Simple Android RecyclerView example

Ask Question

I've made a list of items a few times using Android's `RecyclerView`, but it is a rather complicated process. Going through one of the numerous tutorials online works (this, this, and this are good), but I am looking a bare bones example that I can copy and paste to get up and running quickly. Only the following features are necessary:

- Vertical layout
- A single TextView on each row
- Responds to click events

Because I have wished for this several times, I finally decided to make the answer below for my future reference and yours.

android android-recyclerview

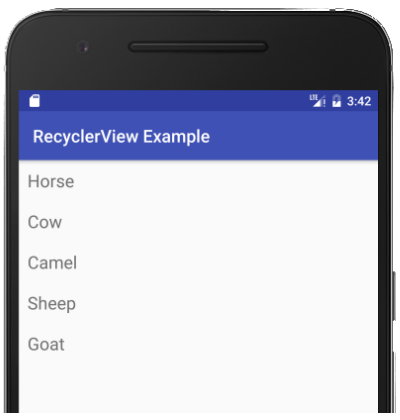
edited Feb 8 '17 at 4:41

asked Nov 14 '16 at 8:19

Suragch
190k 113 656 719

5 Answers

The following is a minimal example that will look like the following image.



Start with an empty activity. You will perform the following tasks to add the RecyclerView. All you need to do is copy and paste the code in each section. Later you can customize it to fit your needs.

- Add dependencies to gradle
- Add the xml layout files for the activity and for the RecyclerView row
- Make the RecyclerView adapter
- Initialize the RecyclerView in your activity

Update Gradle dependencies

Make sure the following dependencies are in your app `gradle.build` file:

```
implementation 'com.android.support:appcompat-v7:27.1.1'
implementation 'com.android.support:recyclerview-v7:27.1.1'
```

You can update the version numbers to whatever is the most current. Use `compile` rather than `implementation` if you are still using Android Studio 2.x.

Create activity layout

Add the `RecyclerView` to your xml layout.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rvAnimals"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

Create row layout

Each row in our `RecyclerView` is only going to have a single `TextView`. Create a new layout resource file.

recyclerview_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

    <TextView
        android:id="@+id/tvAnimalName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"/>

</LinearLayout>
```

Create the adapter

The `RecyclerView` needs an adapter to populate the views in each row with your data. Create a new java file.

MyRecyclerViewAdapter.java

```
public class MyRecyclerViewAdapter extends
RecyclerView.Adapter<MyRecyclerViewAdapter.ViewHolder> {

    private List<String> mData;
    private LayoutInflater mInflater;
    private ItemClickListener mClickListener;

    // data is passed into the constructor
    MyRecyclerViewAdapter(Context context, List<String> data) {
        this.mInflater = LayoutInflater.from(context);
        this.mData = data;
    }

    // inflates the row layout from xml when needed
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = mInflater.inflate(R.layout.recyclerview_row, parent, f
        return new ViewHolder(view);
    }

    // binds the data to the TextView in each row
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        String animal = mData.get(position);
        holder.myTextView.setText(animal);
    }

    // total number of rows
    @Override
    public int getItemCount() {
        return mData.size();
    }


    // stores and recycles views as they are scrolled off screen
    public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {
        TextView myTextView;

        ViewHolder(View itemView) {
            super(itemView);
            myTextView = itemView.findViewById(R.id.tvAnimalName);
            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View view) {
            if (mClickListener != null) mClickListener.onItemClick(view,
getAdapterPosition());
        }
    }

    // convenience method for getting data at click position
```

```
    // allows clicks events to be caught
    void setClickListener(ItemClickListener itemClickListener) {
        this.mClickListener = itemClickListener;
    }

    // parent activity will implement this method to respond to click even
    public interface ItemClickListener {
        void onItemClick(View view, int position);
    }
}
```

Notes

- Although not strictly necessary, I included the functionality for listening for click events on the rows. This was available in the old `ListViews` and is a common need. You can remove this code if you don't need it.

Initialize RecyclerView in Activity

Add the following code to your main activity.

MainActivity.java

