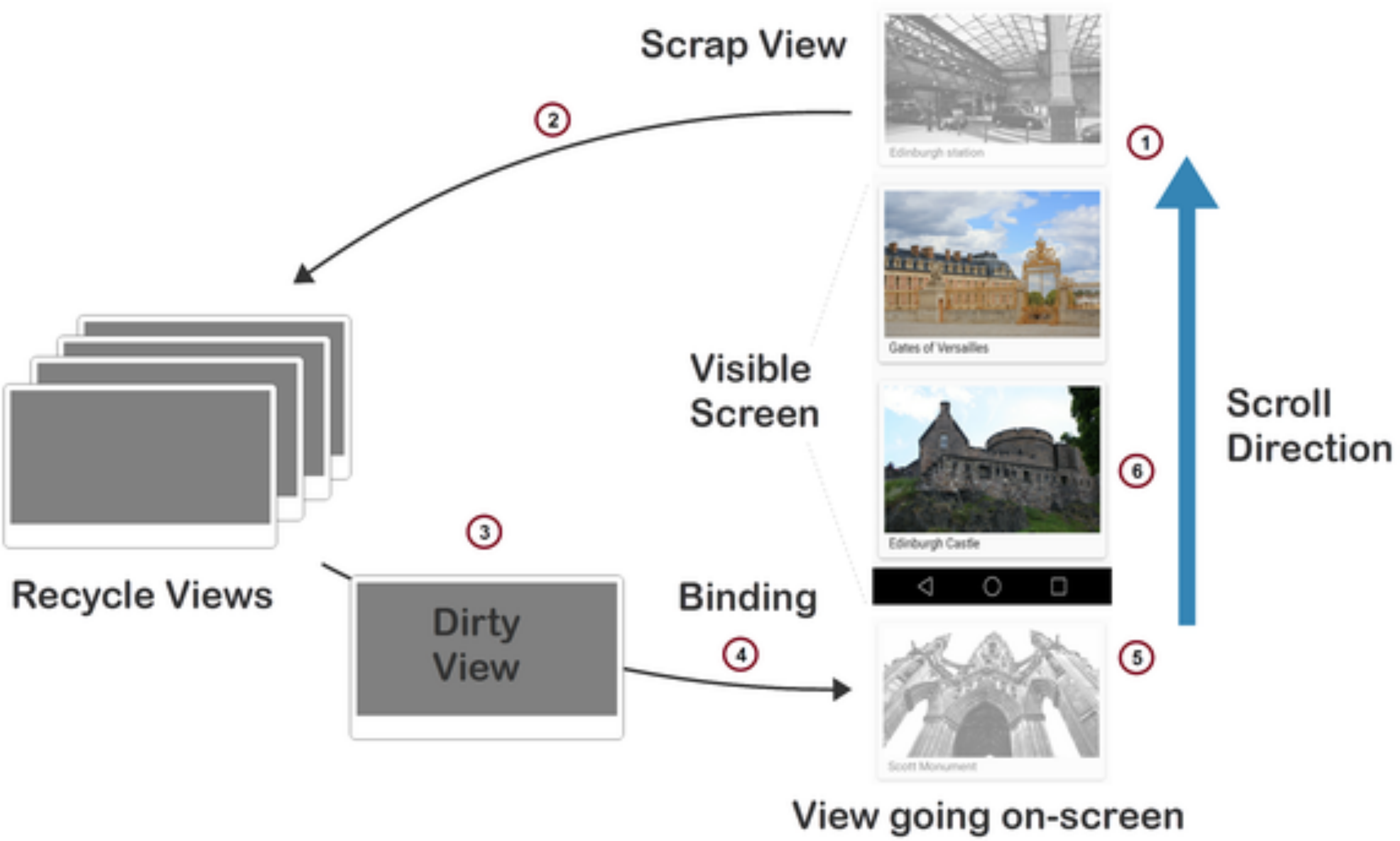
RecyclerView

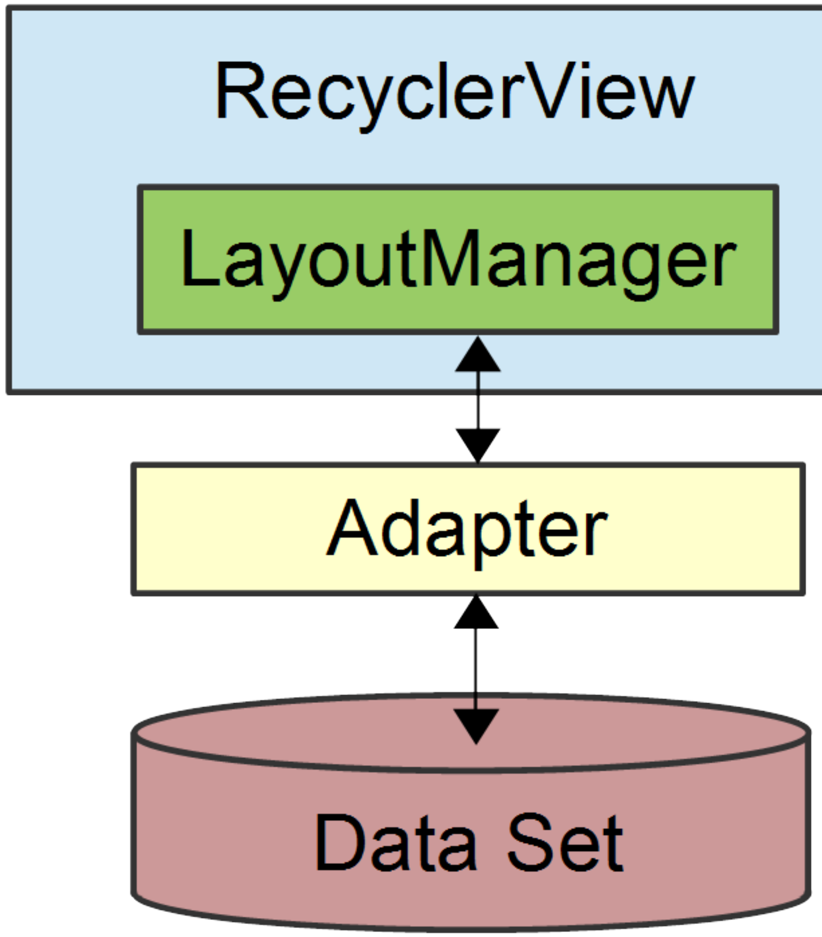
Dev Radio

RecyclerView



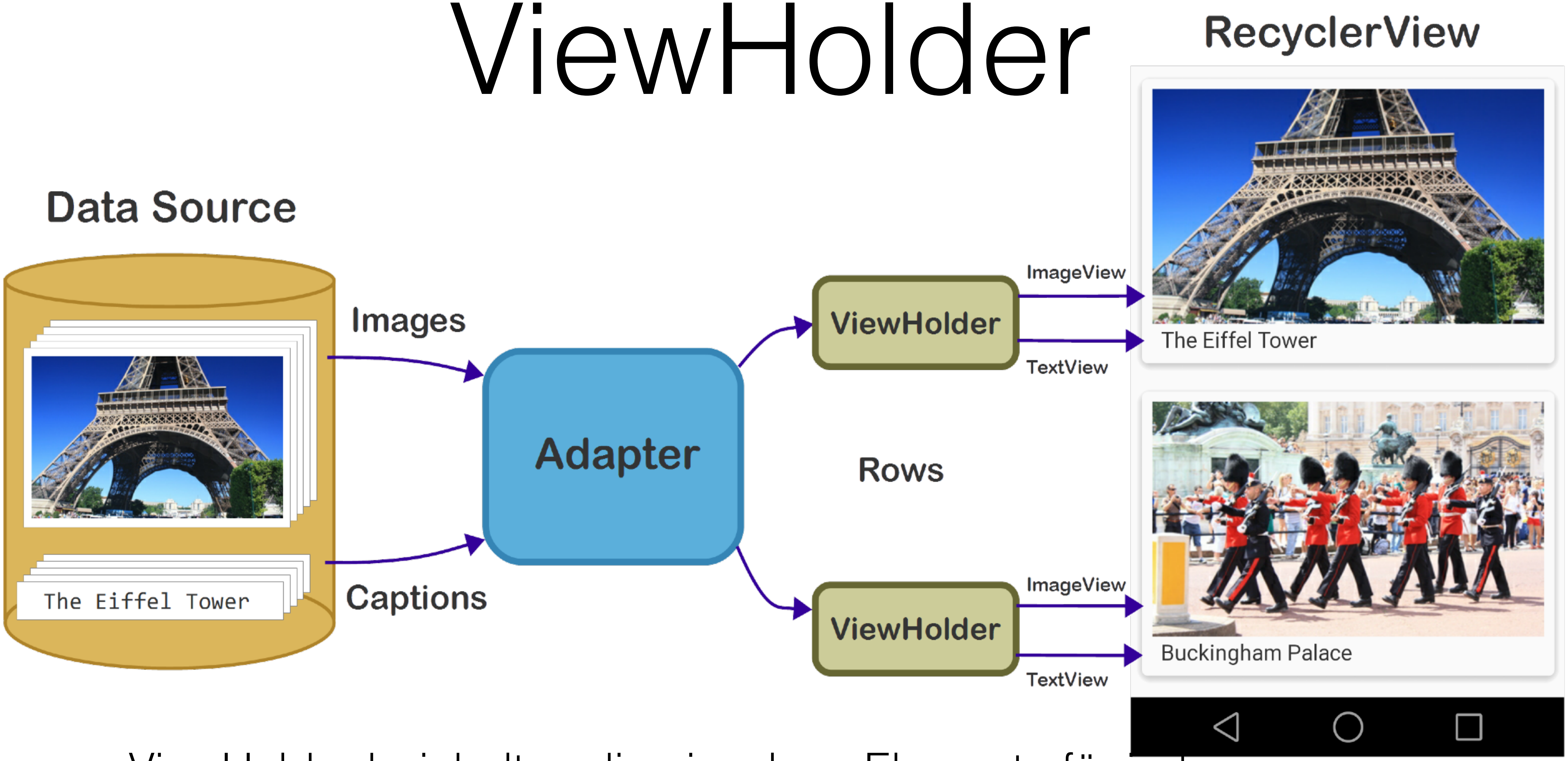
- Eine RecyclerView „recyclet" Ihre einzelnen Element beim Scrollen
- Es werden nicht neue Elemente erzeugt, sondern die nicht mehr sichtbaren Elemente mit neuen Inhalten befüllt und an das sichtbare Ende der Liste angefügt

