

# RecyclerView

<http://www.androidhive.info/2016/01/android-working-with-recycler-view/>

<http://antonioleiva.com/recyclerview/>



# New Project

Android Studio

## Configure your new project

Application name:

Company Domain:

Package name:

at.htl.movies

[Edit](#)

Project location:



Cancel

Previous

Next

Finish



## Target Android Devices

### Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.

By targeting API 18 and later, your app will run on approximately **74.3%** of the devices that are active on the Google Play Store.

[Help me choose](#)

Wear

Minimum SDK

TV

Minimum SDK

Android Auto

Glass

Minimum SDK

Cancel

Previous

Next

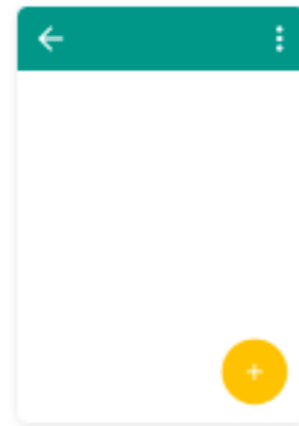
Finish



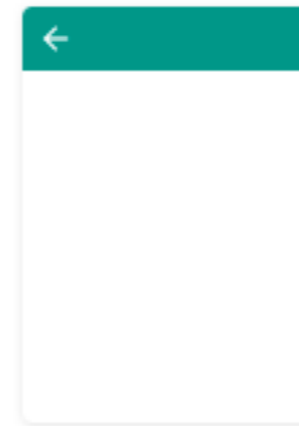
# Add an Activity to Mobile



Add No Activity



Basic Activity



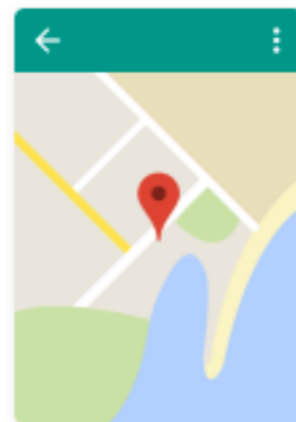
Empty Activity



Fullscreen Activity



Google AdMob Ads Activity



Google Maps Activity



Login Activity



Master/Detail Flow

Cancel

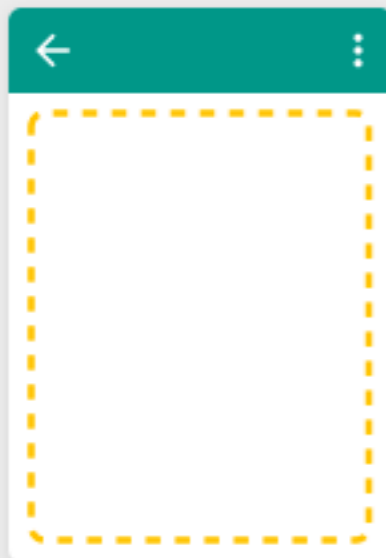
Previous

Next

Finish



## Customize the Activity



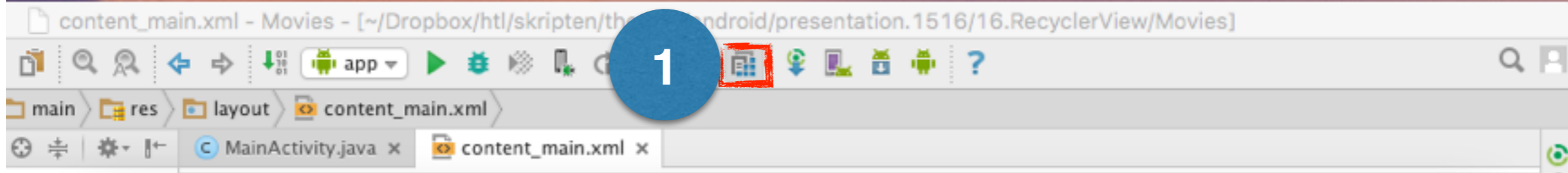
Basic Activity

Creates a new basic activity with an app bar.

