**PRACTICAL**

**OBJECT:**

Displaying Lists with RecyclerView

**THEORY:**

A RecyclerView is used to display lists.

Create a new project with empty activity template. Check the backward compatibility to include appcompat support. You will need to perform the following tasks to add the RecyclerView:

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

**Update Gradle dependencies**

To use recyclerview, you have to added the dependencies in your gradle build. Make sure the following dependencies are in your app gradle.build file:

Listing 1: Adding gradle dependencies

*implementation* ***'com.android.support:appcompat-v7:28.0.0-alpha1'***

*implementation* ***'com.android.support:recyclerview-v7:28.0.0-alpha1'***

You can update the version numbers to whatever is the most current. In case you use drag and drop to put the recyclerview on your activity layout, you do not need to add the gradle dependencies manually. The latest version will be added automatically.

**Create activity layout**

Add the RecyclerView to your xml layout. You can do it either by writing the xml below, or you can just drag and drop it on the design tab as shown in figure 1 (in which case gradle dependency is added automatically).

Listing 2: Adding RecyclerView in activity\_main.xml

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

<**RelativeLayout 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"**

**tools:context=".MainActivity"**

**tools:layout\_marginLeft="22dp"**>

<**android.support.v7.widget.RecyclerView**

**android:id="@+id/rvManufacturer"**

**android:layout\_width="355dp"**

**android:layout\_height="495dp"**

**android:layout\_marginStart="16dp"**

**android:layout\_marginTop="16dp"**

**android:layout\_marginEnd="32dp"**

**android:layout\_marginBottom="16dp"**

**android:background="@android:color/holo\_blue\_light"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="1.0"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"**

**app:layout\_constraintVertical\_bias="0.0"**

**tools:spanCount="5"** />

</**RelativeLayout**>

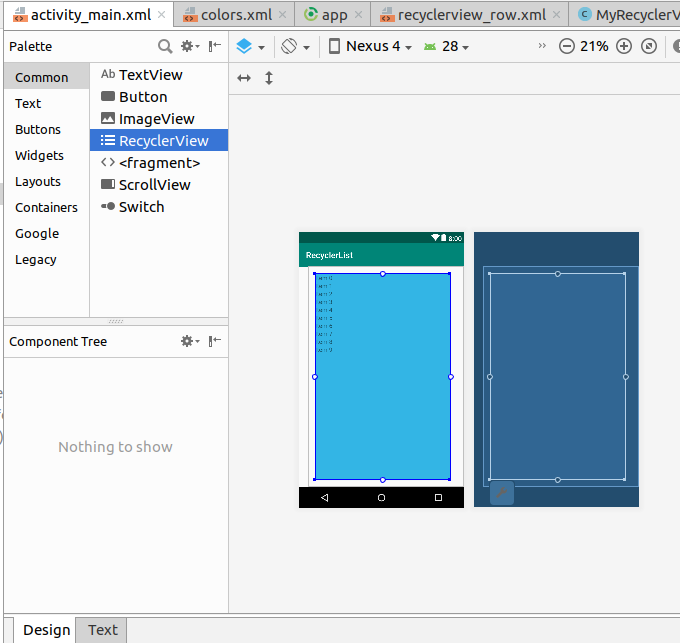


Figure 1: Adding RecyclerView to activity\_main.xml

**Create row layout**

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

Listing 3: recyclerview\_row.xml includes TextView

*<?***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"**>

<**TextView**

**android:id="@+id/*tvManufacturerName*"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:textSize="20dp"**

**android:text="TextView"** />

</**LinearLayout**>

**Create the adapter**

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

Listing 4: MyRecyclerViewAdapter class

**import** android.content.Context;

**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.List;

**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, **false**);

**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.***tvManufacturerName***);

itemView.setOnClickListener(**this**);

}

@Override

**public void** onClick(View view) {

**if** (**mClickListener** != **null**) **mClickListener**.onItemClick(view, getAdapterPosition());

}

}

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

String getItem(**int** id) {

**return mData**.get(id);

}

*// allows clicks events to be caught*

**void** setClickListener(ItemClickListener itemClickListener) {

**this**.**mClickListener** = itemClickListener;

}

*// parent activity will implement this method to respond to click events*

**public interface** ItemClickListener {

**void** onItemClick(View view, **int** position);

}

}

**Initialize RecyclerView in Activity**

Add the following code to your main activity.

Listing 4: MainActivity.java overrides the click listener

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.support.v7.widget.LinearLayoutManager;

**import** android.support.v7.widget.RecyclerView;

**import** android.view.View;

**import** android.widget.Toast;

**import** java.util.ArrayList;

**public class** MainActivity **extends** AppCompatActivity

**implements** MyRecyclerViewAdapter.ItemClickListener {

**private** MyRecyclerViewAdapter **adapter**;

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

ArrayList<String> manufacturerNames = **new** ArrayList<>();

manufacturerNames.add(**"Apple"**);

manufacturerNames.add(**"Samsung"**);

manufacturerNames.add(**"LG"**);

manufacturerNames.add(**"Q Mobile"**);

*// set up the RecyclerView*

RecyclerView recyclerView = findViewById(R.id.***rvManufacturer***);

recyclerView.setLayoutManager(**new** LinearLayoutManager(**this**));

**adapter** = **new** MyRecyclerViewAdapter(**this**, manufacturerNames);

**adapter**.setClickListener(**this**);

recyclerView.setAdapter(**adapter**);

}

@Override

**public void** onItemClick(View view, **int** position) {

Toast.*makeText*(**this**, **"You clicked "** + **adapter**.getItem(position), Toast.***LENGTH\_SHORT***).show();

}

}

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

**ACTIVITIES**

**Activity 1**

Update the above application and add an image for each animal in the list. Further, in onClick event open a new activity to show details.

**REVIEW QUESTIONS**

1. What is a RecyclerView?
2. What is the difference between Android ListView and RecyclerView?
3. Which dependency is required for RecyclerView?
4. How does RecyclerView perform its operation?