Adapter



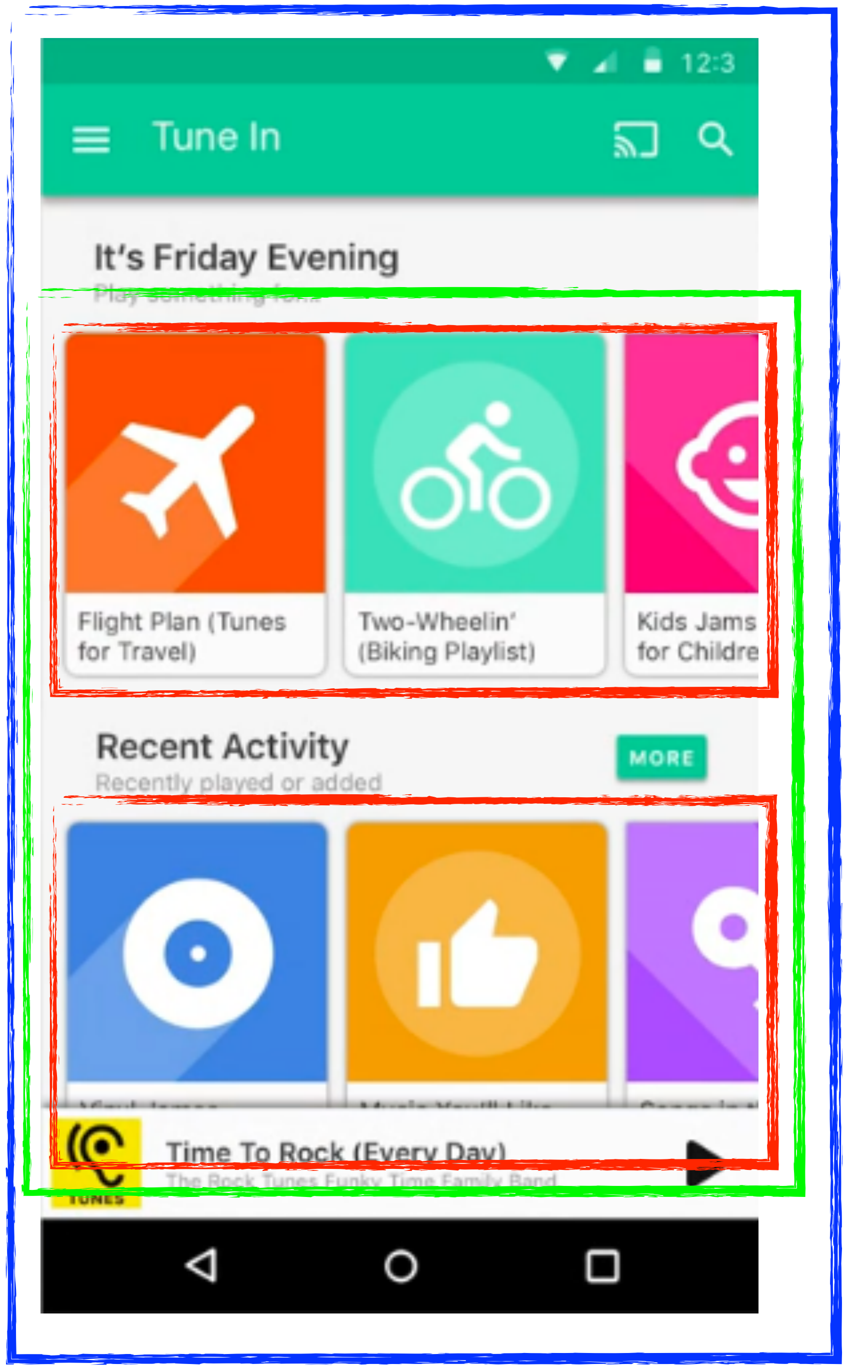
- Ein Adapter ist für die Versorgung der RecylerView mit Daten verantwortlich
- Der LayoutManager für das Aussehen der RecyclerView
- Man kann hier gut das modulare Konzept von Android erkennen

ViewHolder



- ViewHolder beinhalten die einzelnen Elemente für jeden Listeneintrag und sind direkt dem Adapter zugeordnet. So muss man nicht (langwierig) über R.id.xxx suchen, sondern kann direkt über das tag-filed des Adapters auf sie zugreifen