Activity Name:	<input type="text" value="MainActivity"/>
Layout Name:	<input type="text" value="activity_main"/>
Title:	<input type="text" value="MainActivity"/>
Menu Resource Name:	<input type="text" value="menu_main"/>
	<input checked="" type="checkbox"/> Use a Fragment

---

If true, the content will be a fragment



Project Structure

Properties | Signing | Flavors | Build | **Dependencies**

	Scope
{include=[*.jar], dir=libs}	Compile
m junit:junit:4.12	Test compile
m com.android.support:appcompat-v7:23.3.0	Compile
m com.android.support:design:23.3.0	Compile

Choose Library Dependency

com.android.support:recyclerview-v7:23.3.0

Enter terms for Maven Central search, or fully-qualified coordinates (e.g. com.google.code.gson:gson:2.2.4)

- play-services-ads (com.google.android.gms:play-services-ads:8.4.0)
- play-services-wearable (com.google.android.gms:play-services-wearable:8.4.0)
- play-services-maps (com.google.android.gms:play-services-maps:8.4.0)
- cardview-v7 (com.android.support:cardview-v7:23.3.0)
- palette-v7 (com.android.support:palette-v7:23.3.0)
- recyclerview-v7 (com.android.support:recyclerview-v7:23.3.0)**
- leanback-v17 (com.android.support:leanback-v17:23.3.0)
- wearable (com.google.android.support:wearable:1.4.0)