```
public class MainActivity extends AppCompatActivity implements
MyRecyclerViewAdapter.ItemClickListener {

    MyRecyclerViewAdapter adapter;

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

        // data to populate the RecyclerView with
        ArrayList<String> animalNames = new ArrayList<>();
        animalNames.add("Horse");
        animalNames.add("Cow");
        animalNames.add("Camel");
        animalNames.add("Sheep");
        animalNames.add("Goat");

        // set up the RecyclerView
        RecyclerView recyclerView = findViewById(R.id.rvAnimals);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        adapter = new MyRecyclerViewAdapter(this, animalNames);
        adapter.setClickListener(this);
        recyclerView.setAdapter(adapter);
    }

    @Override
    public void onItemClick(View view, int position) {
        Toast.makeText(this, "You clicked " + adapter.getItem(position) +
" + position, Toast.LENGTH_SHORT).show();
    }
}
```

Notes

- Notice that the activity implements the `ItemClickListener` that we defined in our adapter. This allows us to handle row click events in `onItemClick`.

Finished

That's it. You should be able to run your project now and get something similar to the image at the top.

Going on

Adding a divider between rows

You can add a simple divider like this

```
DividerItemDecoration dividerItemDecoration = new
DividerItemDecoration(recyclerView.getContext(),
    layoutManager.getOrientation());
recyclerView.addItemDecoration(dividerItemDecoration);
```

If you want something a little more complex, see the following answers:

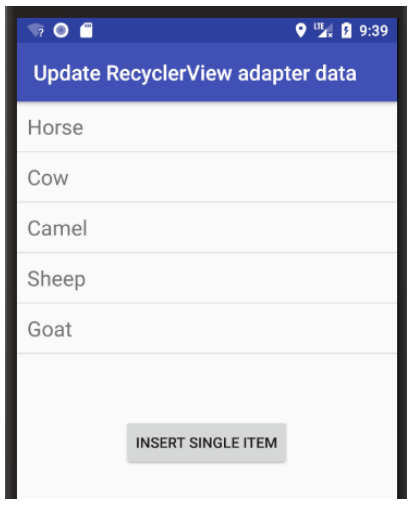
- How to add dividers and spaces between items in RecyclerView?
- How to indent the divider in a linear layout RecyclerView (ie, add padding, margin, or an inset only to the ItemDecoration)

Changing row color on click

See this answer for how to change the background color and add the Ripple Effect when a row is clicked.

Updating rows

See this answer for how to add, remove, and update rows.



Further reading

- CodePath
- YouTube tutorials
- Android RecyclerView Example (stacktips tutorial)
- RecyclerView in Android: Tutorial

edited May 26 at 8:51

answered Nov 14 '16 at 8:19

Suragch
190k 113 656 719

Very nice. THis works until I add a button and try to set its click listener. Any ideas why that might interfere? – john ktejik Oct 8 at 2:23

@johnktejik, it is hard to know exactly what you are talking about from the information you provide here. This would probably best be asked as a new question. My guess is that your button is handling the motion event so that other things don't get it. – Suragch Oct 8 at 8:17

sorry.. to add description of animal? EX: Camel in new TextView: is animal // Sheep in new TextVire : is animal. I need help.. T_T – Montse Mkd Oct 15 at 9:59

Based on different sources I have created Simple Implementation of RecyclerView using a Simple Library.

Add this line in `build.gradle`

```
implementation 'com.hereshem.lib:awesomelib:2.0.1'
```

AddCreate a RecyclerView by adding MyRecyclerView in activity main.xml with

```
        app:layoutManager="LinearLayoutManager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
```

Now in the MainActivity, Create a ViewHolder by passing the name of Class that needs to bind

```
public static class EVHolder extends MyViewHolder<Events> {
    TextView date, title, summary;
    public EVHolder(View v) {
        super(v);
        date = v.findViewById(R.id.date);
        title = v.findViewById(R.id.title);
        summary = v.findViewById(R.id.summary);
    }
    @Override
    public void bindView(Events c) {
        date.setText(c.date);
        title.setText(c.title);
        summary.setText(c.summary);
    }
}
```

Create Items list variable and adapters with very few lines by passing items, class and layout in the adapter