Bsp Dev Radio

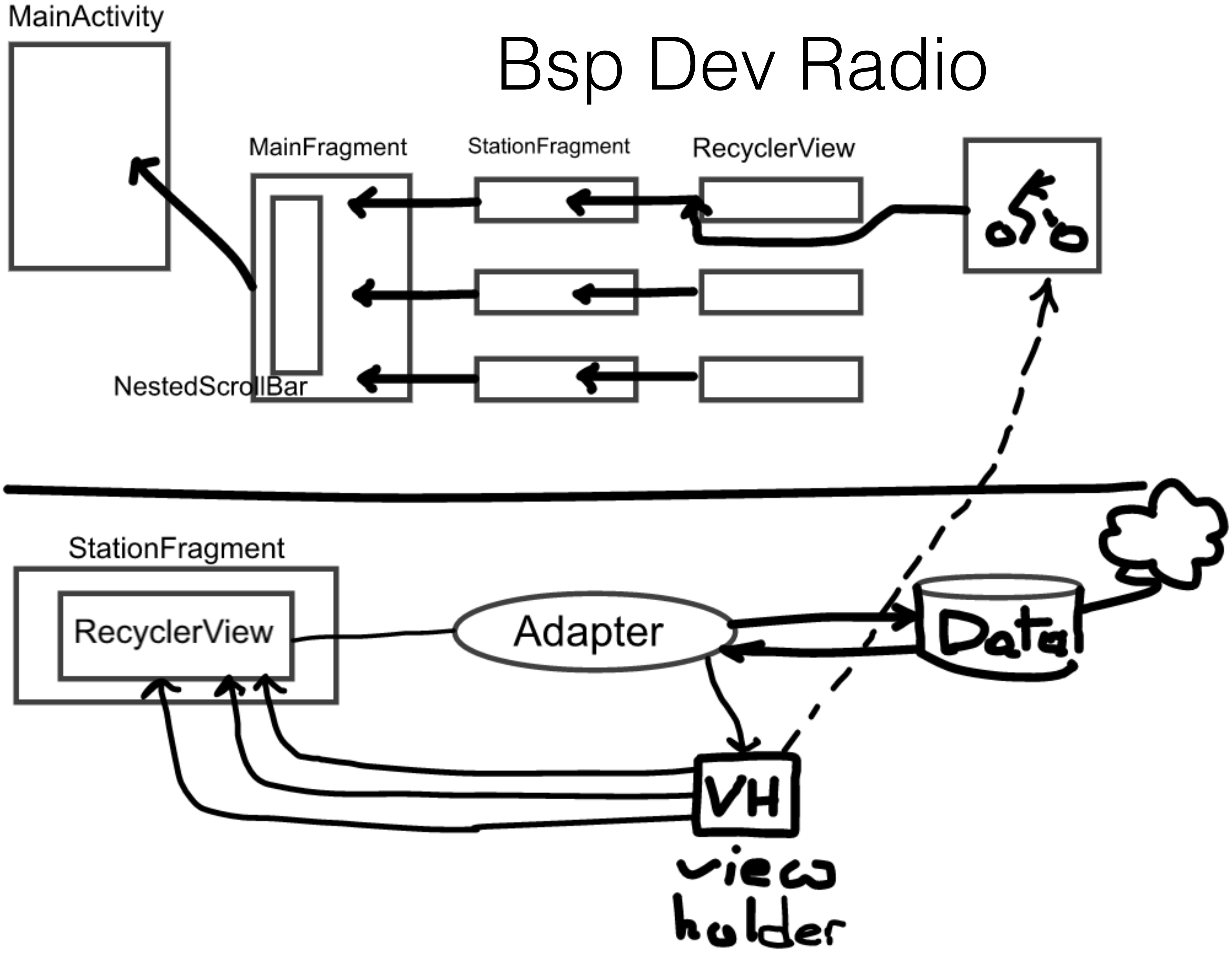


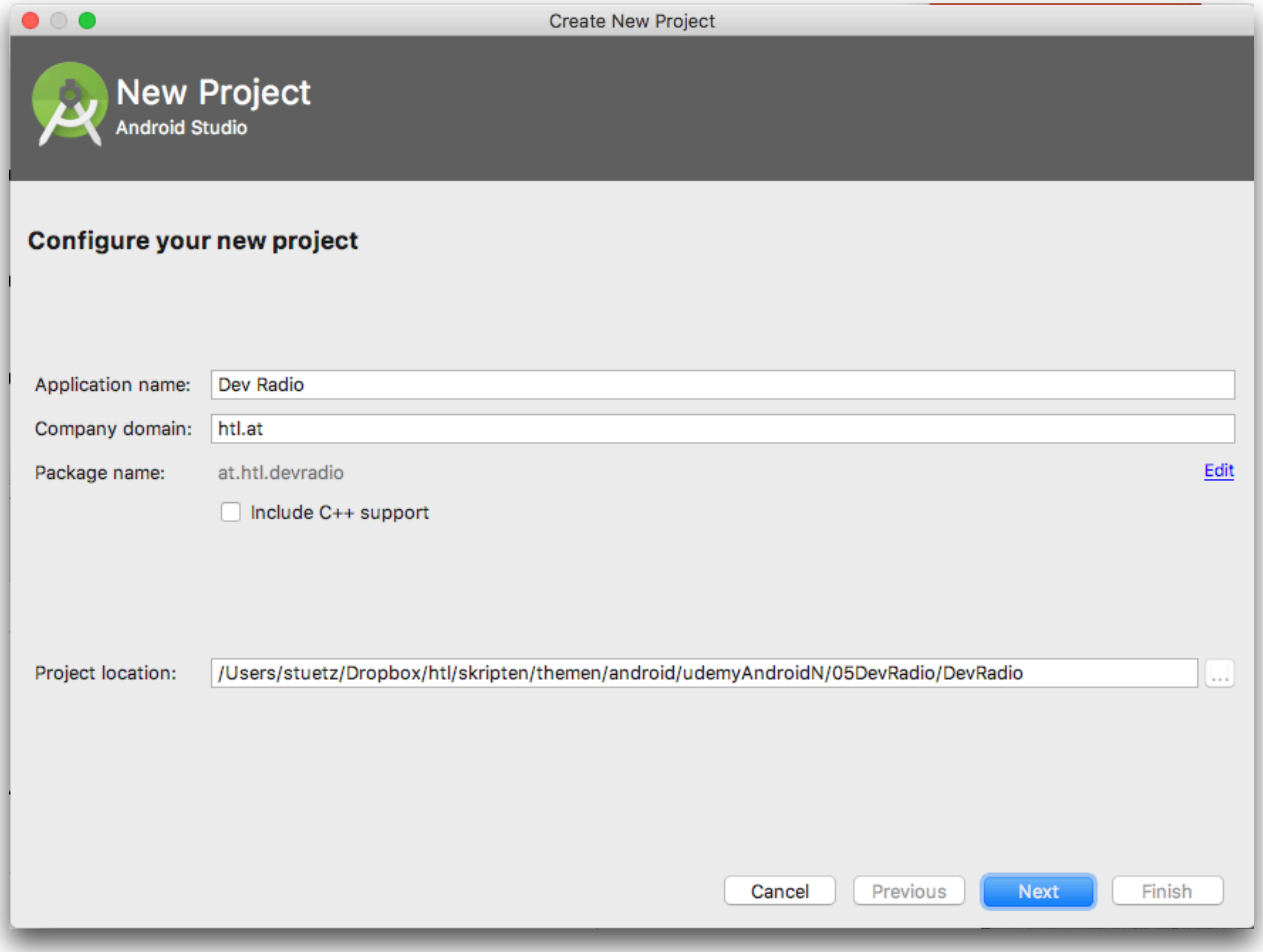
StationFragments

MainFragment

MainActivity

Bsp Dev Radio





Create New Project

New Project
Android Studio

Configure your new project

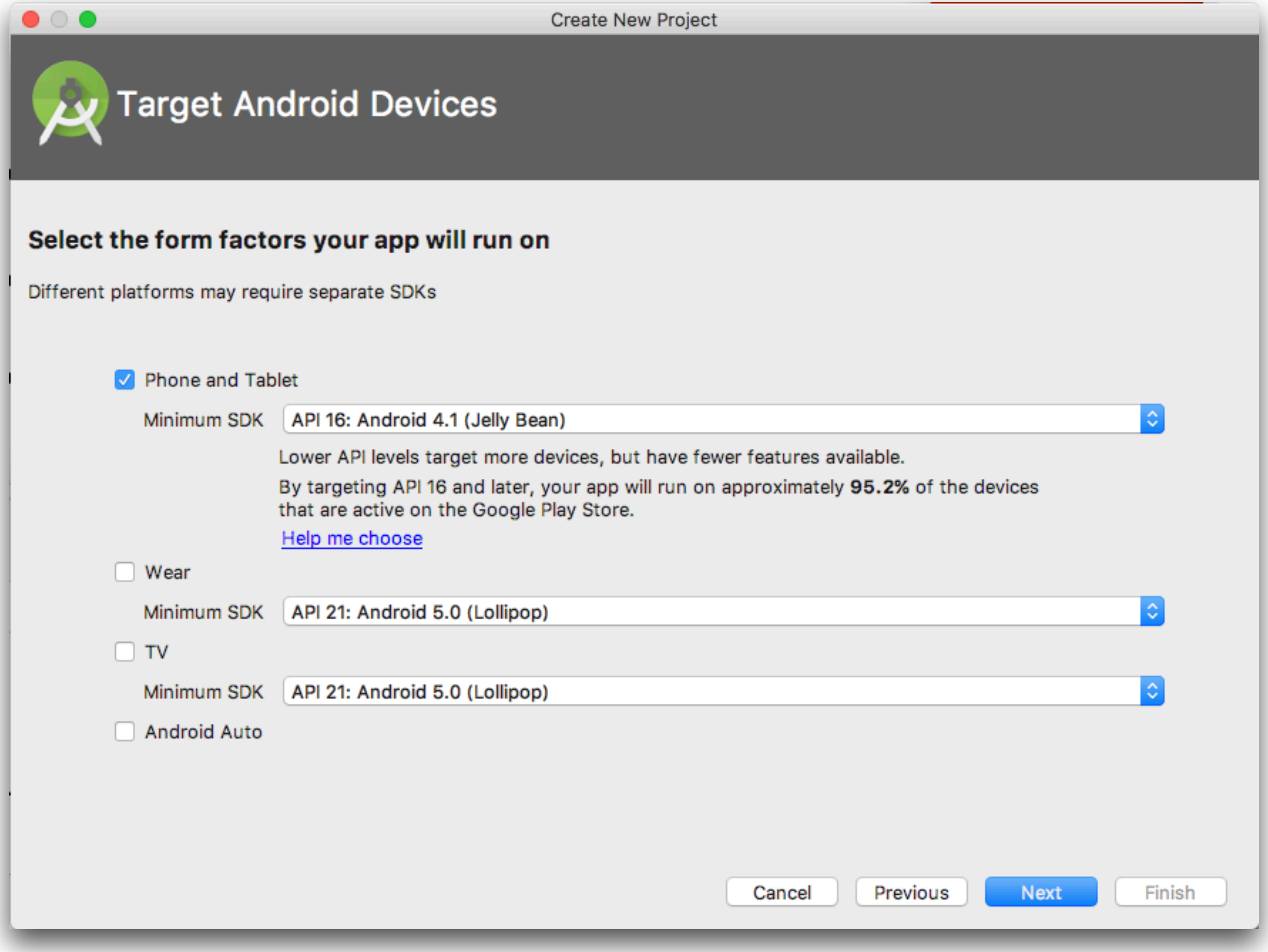
Application name: Dev Radio

Company domain: htl.at

Package name: at.htl.devradio

☐ Include C++ support

Project location: /Users/stuetz/Dropbox/htl/skripten/themen/android/udemyAndroidN/05DevRadio/DevRadio



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 16: Android 4.1 (Jelly Bean)

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

By targeting API 16 and later, your app will run on approximately **95.2%** of the devices that are active on the Google Play Store.

