CUSTOM ADAPTERS WITH MODEL CLASSES

public class MyAdapter extends ArrayAdapter<Item> {

private static class ViewHolder {

TextView textTitle;

TextView textDescription;

ImageView imageView;

}

public MyAdapter(Context context, List<Item> items) {

super(context, R.layout.list\_item, items);

}

@Override

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

ViewHolder viewHolder;

if (convertView == null) {

// Inflate the item layout and create a new ViewHolder

convertView = LayoutInflater.from(getContext()).inflate(R.layout.list\_item, parent, false);

viewHolder = new ViewHolder();

viewHolder.textTitle = convertView.findViewById(R.id.textTitle);

viewHolder.textDescription = convertView.findViewById(R.id.textDescription);

viewHolder.imageView = convertView.findViewById(R.id.imageView);

convertView.setTag(viewHolder);

} else {

// Reuse the existing ViewHolder

viewHolder = (ViewHolder) convertView.getTag();

}

// Set the data for the current item using the ViewHolder

Item currentItem = getItem(position);

viewHolder.textTitle.setText(currentItem.getTitle());

viewHolder.textDescription.setText(currentItem.getDescription());

viewHolder.imageView.setImageResource(currentItem.getImageResourceId());

return convertView;

}

}

package com.mastercoding.planetsapp;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class MyCustomAdapter extends ArrayAdapter<Planet> {

    // Using Custom Layouts --> MyCustomAdapter

    // Using Custom Objects --> extends ArrayAdapter<Planet>

    private ArrayList<Planet> planetsArrayList;

    Context context;

    public MyCustomAdapter( ArrayList<Planet> planetsArrayList, Context context) {

        super(context, R.layout.item\_list\_layout, planetsArrayList);

        this.planetsArrayList = planetsArrayList;

        this.context = context;

    }

    // View Holder Class: used to cache references to the views within

    //                    an item layout, so that they don't need to be

    //                    repeatedly looked up during scrolling

    private static class MyViewHolder{

        TextView planetName;

        TextView moonCount;

        ImageView planetImg;

    }

    // getView(): used to create and return a view for a

    //            specific item in the list.

    @NonNull

    @Override

    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

        // 1- Get the planet object for the current position

        Planet planets = getItem(position);

        // 2- Inflate Layout:

        MyViewHolder myViewHolder;

        final View result;

        if (convertView == null){

            myViewHolder = new MyViewHolder();

            LayoutInflater inflater = LayoutInflater.from(getContext());

            convertView = inflater.inflate(

                    R.layout.item\_list\_layout,

                    parent,

                    false

            );

         // Finding Views:

         myViewHolder.planetName = (TextView) convertView.findViewById(R.id.planet\_name);

         myViewHolder.moonCount  = (TextView) convertView.findViewById(R.id.moon\_count\_text);

         myViewHolder.planetImg  = (ImageView) convertView.findViewById(R.id.imageView);

         result = convertView;

        convertView.setTag(myViewHolder);

        }else{

            // the view is recycled

            myViewHolder = (MyViewHolder) convertView.getTag();

            result = convertView;

        }

        // Getting the data from model class (Planet)

        myViewHolder.planetName.setText(planets.getPlanetName());

        myViewHolder.moonCount.setText(planets.getMoonCount());

        myViewHolder.planetImg.setImageResource(planets.getPlanetImage());

        return result;

    }

}

MVVM NOTE APP

1. Note.java
2. package com.codinginflow.architectureexample;
3. import android.arch.persistence.room.Entity;
4. import android.arch.persistence.room.PrimaryKey;
5. @Entity(tableName = "note\_table")
6. public class Note {
7. @PrimaryKey(autoGenerate = true)
8. private int id;
9. private String title;
10. private String description;
11. private int priority;
12. public Note(String title, String description, int priority) {
13. this.title = title;
14. this.description = description;
15. this.priority = priority;
16. }
17. public void setId(int id) {
18. this.id = id;
19. }
20. public int getId() {
21. return id;
22. }
23. public String getTitle() {
24. return title;
25. }
26. public String getDescription() {
27. return description;
28. }
29. public int getPriority() {
30. return priority;
31. }
32. }