OK

OK

2

5

4

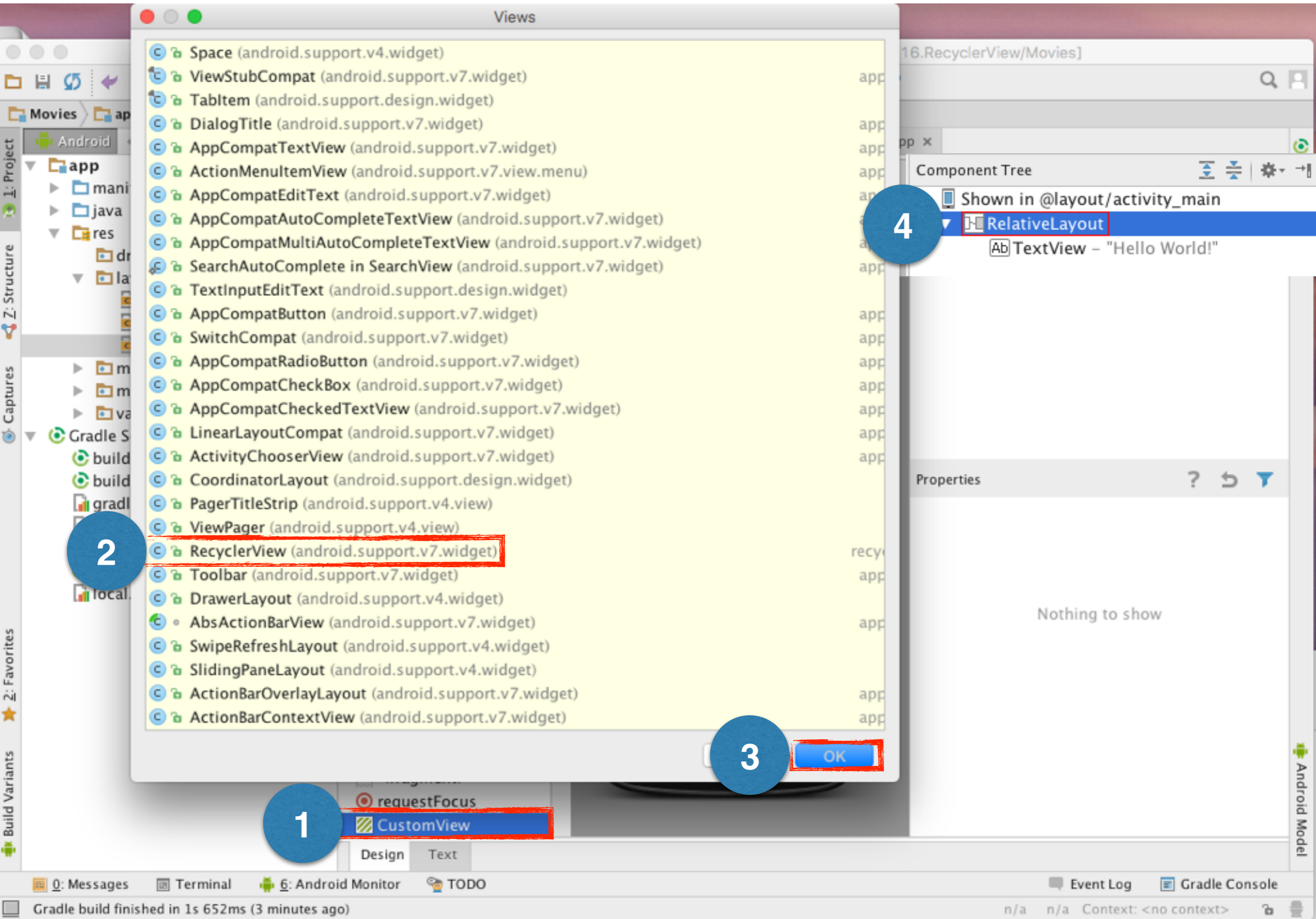
3

6

7

# Eintrag in build.gradle

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.3.0'  
    compile 'com.android.support:design:23.3.0'  
    compile 'com.android.support:recyclerview-v7:23.3.0'  
}
```





# fragment\_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/  
android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:paddingBottom="@dimen/activity_vertical_margin"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"  
    tools:context="at.htl.movies.MainActivityFragment"  
    tools:showIn="@layout/activity_main">
```