Help me choose

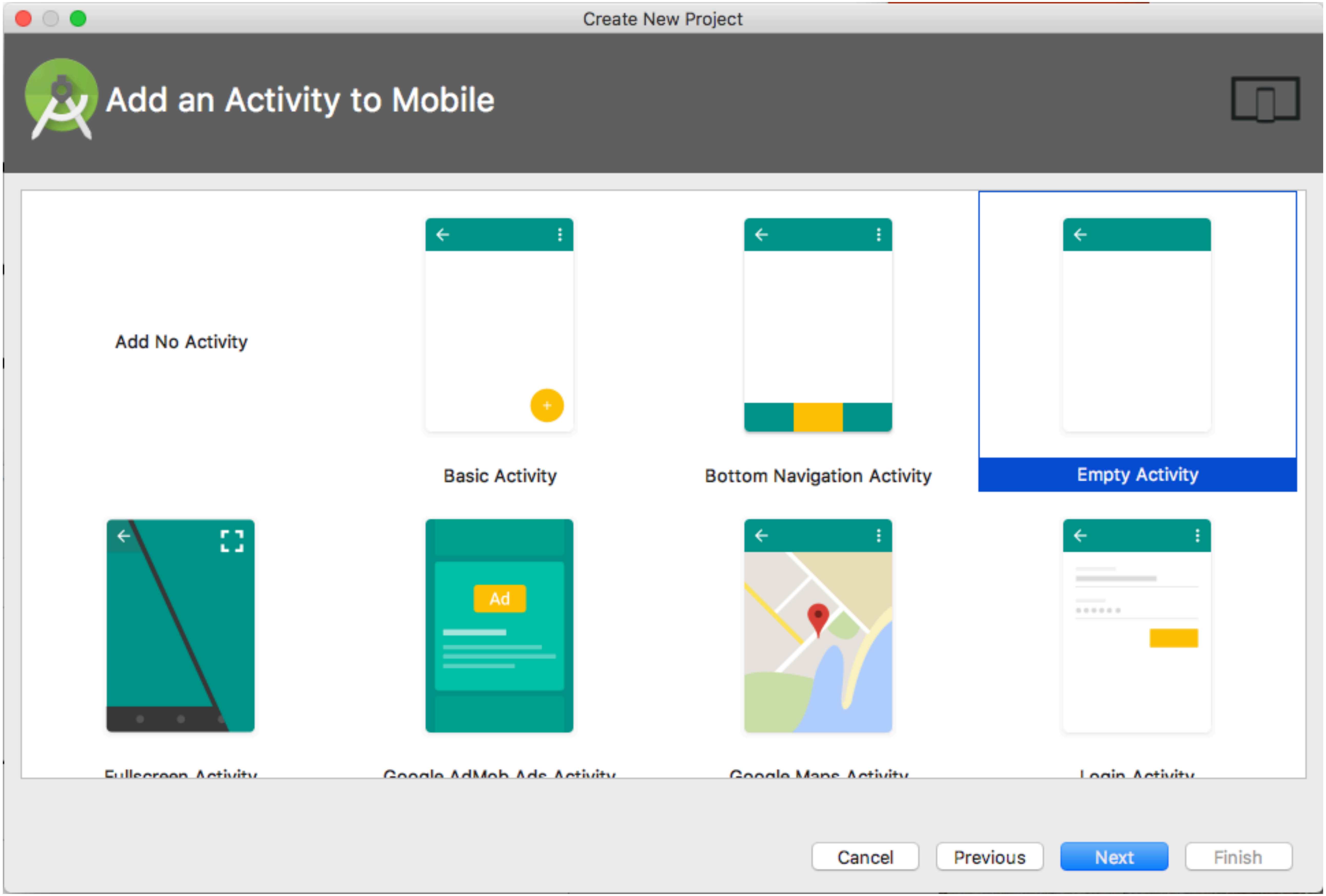
□ Wear

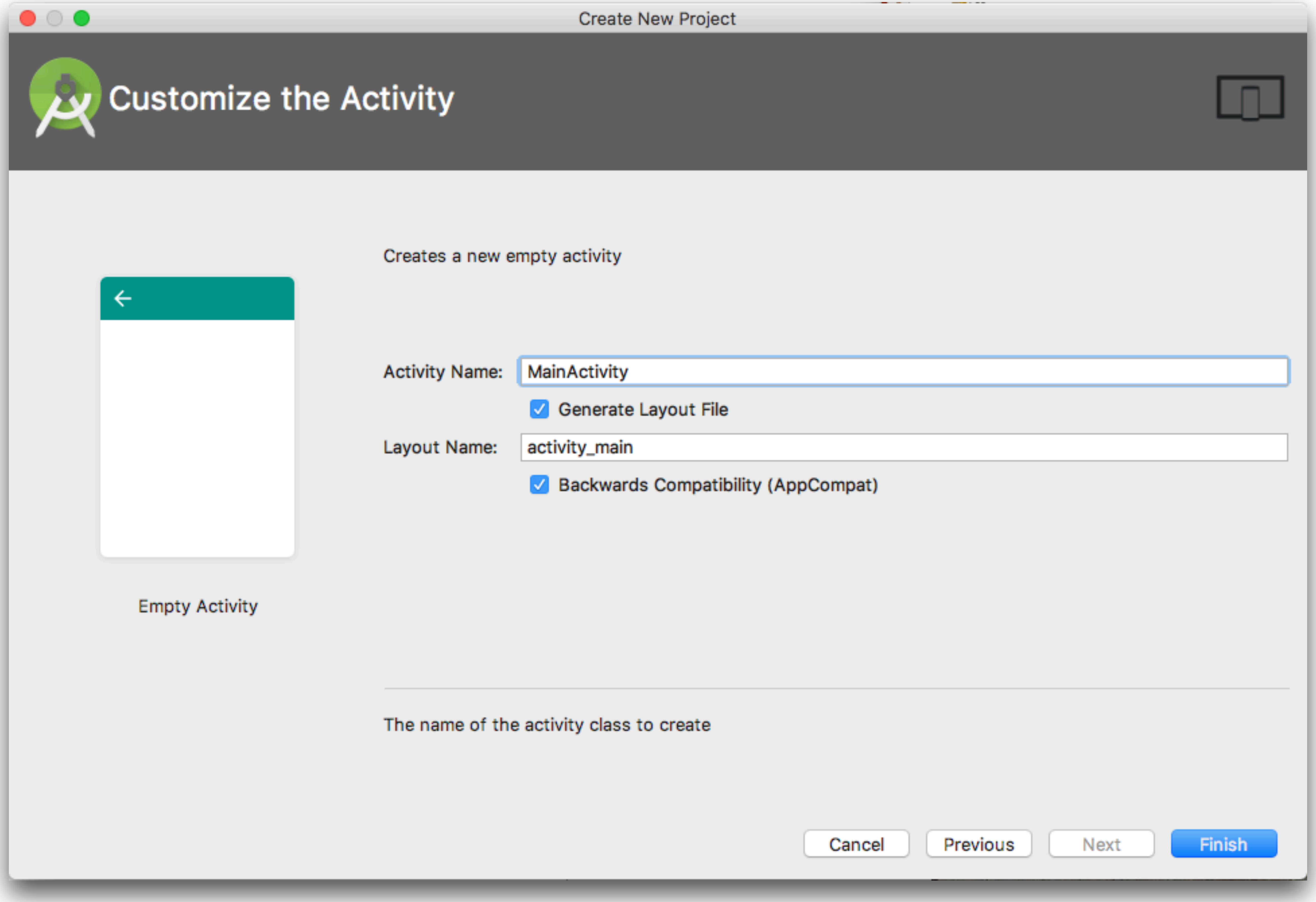
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

Create New Project
Add an Activity to Mobile
Add No Activity
Basic Activity
Bottom Navigation Activity
Empty Activity
Ad
Fullscreen Activity
Google AdMob Ads Activity
Google Maps Activity
Login Activity
Cancel
Previous
Next
Finish



Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

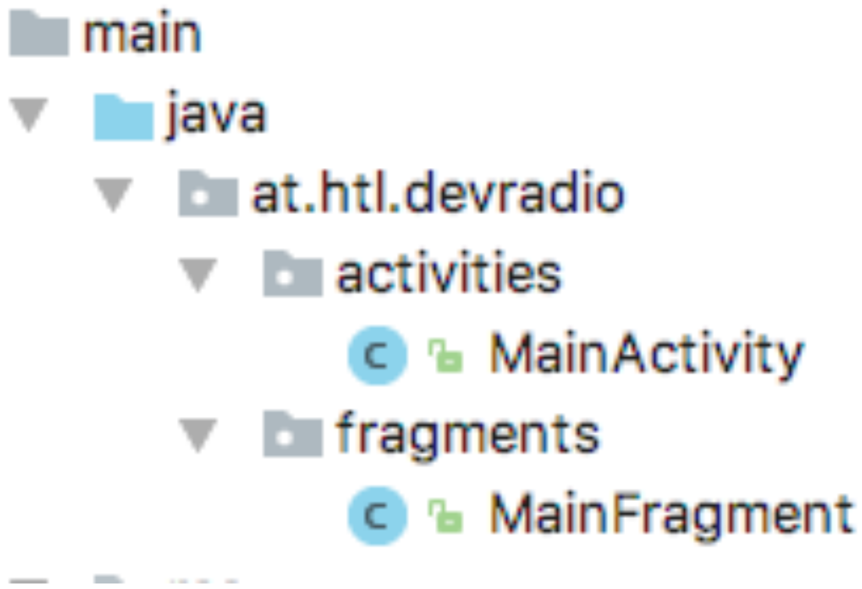
Layout Name: activity_main

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

Vorarbeiten

- Erstellen Sie nun zwei Packages
  - fragments
  - activites

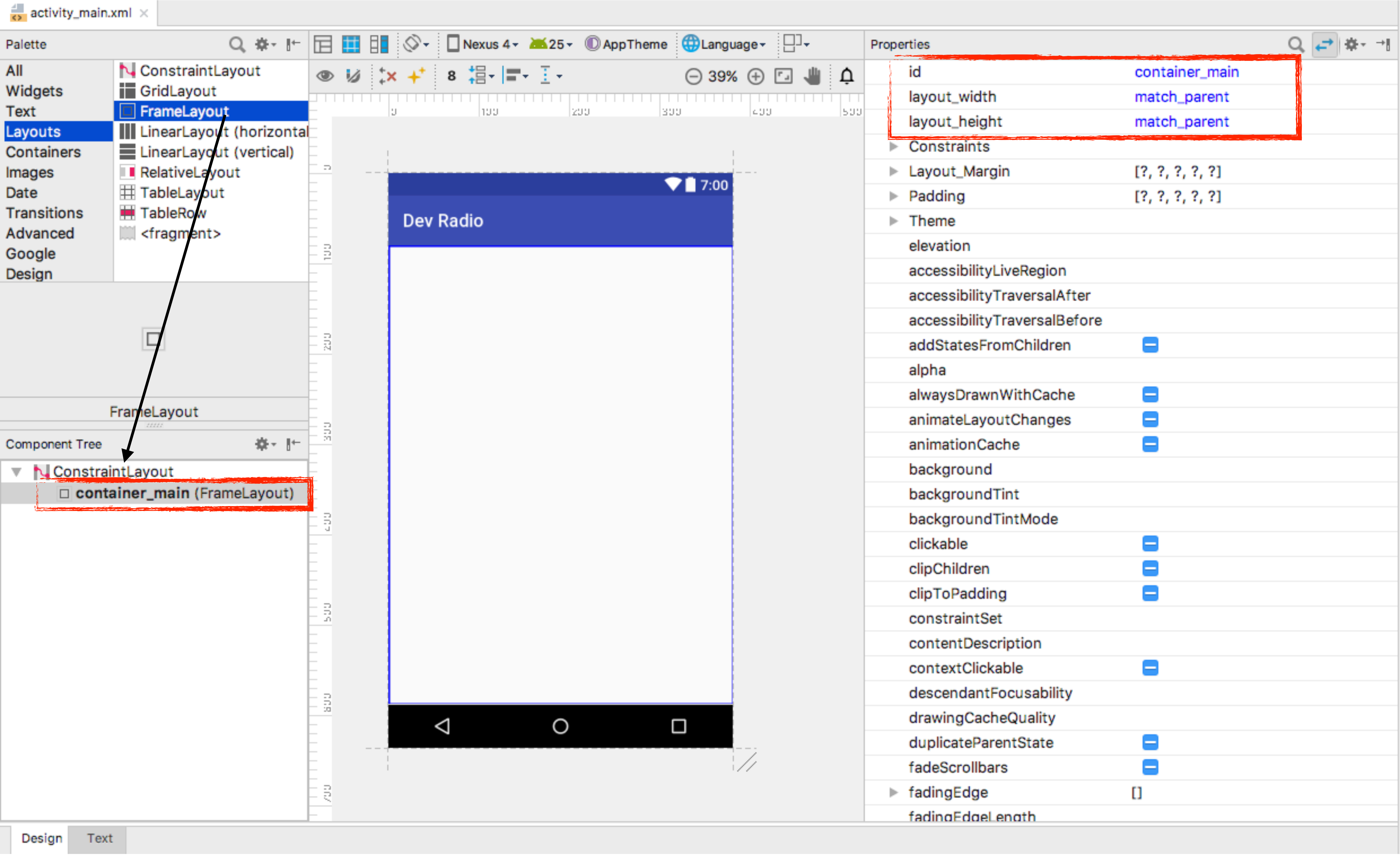


```
main
▼ java
  ▼ at.htl.devradio
    ▼ activities
        MainActivity
    ▼ fragments
        MainFragment
```

- Verschieben Sie die MainActivity.java nach activities
- Erstellen sie weiters ein neues Fragment (blank fragment) mit Namen „MainFragment.java“

Aufgabe

- Rufen Sie nun das Fragment auf



ConstraintLayout
□ container_main (FrameLayout)

id	container_main
layout_width	match_parent
layout_height	match_parent
Constraints	
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
addStatesFromChildren	⊟
alpha	
alwaysDrawnWithCache	⊟
animateLayoutChanges	⊟
animationCache	⊟
background	
backgroundTint	
backgroundTintMode	
clickable	⊟
clipChildren	⊟
clipToPadding	⊟
constraintSet	
contentDescription	
contextClickable	⊟
descendantFocusability	
drawingCacheQuality	
duplicateParentState	⊟
fadeScrollbars	⊟
fadingEdge	[]
fadingEdgeLength	

activity_main.xml

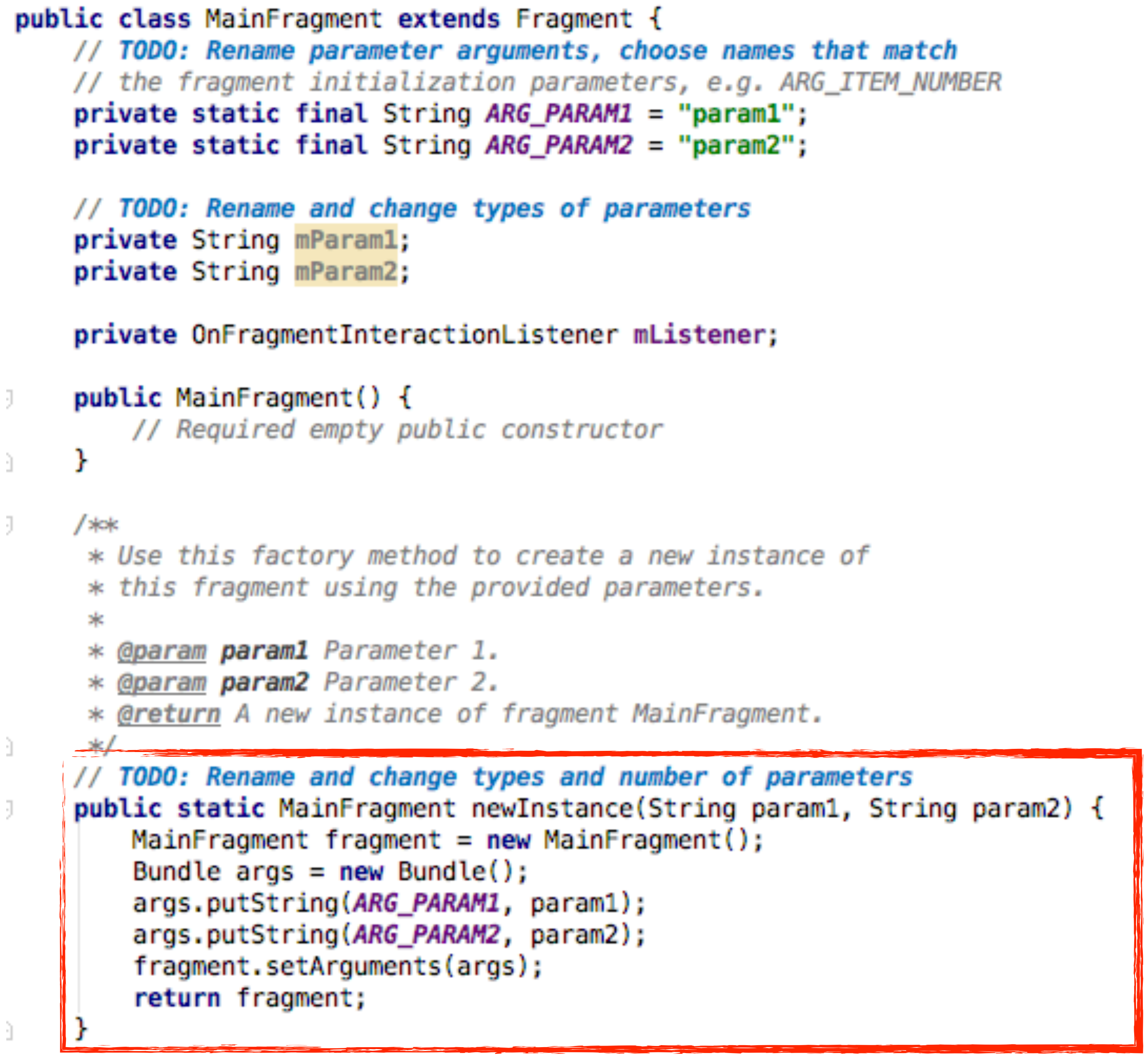
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="at.htl.devradio.activities.MainActivity">


    <FrameLayout
        android:id="@+id/container_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </FrameLayout>

</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```java
package at.htl.devradio.activities;

import android.support.v4.app.FragmentManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import at.htl.devradio.R;
import at.htl.devradio.fragments.MainFragment;

public class MainActivity extends AppCompatActivity {

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

        FragmentManager fm = getSupportFragmentManager();
        MainFragment mainFragment = (MainFragment) fm.findFragmentById(R.id.container_main);

        if (mainFragment == null) {
            mainFragment = MainFragment.newInstance("blah", „kah");
            fm.beginTransaction().add(R.id.container_main, mainFragment).commit();