```
List<Events> items = new ArrayList<>();
MyRecyclerView recycler = findViewById(R.id.recycler);
RecyclerViewAdapter adapter = new RecyclerViewAdapter(this, items, EVHolde
R.layout.row_event);
recycler.setAdapter(adapter);
```

ClickListener can be added with following lines

```
recycler.setOnItemClickListener(new MyRecyclerView.OnItemClickListener() {
    @Override
    public void onItemClick(int position) {
        Toast.makeText(MainActivity.this, "Recycler Item Clicked " + posit
Toast.LENGTH_SHORT).show();
    }
});
```

Its all done.

More example and implementation can be found here . Hope this helps !!!

edited Oct 9 at 4:37

answered Jun 15 at 3:58

Hem Shrestha
396 6 14

what is single activity – john ktejik Oct 8 at 0:30

: error: cannot find symbol summary = v.findViewById(R.id.summary); ^ symbol: variable summary location: class id EVHolder.java:15: error: method does not override or implement a method from a supertype @Override ^ EVHolder.java:17: error: cannot find symbol date.setText(c.date); – john ktejik Oct 8 at 0:40

@johnktejik R.id.summary is the id given for layout resource file. please see this layout file github.com/hereshem/Easy-RecyclerView-Library/blob/master/app/… – Hem Shrestha Oct 9 at 2:57

Dependencies

```
compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support:design:25.3.1'
    compile 'com.android.support:multidex:1.0.1'
    compile 'com.android.support:cardview-v7:25.3.1'
    compile 'com.android.support:support-v4:25.3.1'
    compile 'com.lguipeng.bubbleview:library:1.0.0'
    compile 'com.larswerkman:HoloColorPicker:1.5'
    compile 'com.mcxiaoke.volley:library-aar:1.0.0'
```

One Class For Click Item

```
 import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;

public class RecyclerItemClickListener implements RecyclerView.OnItemTouch
```

```
        public void onItemClick(View view, int position);
    }

    GestureDetector mGestureDetector;

    public RecyclerItemClickListener(Context context, OnItemClickListener
        mListener = listener;
        mGestureDetector = new GestureDetector(context, new
GestureDetector.SimpleOnGestureListener() {
            @Override
            public boolean onSingleTapUp(MotionEvent e) {
                return true;
            }
        });
    }

    @Override
    public boolean onInterceptTouchEvent(RecyclerView view, MotionEvent e)
        View childView = view.findChildViewUnder(e.getX(), e.getY());
        if (childView != null && mListener != null && mGestureDetector.onT
            mListener.onItemClick(childView, view.getChildPosition(childVi
            return true;
        }
        return false;
    }

    @Override
    public void onTouchEvent(RecyclerView view, MotionEvent motionEvent) {

    @Override
    public void onRequestDisallowInterceptTouchEvent(boolean disallowInter

    }
```

```
}
```

Second Class RecyclerView

```
import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

public class SLByTopics extends Fragment {


    public static ArrayList<MByTopics> byTopicsMainArrayList=new ArrayList


    TabRefreshReceiver tabRefreshReceiver;
    RecyclerView recyclerView;
    SAdpByTopics sAdpByTopics;
    public ArrayList<MByTopics> mByTopicsArrayList=new ArrayList<>();
    ProgressDialog progressDialog;

    public SLByTopics(){
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
@Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.sl_fragment_by_topics, conta
```

```
        } else{
            IsOnline.showNoInterNetMessage(getActivity());
        }
        tabRefreshReceiver = new TabRefreshReceiver();

