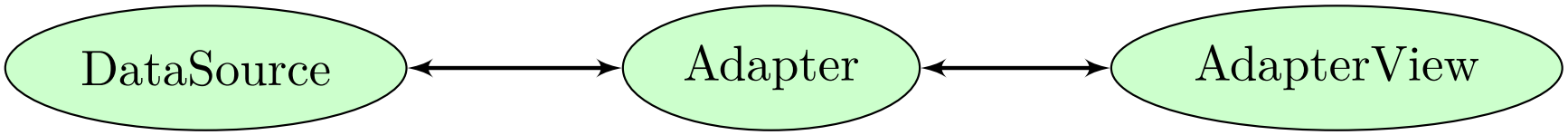
Affichage automatisé de données

CC BY NC SA

Principe général



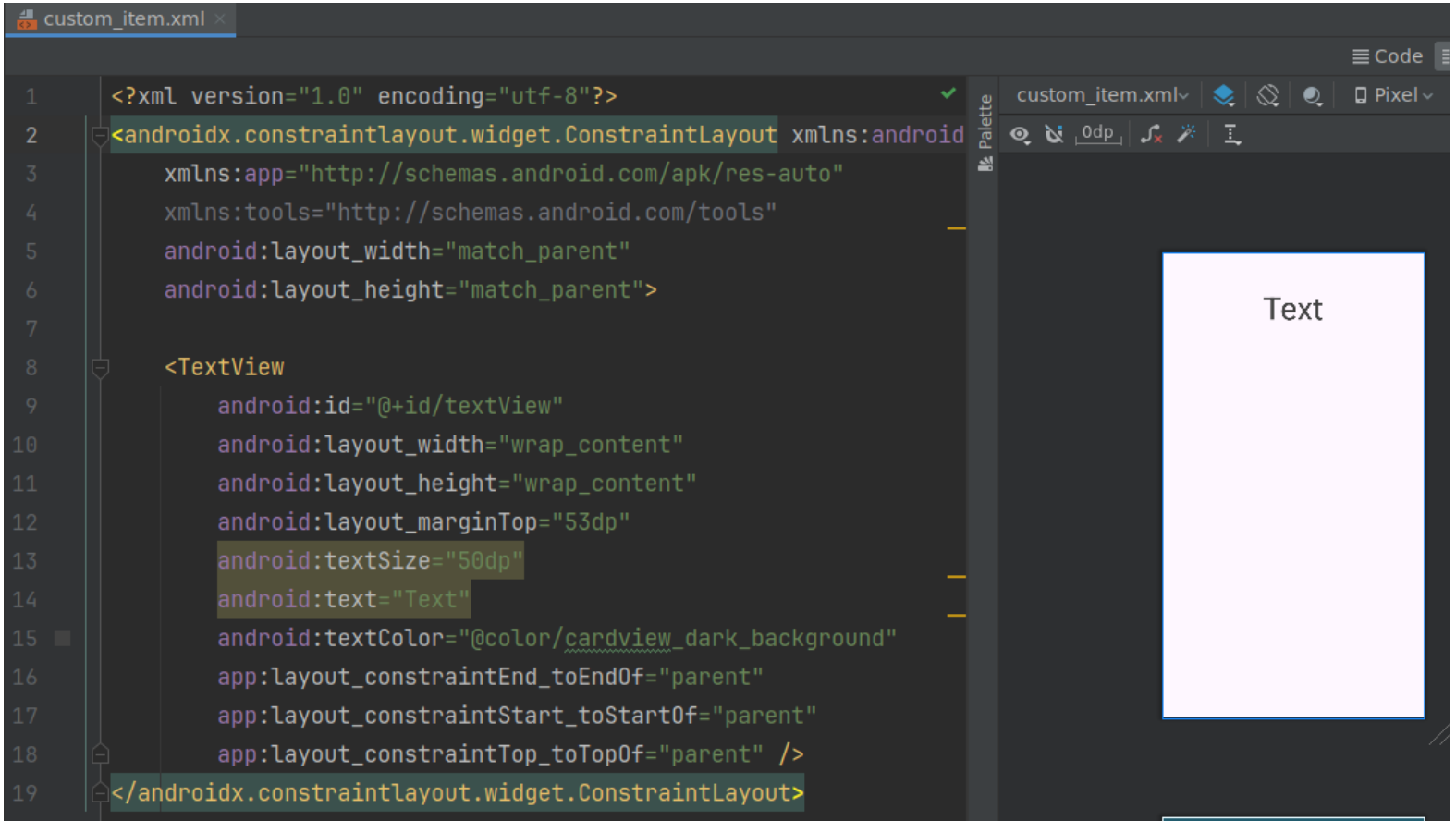
1. android.widget.ListView

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 customisée : `custom_item.xml`



2. android.widget.ListView

Utilisation de la mise en page customisée dans la définition de l'`ArrayAdapter`

```java
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","Tech2CO"};

        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

```java
public class CustomAdapterExample extends AppCompatActivity {

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

        String[] departments = {"Info", "GEA", "Tech2CO"};

        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;
    }
}
```