        }
    }
}
```

Wird ein Fragment auf diese Weise erstellt, können gleichzeitig Daten übergeben werden (zB Hintergrundfarbe)

⌘-Click auf MainFragment

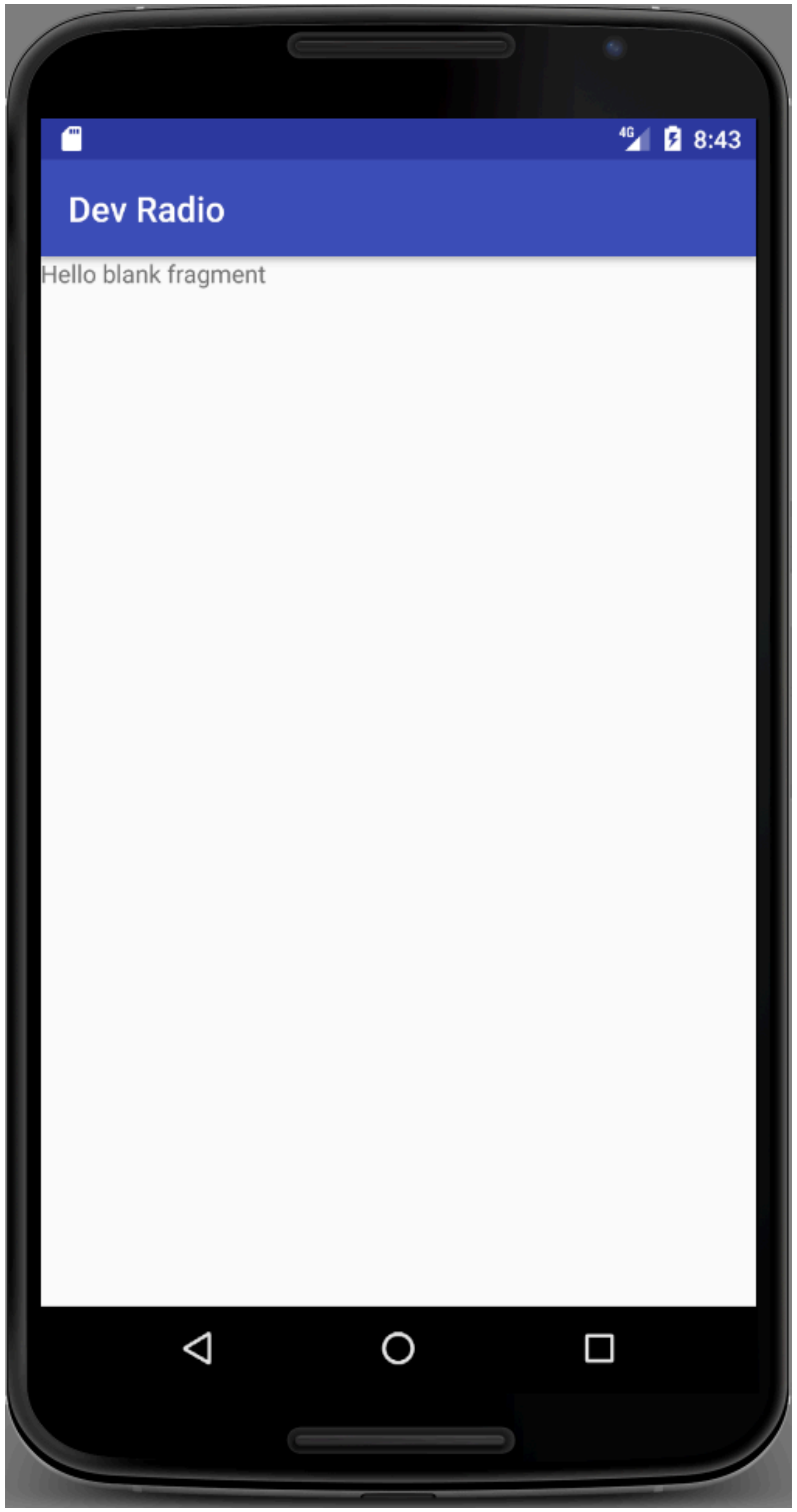


```java
public class MainFragment extends Fragment {
    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    private OnFragmentInteractionListener mListener;

    public MainFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment MainFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static MainFragment newInstance(String param1, String param2) {
        MainFragment fragment = new MainFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }
```

Hier sieht man erstmalig die Verwendung von Bundles

Bundles sind key/value-Speicher

Probelauf



8:43

Dev Radio

Hello blank fragment

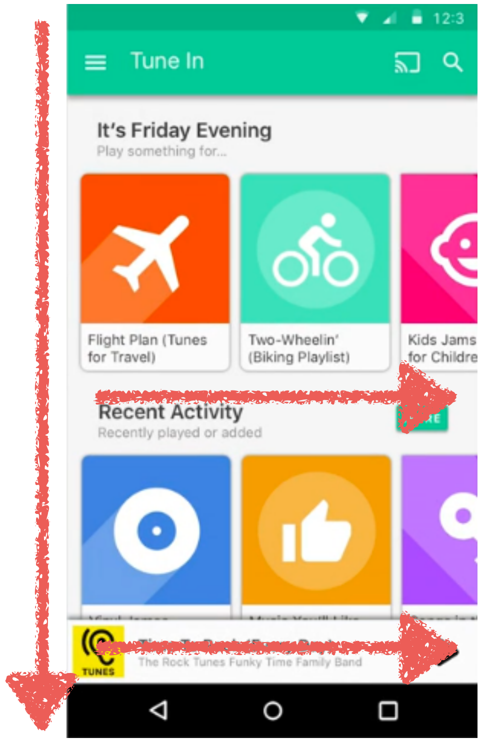
- Nicht vergessen: OnFragmentInteractionListener in MainActivity.java implementieren

MainActivity.java

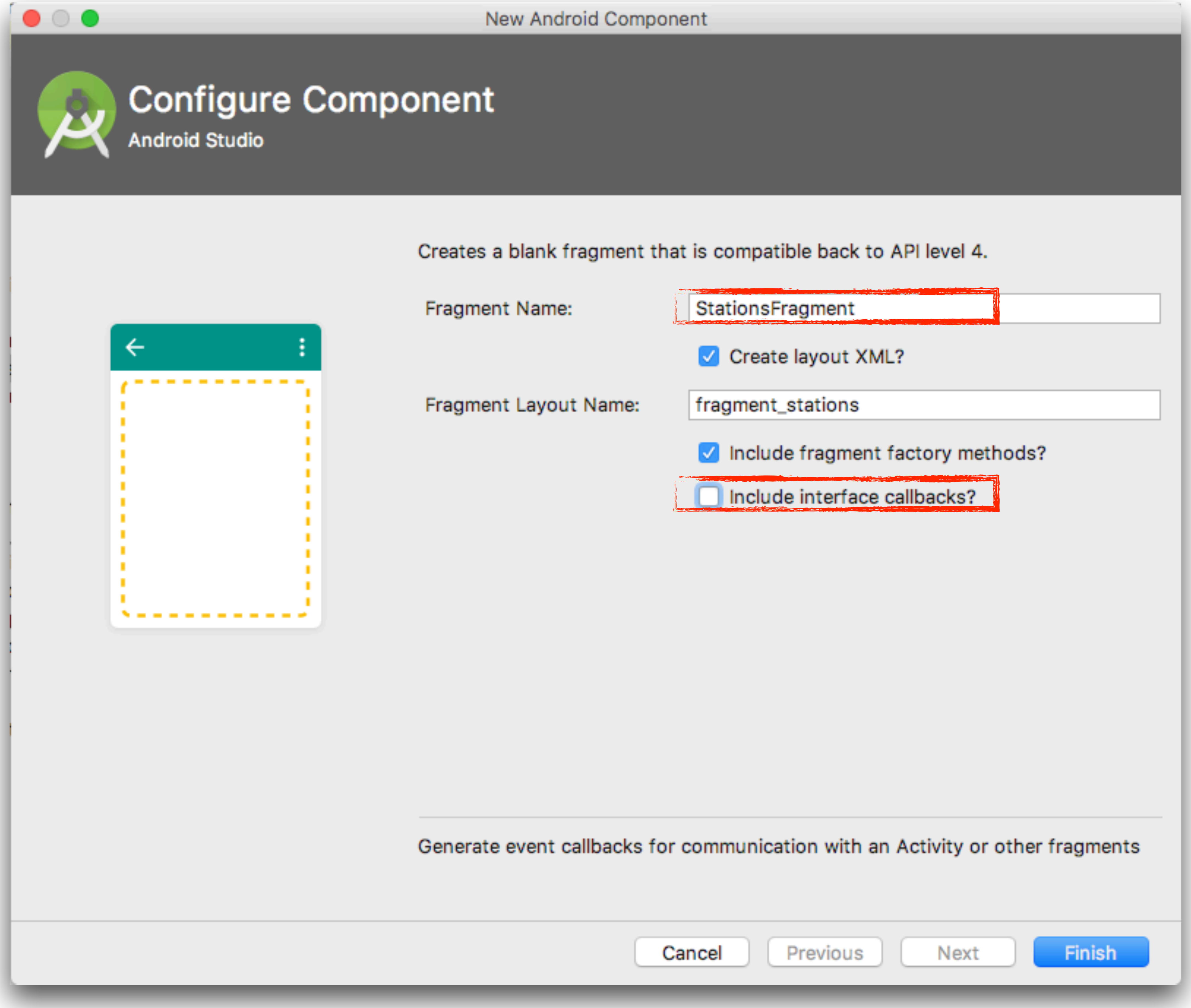
```java
package at.htl.devradio.activities;

import android.net.Uri;
import android.support.v4.app.FragmentManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import at.htl.devradio.R;
import at.htl.devradio.fragments.MainFragment;

public class MainActivity extends AppCompatActivity implements MainFragment.OnFragmentInteractionListener {