LocalBroadcastManager.getInstance(getContext()).registerReceiver(tabRefres
IntentFilter("BY_TOPICS"));

        setUpView(view);
        return view;
    }

    private void setUpView(View view) {

        recyclerView=(RecyclerView)view.findViewById(R.id.by_topics_list_r
        LinearLayoutManager linearLayoutManager=new LinearLayoutManager(ge
        linearLayoutManager.setOrientation(LinearLayoutManager.VERTICAL);
        recyclerView.setLayoutManager(linearLayoutManager);
    }

    @Override
    public void onResume() {
        super.onResume();

        recyclerView.addOnItemTouchListener(new RecyclerItemClickListener(
new RecyclerItemClickListener.OnItemClickListener() {
            @Override
            public void onItemClick(View view, final int position) {

                if (mByTopicsArrayList.get(position).getChild().size()>0){
                    Intent intent = new Intent(getActivity(), SByTopicCate

intent.putExtra("selectedCategoryName",mByTopicsArrayList.get(position).ge

intent.putExtra("jsonData",mByTopicsArrayList.get(position).getMainTopicJs
                    startActivity(intent);
                    getActivity().overridePendingTransition(R.anim.activit
R.anim.activity_out);
                }else {
                    Intent intent = new Intent(getActivity(),
SByCategoryQuestionList.class);

intent.putExtra("selectedSubCategoryName",mByTopicsArrayList.get(position)
                    intent.putExtra("catID",mByTopicsArrayList.get(positio
                    startActivity(intent);
                    getActivity().overridePendingTransition(R.anim.activit
R.anim.activity_out);
                }
            }
        }));

    }

    private class TabRefreshReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            try {
                FragmentTransaction ft = getFragmentManager().beginTransac
                ft.detach(SLByTopics.this).attach(SLByTopics.this).commit(

LocalBroadcastManager.getInstance(getContext()).unregisterReceiver(tabRefr
            } catch (Exception e) {
                e.printStackTrace();
            }

        }
    }
```



```
    private void getCategoryTree() {
        progressDialog.setMessage("Please Wait...");
        progressDialog.setCancelable(false);
        progressDialog.show();

        StringRequest stringRequest = new StringRequest(Request.Method.POS
Const.HOSTNAME + Const.STUDENT_GET_CATEGORY_TREE,
                new Response.Listener<String>() {
                    @SuppressLint("LongLogTag")
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject object = new JSONObject(response);
                            String status = object.getString("status");
                            int i = Integer.parseInt(status);

                            switch (i) {

                                case 0:
                                    progressDialog.dismiss();
//                                    Toast.makeText(getActivity(), "getCa
Toast.LENGTH_SHORT).show();
```

```
                try {


                    byTopicsMainArrayList.clear();
                    JSONArray info = object.getJSONArr
                    if (info.length() > 0) {
                        for (i = 0; i < info.length();
                            JSONObject data = info.get
                            MByTopics mByTopics = new
                            mByTopics.setId(data.getSt

mByTopics.setCatname(data.getString("catname"));

mByTopics.setMainTopicJson(data.toString());

                            JSONArray topicChildren =

data.getJSONArray("children");
                            ArrayList<SMByTopicCategor

byChildrenArrayList = new ArrayList<>();


                            for (int j = 0; j <

topicChildren.length(); j++) {
                                JSONObject topicChildr

topicChildren.getJSONObject(j);
                                SMByTopicCategory smBy

new SMByTopicCategory();

smByTopicCategory.setId(topicChildrenData.getString("id"));

smByTopicCategory.setCatname(topicChildrenData.getString("catname"));

smByTopicCategory.setChildTopicJson(topicChildrenData.toString());


                                JSONArray topicChildre

topicChildrenData.getJSONArray("children");
                                ArrayList<SMByTopicSub

byChildrenSubArrayList = new ArrayList<>();