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

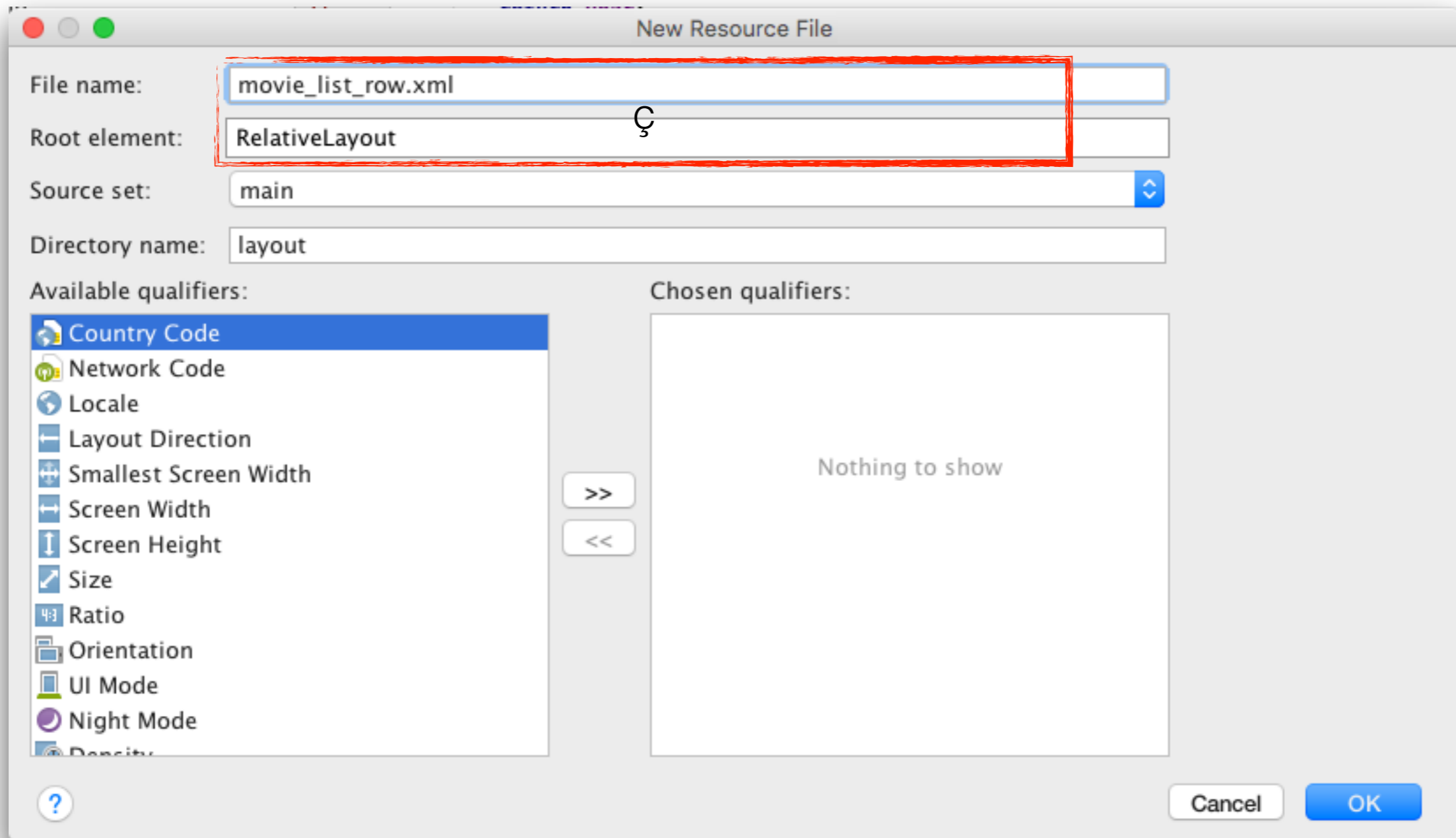
```
</RelativeLayout>
```

Text mit copy-  
paste kopierbar

# colors.xml

```
colors.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#3F51B5</color>
4   <color name="colorPrimaryDark">#303F9F</color>
5   <color name="colorAccent">#FF4081</color>
6   <color name="year">#999999</color>
7   <color name="title">#222222</color>
8 </resources>
```

# New Resource File



# movie\_list\_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusable="true"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:clickable="true"
    android:background="?android:attr/selectableItemBackground"
    android:orientation="vertical">

    <TextView
        android:id="@+id/title"
        android:textColor="@color/title"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_alignParentTop="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/genre"
        android:layout_below="@id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/year"
        android:textColor="@color/year"
        android:layout_width="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Text mit copy-  
paste kopierbar

# MoviesAdapter.java

```
package at.htl.movies;
```

```
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 MoviesAdapter extends RecyclerView.Adapter<MoviesAdapter.MyViewHolder> {
```

```
    private List<Movie> moviesList;
```

```
    public static class MyViewHolder extends RecyclerView.ViewHolder {  
        public TextView title, year, genre;  
  
        public MyViewHolder(View view) {  
            super(view);  
            title = (TextView) view.findViewById(R.id.title);  
            genre = (TextView) view.findViewById(R.id.genre);  
            year = (TextView) view.findViewById(R.id.year);  
        }  
    }  
}
```

```
    public MoviesAdapter(List<Movie> moviesList) {  
        this.moviesList = moviesList;  
    }  
}
```

@Override

```
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    View itemView = LayoutInflater.from(parent.getContext())  
        .inflate(R.layout.movie_list_row, parent, false);  
  
    return new MyViewHolder(itemView);  
}
```

@Override

```
public void onBindViewHolder(MyViewHolder holder, int position) {  
    Movie movie = moviesList.get(position);  
    holder.title.setText(movie.getTitle());  
    holder.genre.setText(movie.getGenre());  
    holder.year.setText(movie.getYear());  
}
```