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

        FragmentManager fm = getSupportFragmentManager();
        MainFragment mainFragment = (MainFragment) fm.findFragmentById(R.id.container_main);

        if (mainFragment == null) {
            mainFragment = MainFragment.newInstance("blah", "kah");
            fm.beginTransaction().add(R.id.container_main, mainFragment).commit();
        }
    }

    @Override
    public void onFragmentInteraction(Uri uri) {

    }
}
```

fragment_main.xml

```xml
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="at.htl.devradio.fragments.MainFragment">

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"></LinearLayout>
    </android.support.v4.widget.NestedScrollView>

</FrameLayout>
```



Bei verschachtelten Scrollbars ist eine NestedScrollView empfehlenswert. An den Pfeilen sieht man die Scrollbars.



New Android Component

Configure Component
Android Studio

Creates a blank fragment that is compatible back to API level 4.

Fragment Name: StationsFragment

☑ Create layout XML?

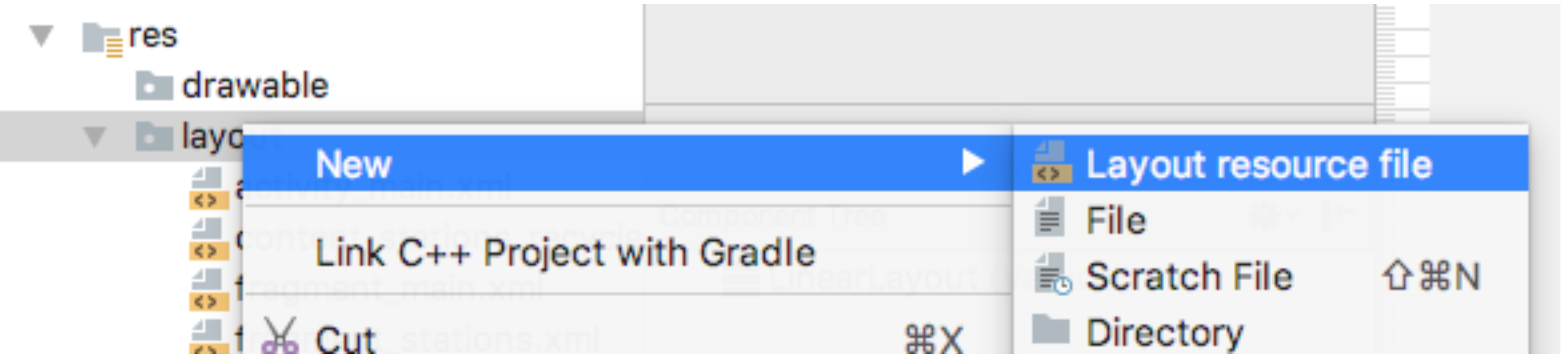
Fragment Layout Name: fragment_stations

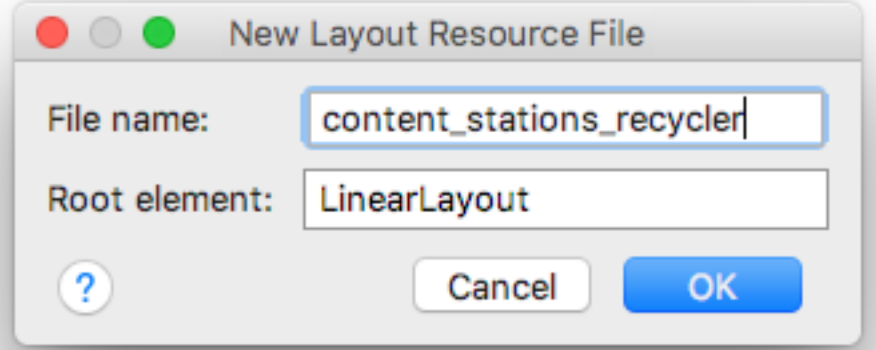
☑ Include fragment factory methods?

□ Include interface callbacks?

Generate event callbacks for communication with an Activity or other fragments

Erstellen einer RecyclerView

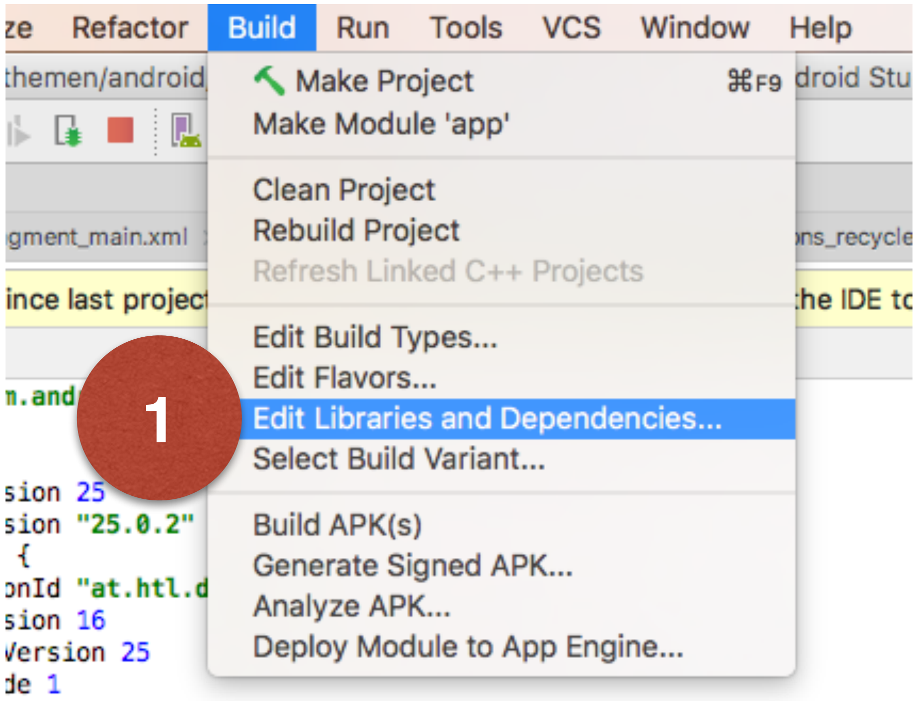



New Layout Resource File

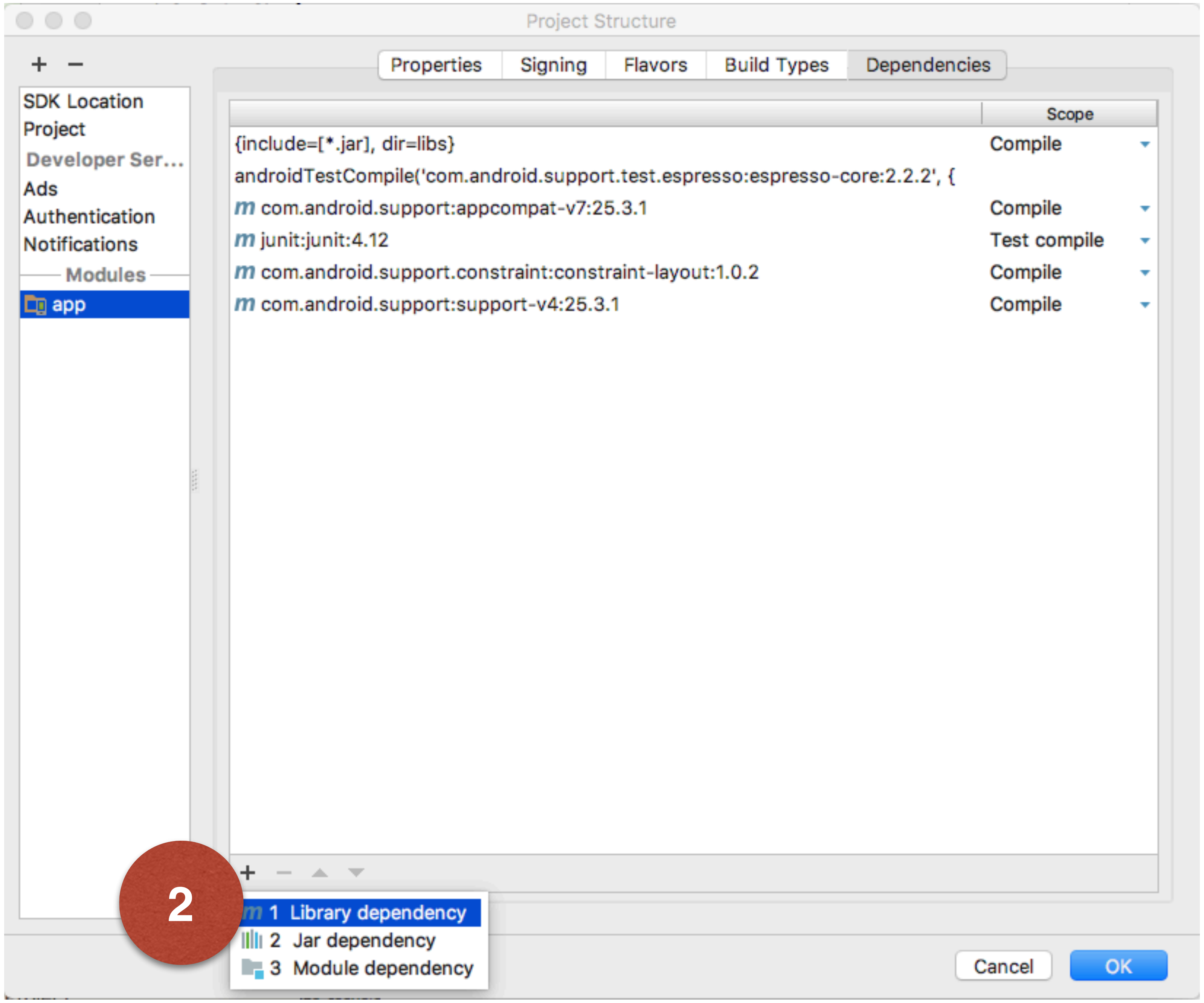
File name: content_stations_recycler

Root element: LinearLayout

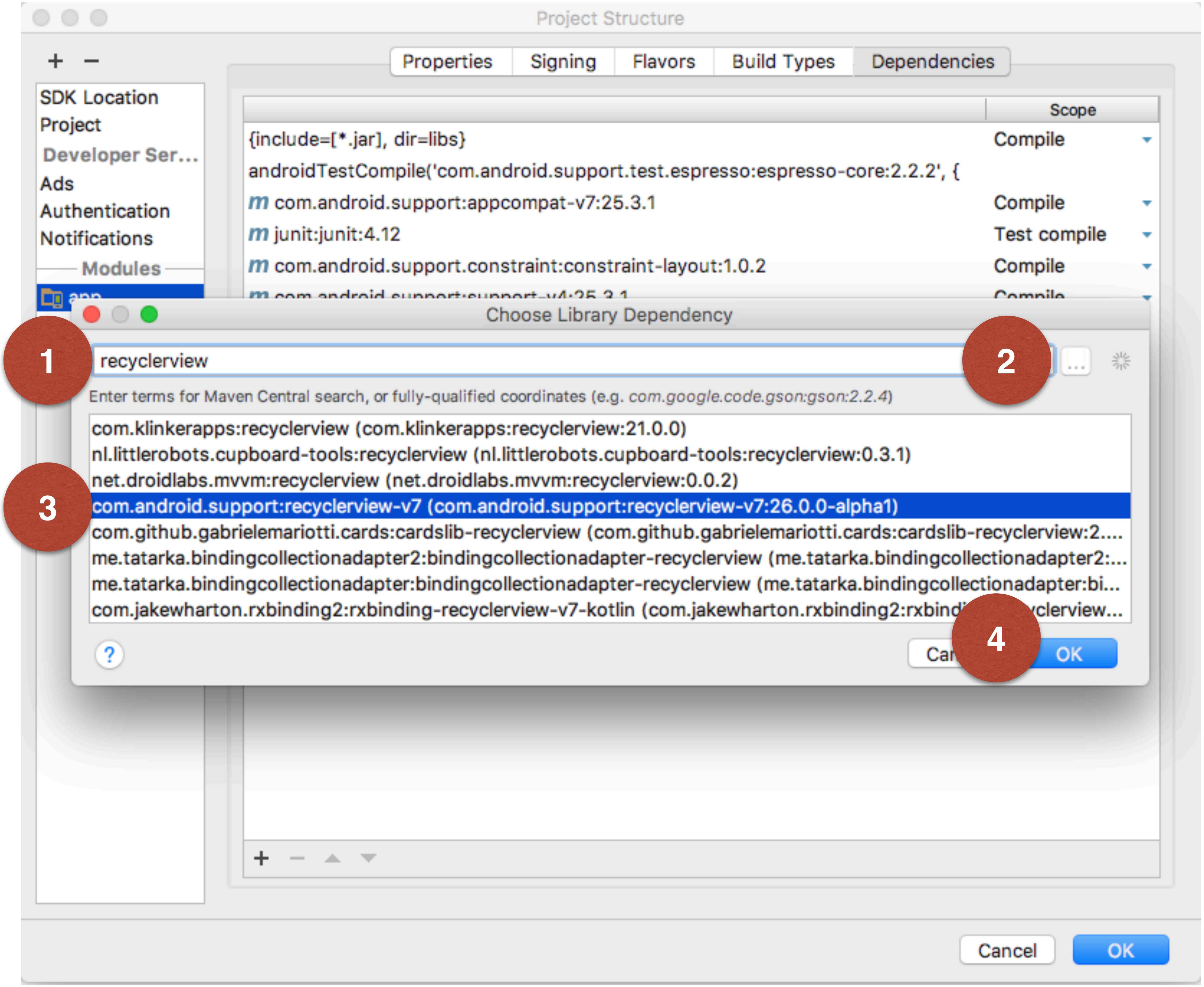
Hinzufügen der Libraries