                                for (int k = 0; k <

topicChildrenQuestion.length(); k++) {
                                    JSONObject topicCh

topicChildrenQuestion.getJSONObject(k);
                                    SMByTopicSubCatego

smByTopicSubCategory = new SMByTopicSubCategory();

smByTopicSubCategory.setId(topicChildrenSubData.getString("id"));

smByTopicSubCategory.setCatname(topicChildrenSubData.getString("catname"))

smByTopicSubCategory.setChildSubTopicJson(topicChildrenSubData.toString())


byChildrenSubArrayList.add(smByTopicSubCategory);
                                }


smByTopicCategory.setQuestions(byChildrenSubArrayList);


byChildrenArrayList.add(smByTopicCategory);
                            }
                            mByTopics.setChild(byChild
                            byTopicsMainArrayList.add(
                        }


                        mByTopicsArrayList.clear();
                        mByTopicsArrayList=byTopicsMai
                        sAdpByTopics=new
SAdpByTopics(mByTopicsArrayList,getActivity());
                        recyclerView.setAdapter(sAdpBy
                        sAdpByTopics.notifyDataSetChan

                    }

                }catch (Exception e){
                    e.printStackTrace();
                }
                break;

            default:
                progressDialog.dismiss();
//                  Toast.makeText(getActivity(), "getCa
+ response, Toast.LENGTH_SHORT).show();
                Log.e("getCategoryTree Not Response",
"getCategoryTree Uploading Not Response : " + response);
            }

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
```

```
//                        Toast.makeText(getActivity(), error.getMessage()
Toast.LENGTH_LONG).show();
                    }
                }){

        };/* {
            @Override
            protected Map<String, String> getParams() throws AuthFailureEr

                Map<String, String> map = new HashMap<String, String>();
//                map.put("uid", String.valueOf(ConfigManager.getUserId()),
                return map;
            }
        };*/

        stringRequest.setRetryPolicy(new DefaultRetryPolicy(
                0,
                DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
                DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));

        RequestQueue requestQueue = Volley.newRequestQueue(getActivity());
        requestQueue.add(stringRequest);
    }
}
```

Adapter Class For Recycler Item

```
import android.app.Activity;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.util.ArrayList;

public class SAdpByTopics extends RecyclerView.Adapter<RecyclerView.ViewHo
        ArrayList<MByTopics> topicsArrayList=new ArrayList<>();
        Activity activity;

 public SAdpByTopics(ArrayList<MByTopics> topicsArrayList,Activity activit
        this.topicsArrayList=topicsArrayList;
        this.activity=activity;
        }

@Override
public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int vi
        View itemeView=
LayoutInflater.from(parent.getContext()).inflate(R.layout.list_item_by_top

        RecyclerView.ViewHolder holder=new Holder(itemeView);
        holder.setIsRecyclable(false);
        return holder;
        }

@Override
public void onBindViewHolder(RecyclerView.ViewHolder holder, int position)
final Holder classHolder = (Holder) holder;
        try{
        classHolder.txt_topic_name.setText(topicsArrayList.get(position).g
        }catch (Exception e){
        e.printStackTrace();
        }
        }

@Override
public int getItemCount() {
        return topicsArrayList.size();
        }


class Holder extends RecyclerView.ViewHolder implements View.OnClickListen
    TextView txt_topic_name;

    public Holder(View itemView) {
        super(itemView);
        txt_topic_name = (TextView) itemView.findViewById(R.id.txt_topic_n
    }

    @Override
    public void onClick(View v) {

    }
}
}
```

Module Class

```
public class MByTopics {

    String id;
    String topicName;
    String catname;
    String MainTopicJson;
    ArrayList<SMByTopicCategory> child;
    ArrayList<SMByTopicSubCategory> questions;
```

```
public void setId(String id){
    this.id=id;
}
public String getId(){
    return  id;
}

public void setCatname(String catname) {
    this.catname = catname;
}

public String getCatname() {
    return catname;
}

public void setTopicName(String topicName) {
    this.topicName = topicName;
}
public String getTopicName() {
    return topicName;
}

public void setChild(ArrayList<SMByTopicCategory> child) {
    this.child = child;
}


public String getMainTopicJson() {
    return MainTopicJson;
}

public void setMainTopicJson(String mainTopicJson) {
    MainTopicJson = mainTopicJson;
}


