Afficher une liste

Utilisation du RecyclerView

Pourquoi faire?

Composant qui remplace les ListView / GridView

Plus performant dans la gestion des items

Moins de fonctionnalités disponibles de base

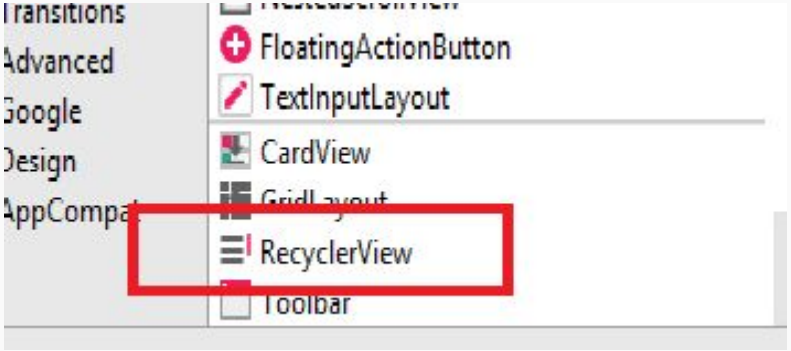
Implémenter un RecyclerView

1. Déclarer un RecyclerView dans le layout de l'activité et le référencer dans le fichier java de l'activité
2. Créer un layout XML personnalisé pour afficher les items dans le RecyclerView
3. Créer une classe ViewHolder pour l'item à afficher, accrocher la source de données au RecyclerView et gérer la logique en créant un RecyclerView.Adapter
4. Attacher l'adapter au RecyclerView / définir le LayoutManager

Implémentation 1/7

Ajouter un RecyclerView au layout de l’activité

Pour cela on utilise l’interface graphique en faisant un drag and drop



Implémentation 2/7

1 - Récupérer une référence dans l'activité concernée:

```
RecyclerView rv = (RecyclerView) findViewById(R.id.rvCities);
```

Implémentation 3/7

Créer un layout pour afficher l’item (Pour nous un TextView et une ImageView)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVie..."
    tools:context="com.estiam.eventcity.CityActivity"
    tools:showIn="@layout/activity_city">

    <ImageView
        android:id="@+id/ivCity"
        android:layout_width="337dp"
        android:layout_height="233dp"
        app:srcCompat="@mipmap/ic_launcher_round"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="8dp"
        android:layout_marginRight="8dp"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginLeft="8dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintHorizontal_bias="0.502" />

    <TextView
        android:id="@+id/tvCity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="TextView"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ivCity"
        app:layout_constraintVertical_bias="0.088" />
</android.support.constraint.ConstraintLayout>
```

Implémentation 5/7

Créer un adapter en héritant de la classe RecyclerView.Adapter et un ViewHolder

```
public class CityAdapter extends RecyclerView.Adapter<CityAdapter.CityHolder>
```

[...]

```
public static class CityHolder extends RecyclerView.ViewHolder {
```

[...]

Implémentation 6/7

Redéfinir les trois méthodes ci-dessous:

```
CityHolder onCreateViewHolder(ViewGroup parent,  int viewType)

void onBindViewHolder(CityHolder holder,  int position)

int getItemCount()
```

Implémentation 7/7

Pour terminer, dans l'actvité, on associe le LayoutManager à utiliser (Linear ou Grid) et l'Adapter qu'on vient de définir:

```
rv.setLayoutManager(new LinearLayoutManager(this));
rv.setAdapter(new CityAdapter(this, cities));
```

Gestion des clicks 1/3

Pour gérer les clicks des utilisateurs, nous allons déclarer une interface afin de pouvoir générer un nouvel évènement:

```
public interface OnCityClickListener {

   void onCityClick(City city);

}
```

Gestion des click 2/3

On ajoute un paramètre au constructeur de CityAdapter:

```
public CityAdapter(Context context, List<City> cities, OnCityClickListener listener) {
```

Puis dans le bind on set le click sur l'itemView du CItyHolder

```
itemView.setOnClickListener(new View.OnClickListener() {
   @Override public void onClick(View v) {
       listener.onCityClick(city);
   }
});
```

Gestion des clicks 3/3

Dans l'activité, on met à jour la déclaration de l'adapter:

```
rv.setAdapter(new CityAdapter(this, cities, new CityAdapter.OnCityClickListener() {
    @Override
    public void onCityClick(City city) {
        Intent intent =  new Intent(getApplicationContext(), CityActivity.  class);
  [..]
        startActivityForResult(intent,  1);
    }
}));
```