

Andorid Programming

Week 4

Example Code for RecyclerView

Mina Jung

EECS, Syracuse University

Spring 2017

Outline I

Layout File of RecyclerView

Setup in a Fragment for RecyclerView

Create Adapter

Create ViewHolder in your Adapter

Setup the View for Item Using ViewHolder

Fill in Data

CardView Layout File of Item View

Event Handler in RecyclerView

Items with Different Layout with their Type

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/recycler_view2"
        android:scrollbars="vertical">
    </android.support.v7.widget.RecyclerView>

</LinearLayout>
```

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
    container,
        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View currentView = inflater.inflate(R.layout.fragment_rv,
        container, false);

    recyclerView = (RecyclerView)
        currentView.findViewById(R.id.recycler_view2);
    recyclerView.setHasFixedSize(true);

    layoutManager = new
        LinearLayoutManager(currentView.getContext());
    // Set the LayoutManager to RecyclerView
    recyclerView.setLayoutManager(layoutManager);
    myAdapter = new MyRVAdapter();
    recyclerView.setAdapter(myAdapter);
    recyclerView.setItemAnimator(new SlideInLeftAnimator());

    return currentView;
}
```

```
public class MyRVAdapter extends
    RecyclerView.Adapter<MyRVAdapter.ViewHolder> {
    private MovieData mItems;
    private int lastPosition = -1;
    static OnItemClickListener mItemClickListener;

    public MyRVAdapter() {
        mItems = new MovieData();
    }

    ...
}
```

```
public static class ViewHolder extends RecyclerView.ViewHolder {  
  
    public ImageView movieImage;  
    public TextView movieName;  
    public TextView movieDesc;  
    public CheckBox movieCheck;  
  
    public ViewHolder(View view) {  
        super(view);  
        movieImage = (ImageView)  
            view.findViewById(R.id.movieImage);  
        movieName = (TextView)  
            view.findViewById(R.id.movieName);  
        movieDesc = (TextView)  
            view.findViewById(R.id.movieDesc);  
        movieCheck = (CheckBox)  
            view.findViewById(R.id.movieCheck);  
  
        ....  
    }  
}
```

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int
    viewType) {
    // create a new view
    View v;

    v = LayoutInflater.from( parent.getContext()).inflate(
        R.layout.rv_item, parent, false);

    return new ViewHolder(v);
}
```

```
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    HashMap<String,?> movie = mItems.getItem(position);
    holder.movieImage.setImageResource((Integer)
        movie.get("image"));
    holder.movieName.setText((String)movie.get("name"));
    holder.movieDesc.setText((String)movie.get("description"));
    holder.movieCheck.setChecked(false);
    setAnimation(holder.movieImage, position);
}

    // must Generate
@Override
public int getItemCount() {
    return mItems.getSize();
}

@Override
public int getItemViewType(int position){
    return position % 3;
}
```



```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.v7.widget.CardView
        app:cardCornerRadius="10dp"
        app:cardElevation="10dp"
        app:cardUseCompatPadding="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

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

            <ImageView
                android:id="@+id/movieImage"
                android:layout_weight="1"
                android:layout_width="100dp"
                android:layout_height="100dp" />

            <LinearLayout
```

```
        android:orientation="vertical"
        android:layout_weight="3"
        android:layout_marginLeft="3dp"
        android:layout_width="wrap_content"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/movieName"
            android:textSize="16dp"
            android:textColor="#000000"
            android:gravity="center_vertical"
            android:layout_weight="1" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/movieDesc"
            android:textSize="12dp"
            android:textColor="#666666"
            android:layout_weight="2" />

    </LinearLayout>
    <CheckBox
        android:id="@+id/movieCheck"
```

```
        android:layout_weight="1"
        android:layout_width="60dp"
        android:layout_height="match_parent" />
    </LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
```

1. Set Event Listener on the ItemView (inside ViewHolder)

```
public static class ViewHolder extends RecyclerView.ViewHolder {  
  
    ....  
    public ViewHolder(View view) {  
        super(view);  
        ....  
        view.setOnClickListener(new View.OnClickListener(){  
  
            @Override  
            public void onClick(View v) {  
                if(mItemClickListener != null){  
                    if(getAdapterPosition() !=  
                        RecyclerView.NO_POSITION) {  
                        mItemClickListener.onItemClick(v,  
                            getAdapterPosition());  
                    }  
                }  
            }  
        });  
  
        view.setOnLongClickListener(new  
            View.OnLongClickListener(){
```

```
    @Override
    public boolean onLongClick(View v) {
        if (mItemClickListener != null) {
            if (getAdapterPosition() !=
                RecyclerView.NO_POSITION) {
                mItemClickListener.onItemLongClick(v,
                    getAdapterPosition());
            }
        }
        return true;
    }
});
....
```

2. Receive the external event handler (in Adapter)

2.1 Interface

```
public interface OnItemClickListener{  
    public void onItemClick(View view, int position);  
    public void onItemLongClick(View view, int position);  
}
```

2.2 Allow app to set the event listeners

```
public void setOnItemClickListener(final OnItemClickListener  
    mItemClickListener){  
    this.mItemClickListener = mItemClickListener;  
}
```

3. Add Click Listener to RecyclerView.Adapter (in the Fragment)

```
myAdapter.setOnItemClickListener(new  
    MyRVAdapter.OnItemClickListener(){  
        @Override  
        public void onItemClick(View view, int position) {  
            Toast.makeText(getContext(), "Move "+  
                (position+1)+"  
                Selected",Toast.LENGTH_SHORT).show();  
        }  
  
        @Override  
        public void onItemLongClick(View view, int position) {  
            myAdapter.removeItem(position);  
            myAdapter.notifyItemRemoved(position);  
        }  
    });
```

- when data is changed, the list is also updated (check LongClick handler)

4. Check the Event Handlers on Item View again (External Event Handler is now connected to the inside Event Handler)

```
view.setOnClickListener(new View.OnClickListener(){  
  
    @Override  
    public void onClick(View v) {  
        if(mItemClickListener != null){  
            if(getAdapterPosition() !=  
                RecyclerView.NO_POSITION) {  
                mItemClickListener.onItemClick(v,  
                    getAdapterPosition());  
            }  
        }  
    }  
});
```


1. Get the type for each Item (in the Adapter)

```
@Override
public int getItemViewType(int position){
    return position % 3;
}
```

2. Inflate corresponding Layouts on View Creation

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int
    viewType) {
    // create a new view
    View v;
    switch(viewType){
        case 2:
            v =
                LayoutInflater.from(parent.getContext()).inflate(R
                    parent, false);
            break;
        case 1:
            v =
                LayoutInflater.from(parent.getContext()).inflate(R
                    parent, false);
```

```
        break;
    case 0:
    default:
        v =
            LayoutInflater.from(parent.getContext()).inflate(R
            parent, false);
        break;
    }

    return new ViewHolder(v);
}
```