public ArrayList<SMByTopicCategory> getChild() {
    return child;
}

public void setQuestions(ArrayList<SMByTopicSubCategory> questions) {
    this.questions = questions;
}

public ArrayList<SMByTopicSubCategory> getQuestions() {
    return questions;
}

public ArrayList<MByTopics> getByTopicList() {
    ArrayList<MByTopics> mByTopicsArrayList = new ArrayList<>();

    for (int i=0;i<11;i++){
        MByTopics mQuestionBankCategory=new MByTopics();

        if (i==1 || i== 5|| i==9){
            mQuestionBankCategory.setTopicName("Microeconomics");
        }else  if (i==2 || i== 10|| i==6) {
            mQuestionBankCategory.setTopicName("Macroeconomics");
        }else {
            mQuestionBankCategory.setTopicName("Current Isssues");
        }

        mByTopicsArrayList.add(mQuestionBankCategory);
    }

    return mByTopicsArrayList;
}
```

answered Jan 4 at 12:32

Ranjit Barve
11 1

2 explanations would be nice – ThomasEdwin Jan 4 at 12:38

Since I cant comment yet im gonna post as an answer the link.. I have found a simple, well organized tutorial on recyclerview http://www.androiddeft.com/2017/10/01/recyclerview-android/

Apart from that when you are going to add a recycler view into you activity what you want to do is as below and how you should do this has been described on the link

- add RecyclerView component into your layout file
- make a class which you are going to display as list rows
- make a layout file which is the layout of a row of you list

- add recyclerview into your mainActivity oncreate
- adding separators
- adding Touch listeners

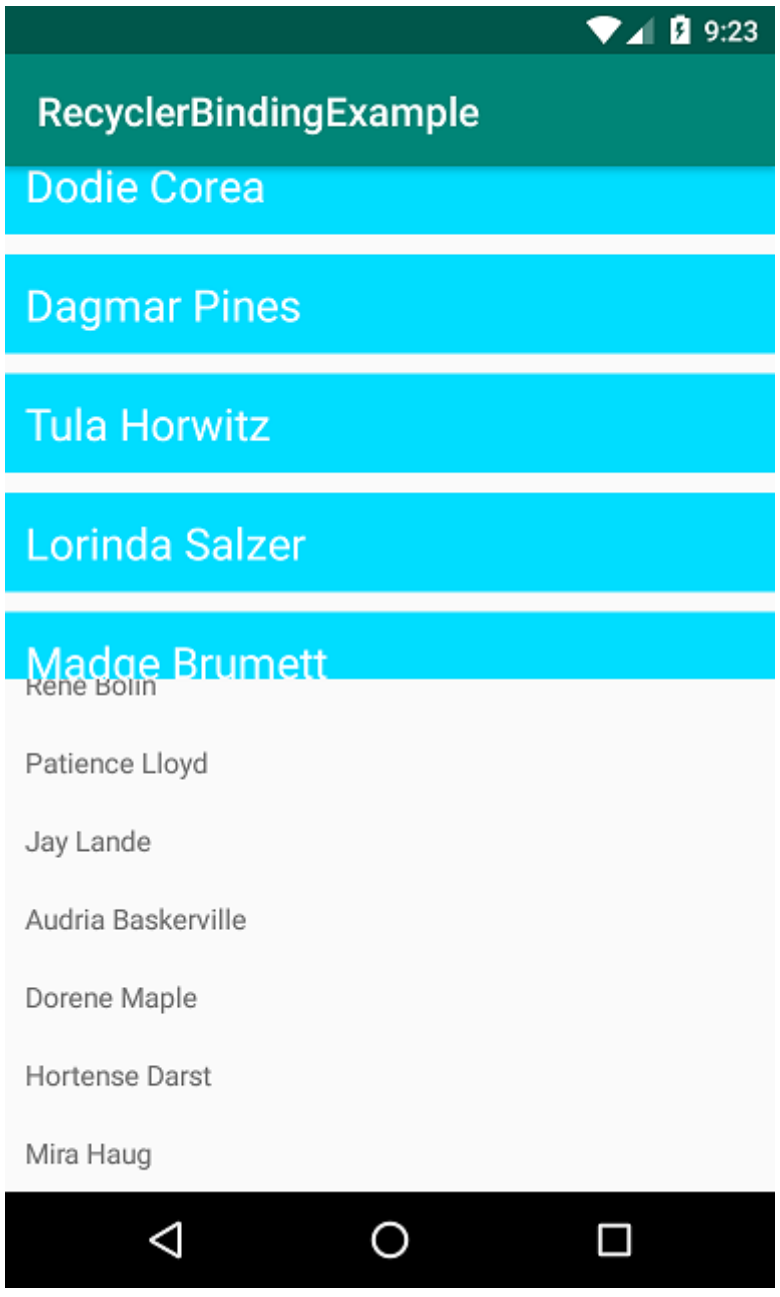
edited May 26 at 9:08

answered May 26 at 8:44

NuOne T Attygalle
612 4 13

Now you need [1 adapter] for all `RecyclerView`

- One adapter can be used in for all RecyclerView. So **NO** `onBindViewHolder` , **No** `onCreateViewHolder` handling.
- No code for setting adapter from Java/Kotlin class. Check [sample class].
- You can set events and custom data for every list by using [Binding Adapters].



I show here setting two different `RecyclerView` by 1 adapter -

activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <data>

        <variable
            name="listOne"
            type="java.util.List"/>

        <variable
            name="listTwo"
            type="java.util.List"/>

        <variable
            name="onItemClickListenerOne"
            type="com.ks.nestedrecyclerbindingexample.callbacks.OnItemClic
```

```
    </data>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <android.support.v7.widget.RecyclerView
            rvItemLayout="@{@layout/row_one}"
            rvList="@{listOne}"
            rvOnItemClick="@{onItemClickListenerOne}"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:layoutManager="android.support.v7.widget.LinearLayoutManag
            />

        <android.support.v7.widget.RecyclerView
            rvItemLayout="@{@layout/row_two}"
            rvList="@{listTwo}"
            rvOnItemClick="@{onItemClickListenerTwo}"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:layoutManager="android.support.v7.widget.LinearLayoutManag
            />

    </LinearLayout>

</layout>
```