@Override

```
public int getItemCount() {  
    return moviesList.size();  
}
```

```
}
```

# MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    private List<Movie> movieList = new ArrayList<>();  
    private RecyclerView recyclerView;  
    private MoviesAdapter mAdapter;
```

`@Override`

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
    setSupportActionBar(toolbar);
```

```
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);  
    fab.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)  
                .setAction("Action", null).show();  
        }  
    });
```

```
    recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
```

```
    mAdapter = new MoviesAdapter(movieList);  
    RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());  
    recyclerView.setLayoutManager(mLayoutManager);  
    recyclerView.setItemAnimator(new DefaultItemAnimator());  
    recyclerView.setAdapter(mAdapter);
```

```
    prepareMovieData();
```

```
}
```

```
private void prepareMovieData() {
    Movie movie = new Movie("Mad Max: Fury Road", "Action & Adventure", "2015");
    movieList.add(movie);

    movie = new Movie("Inside Out", "Animation, Kids & Family", "2015");
    movieList.add(movie);

    movie = new Movie("Star Wars: Episode VII – The Force Awakens", "Action", "2015");
    movieList.add(movie);

    movie = new Movie("Shaun the Sheep", "Animation", "2015");
    movieList.add(movie);

    movie = new Movie("The Martian", "Science Fiction & Fantasy", "2015");
    movieList.add(movie);

    movie = new Movie("Mission: Impossible Rogue Nation", "Action", "2015");
    movieList.add(movie);

    movie = new Movie("Up", "Animation", "2009");
    movieList.add(movie);

    movie = new Movie("Star Trek", "Science Fiction", "2009");
    movieList.add(movie);

    movie = new Movie("The LEGO Movie", "Animation", "2014");
    movieList.add(movie);

    movie = new Movie("Iron Man", "Action & Adventure", "2008");
    movieList.add(movie);
}
```



```
movie = new Movie("Aliens", "Science Fiction", "1986");
movieList.add(movie);

movie = new Movie("Chicken Run", "Animation", "2000");
movieList.add(movie);

movie = new Movie("Back to the Future", "Science Fiction", "1985");
movieList.add(movie);

movie = new Movie("Raiders of the Lost Ark", "Action & Adventure", "1981");
movieList.add(movie);

movie = new Movie("Goldfinger", "Action & Adventure", "1965");
movieList.add(movie);

movie = new Movie("Guardians of the Galaxy", "Science Fiction & Fantasy", "2014");
movieList.add(movie);

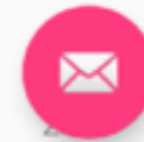
mAdapter.notifyDataSetChanged();
}

...
```

# Movies



- Mad Max: Fury Road** 2015  
Action & Adventure
- Inside Out** 2015  
Animation, Kids & Family
- Star Wars: Episode VII - The Force Awakens** 2015  
Action
- Shaun the Sheep** 2015  
Animation
- The Martian** 2015  
Science Fiction & Fantasy
- Mission: Impossible Rogue Nation** 2015  
Action
- Up** 2009  
Animation
- Star Trek** 2009  
Science Fiction
- The LEGO Movie** 2014  
Animation
- Iron Man**



Item Divider einfügen

# DividerItemDecoration.java

```
package at.htl.movies;

import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.graphics.Rect;
import android.graphics.drawable.Drawable;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;

public class DividerItemDecoration extends RecyclerView.ItemDecoration {
    private static final int[] ATTRS = new int[]{
        android.R.attr.listDivider
    };

    public static final int HORIZONTAL_LIST = LinearLayoutManager.HORIZONTAL;

    public static final int VERTICAL_LIST = LinearLayoutManager.VERTICAL;

    private Drawable mDivider;

    private int mOrientation;

    public DividerItemDecoration(Context context, int orientation) {
        final TypedArray a = context.obtainStyledAttributes(ATTRS);
        mDivider = a.getDrawable(0);
        a.recycle();
        setOrientation(orientation);
    }
}
```