You can see I pass list, item layout id and click listener from layout.

```
rvItemLayout="@{@layout/row_one}"
rvList="@{listOne}"
rvOnItemClick="@{onItemClickListenerOne}"
```

This custom attributes are created by **BindingAdapter**.

```
public class BindingAdapters {
    @BindingAdapter(value = {"rvItemLayout", "rvList", "rvOnItemClick"}, r
false)
    public static void setRvAdapter(RecyclerView recyclerView, int rvItemL
rvList, @Nullable OnItemClickListener onItemClickListener) {
        if (rvItemLayout != 0 && rvList != null && rvList.size() > 0)
            recyclerView.setAdapter(new GeneralAdapter(rvItemLayout, rvLis
onItemClickListener));
    }
}
```

Now from Activity, you pass list, click listener like

HomeActivity.java

```
public class HomeActivity extends AppCompatActivity {
    ActivityHomeBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView(this, R.layout.activity_h
        binding.setListOne(new ArrayList()); // pass your list or set list
of API
        binding.setListTwo(new ArrayList());
        binding.setOnItemClickListenerOne(new OnItemClickListener() {
            @Override
            public void onItemClick(View view, Object object) {
                if (object instanceof ModelParent) {
                    // TODO: your action here
                }
            }
        });
        binding.setOnItemClickListenerTwo(new OnItemClickListener() {
            @Override
            public void onItemClick(View view, Object object) {
                if (object instanceof ModelChild) {
                    // TODO: your action here
                }
            }
        });
    }
}
```

You don't want read too much, directly clone/download full example on from my github repo. And try it yourself.

You can see `GeneralAdapter.java` in above repo.

If you have problems while setting up data binding, please see this answer.

edited Oct 23 at 9:32

Khemraj
10.3k 2 24 62