Text mit copy-  
paste kopierbar

```

public void setOrientation(int orientation) {
    if (orientation != HORIZONTAL_LIST && orientation != VERTICAL_LIST) {
        throw new IllegalArgumentException("invalid orientation");
    }
    mOrientation = orientation;
}

@Override
public void onDrawOver(Canvas c, RecyclerView parent, RecyclerView.State state) {
    if (mOrientation == VERTICAL_LIST) {
        drawVertical(c, parent);
    } else {
        drawHorizontal(c, parent);
    }
}

public void drawVertical(Canvas c, RecyclerView parent) {
    final int left = parent.getPaddingLeft();
    final int right = parent.getWidth() - parent.getPaddingRight();

    final int childCount = parent.getChildCount();
    for (int i = 0; i < childCount; i++) {
        final View child = parent.getChildAt(i);
        final RecyclerView.LayoutParams params = (RecyclerView.LayoutParams) child
            .getLayoutParams();
        final int top = child.getBottom() + params.bottomMargin;
        final int bottom = top + mDivider.getIntrinsicHeight();
        mDivider.setBounds(left, top, right, bottom);
        mDivider.draw(c);
    }
}

```

Text mit copy-  
paste kopierbar

```

public void drawHorizontal(Canvas c, RecyclerView parent) {
    final int top = parent.getPaddingTop();
    final int bottom = parent.getHeight() - parent.getPaddingBottom();

    final int childCount = parent.getChildCount();
    for (int i = 0; i < childCount; i++) {
        final View child = parent.getChildAt(i);
        final RecyclerView.LayoutParams params = (RecyclerView.LayoutParams) child
            .getLayoutParams();
        final int left = child.getRight() + params.rightMargin;
        final int right = left + mDivider.getIntrinsicHeight();
        mDivider.setBounds(left, top, right, bottom);
        mDivider.draw(c);
    }
}

@Override
public void getItemOffsets(Rect outRect, View view, RecyclerView parent,
RecyclerView.State state) {
    if (mOrientation == VERTICAL_LIST) {
        outRect.set(0, 0, 0, mDivider.getIntrinsicHeight());
    } else {
        outRect.set(0, 0, mDivider.getIntrinsicWidth(), 0);
    }
}
}

```

Text mit copy-  
paste kopierbar

# MainActivity.java

```
mAdapter = new MoviesAdapter(movieList);
RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());
recyclerView.setLayoutManager(mLayoutManager);
recyclerView.setItemAnimator(new DefaultItemAnimator());
recyclerView.addItemDecoration(new DividerItemDecoration(this, LinearLayoutManager.VERTICAL));
recyclerView.setAdapter(mAdapter);

prepareMovieData();
```

# Movies



**Mad Max: Fury Road** 2015  
Action & Adventure

**Inside Out** 2015  
Animation, Kids & Family

**Star Wars: Episode VII - The Force Awakens** 2015  
Action

**Shaun the Sheep** 2015  
Animation

**The Martian** 2015  
Science Fiction & Fantasy

**Mission: Impossible Rogue Nation** 2015  
Action

**Up** 2009  
Animation

**Star Trek** 2009  
Science Fiction

**The LEGO Movie** 2014  
Animation





# Probleme beim ViewHolder pattern

<http://androidshenanigans.blogspot.co.at/2015/02/viewholder-pattern-common-mistakes.html>

Add Item Click  
Listener

# Problems

- RecyclerView doesn't have OnItemClickListener method too to identify item click. You need to write your own class extending RecyclerView.OnItemTouchListener.

# MainActivity.java

```
package at.htl.movies;

import android.content.Context;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.DefaultItemAnimator;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private List<Movie> movieList = new ArrayList<>();
    private RecyclerView recyclerView;
    private MoviesAdapter mAdapter;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });

    recyclerView = (RecyclerView) findViewById(R.id.recycler_view);

    mAdapter = new MoviesAdapter(movieList);

    // recyclerView.setHasFixedSize(true);
    RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());
    recyclerView.setLayoutManager(mLayoutManager);
    recyclerView.setItemAnimator(new DefaultItemAnimator());
    recyclerView.addItemDecoration(new DividerItemDecoration(this, LinearLayoutManager.VERTICAL));
    recyclerView.setAdapter(mAdapter);

    recyclerView.addOnItemTouchListener(new RecyclerViewTouchListener(
        getApplicationContext(),
        recyclerView,
        new ClickListener() {
            @Override
            public void onClick(View view, int position) {
                Movie movie = movieList.get(position);
                Toast.makeText(getApplicationContext(), movie.getTitle() + " is selected!",
                    Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onLongClick(View view, int position) {

            }
        }
    ));

    prepareMovieData();
}

```

```
private void prepareMovieData() {  
    Movie movie = new Movie("Mad Max: Fury Road", "Action & Adventure", "2015");  
    movieList.add(movie);  
  
    movie = new Movie("Inside Out", "Animation, Kids & Family", "2015");  
    movieList.add(movie);  
  
    movie = new Movie("Star Wars: Episode VII – The Force Awakens", "Action", "2015");  
    movieList.add(movie);  
  
    ...  
  
    movie = new Movie("Raiders of the Lost Ark", "Action & Adventure", "1981");  
    movieList.add(movie);  
  
    movie = new Movie("Goldfinger", "Action & Adventure", "1965");  
    movieList.add(movie);  
  
    movie = new Movie("Guardians of the Galaxy", "Science Fiction & Fantasy", "2014");  
    movieList.add(movie);  
  
    mAdapter.notifyDataSetChanged();  
}
```

```

public interface ClickListener {
    void onClick(View view, int position);

    void onLongClick(View view, int position);
}

public static class RecyclerTouchListener implements RecyclerView.OnItemTouchListener {

    private GestureDetector gestureDetector;
    private MainActivity.ClickListener clickListener;

    public RecyclerTouchListener(Context context,
                                final RecyclerView recyclerView,
                                final MainActivity.ClickListener clickListener) {
        this.clickListener = clickListener;
        gestureDetector = new GestureDetector(context,
                                             new GestureDetector.SimpleOnGestureListener() {

@OVERRIDE
public boolean onSingleTapUp(MotionEvent e) {
    return true;
}

@OVERRIDE
public void onLongPress(MotionEvent e) {
    View child = recyclerView.findViewById(e.getX(), e.getY());
    if (child != null && clickListener != null) {
        clickListener.onLongClick(child, recyclerView.getChildPosition(child));
    }
}
});
}
}

```

@Override

```
public boolean onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {
```

```
    View child = rv.findViewById(e.getX(), e.getY());
```

```
    if (child != null && clickListener != null && gestureDetector.onTouchEvent(e)) {  
        clickListener.onClick(child, rv.getChildPosition(child));
```

```
    }
```

```
    return false;
```

```
}
```

@Override

```
public void onTouchEvent(RecyclerView rv, MotionEvent e) {
```

```
}
```

@Override

```
public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {
```

```
}
```

```
}
```



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}
```

# Movies



**Mad Max: Fury Road** 2015  
Action & Adventure

**Inside Out** 2015  
Animation, Kids & Family

**Star Wars: Episode VII - The Force Awakens** 2015  
Action

**Shaun the Sheep** 2015  
Animation

**The Martian** 2015  
Science Fiction & Fantasy

**Mission: Impossible Rogue Nation** 2015  
Action

**Up** 2009  
Animation

**Star Trek** 2009  
Science Fiction

**The LEGO Movie** 2014  
Animation

The Martian is selected!





Noch  
Fragen?