```
n.and

sion 25
sion "25.0.2"
 {
onId "at.htl.d
sion 16
Version 25
de 1
```



	Scope
{include=[*.jar], dir=libs}	Compile
androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {	
m com.android.support:appcompat-v7:25.3.1	Compile
m junit:junit:4.12	Test compile
m com.android.support.constraint:constraint-layout:1.0.2	Compile
m com.android.support:support-v4:25.3.1	Compile

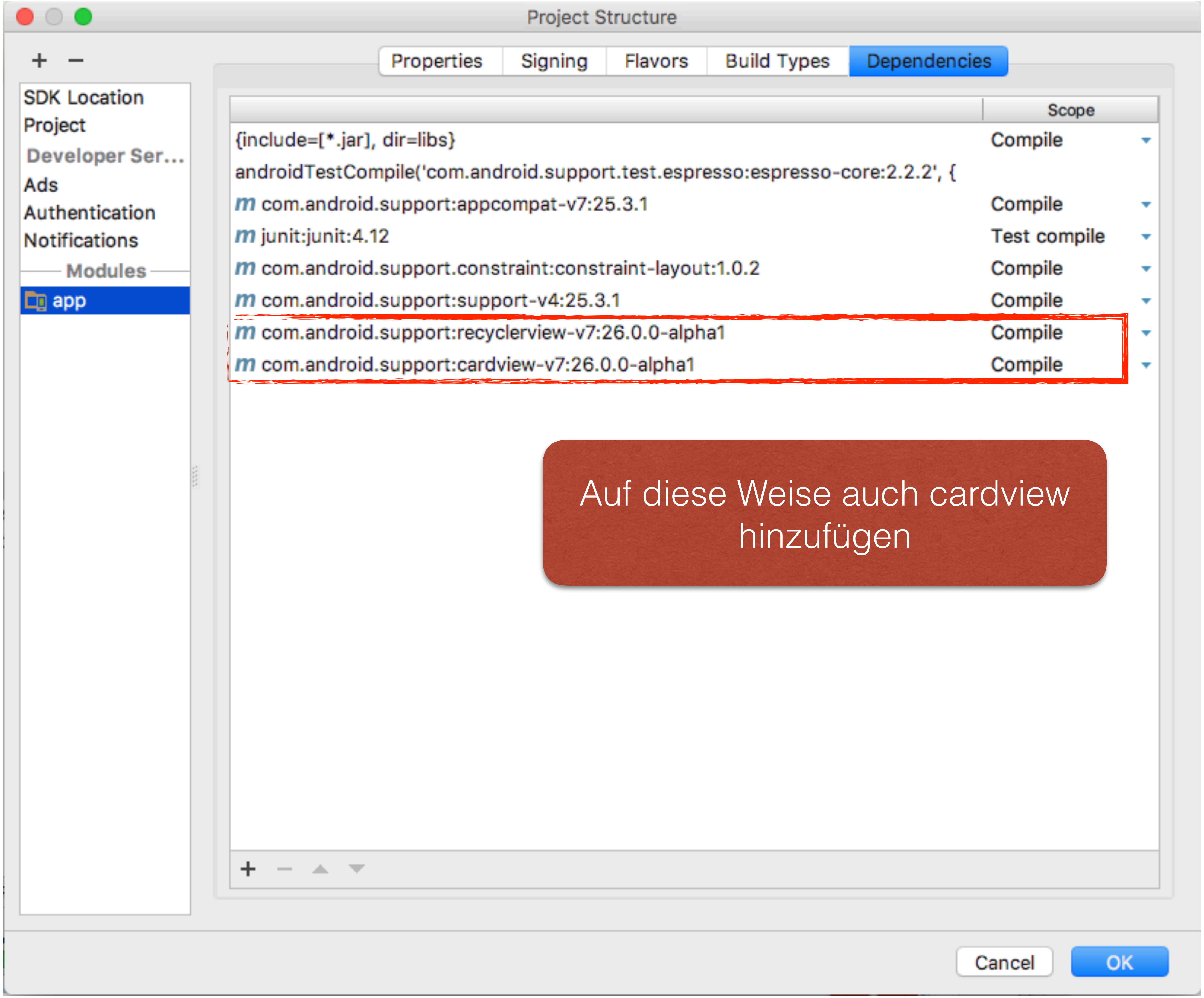


	Scope
{include=[*.jar], dir=libs}	Compile
androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {	
m com.android.support:appcompat-v7:25.3.1	Compile
m junit:junit:4.12	Test compile
m com.android.support.constraint:constraint-layout:1.0.2	Compile
m com.android.support:support-v4:25.3.1	Compile

recyclerview

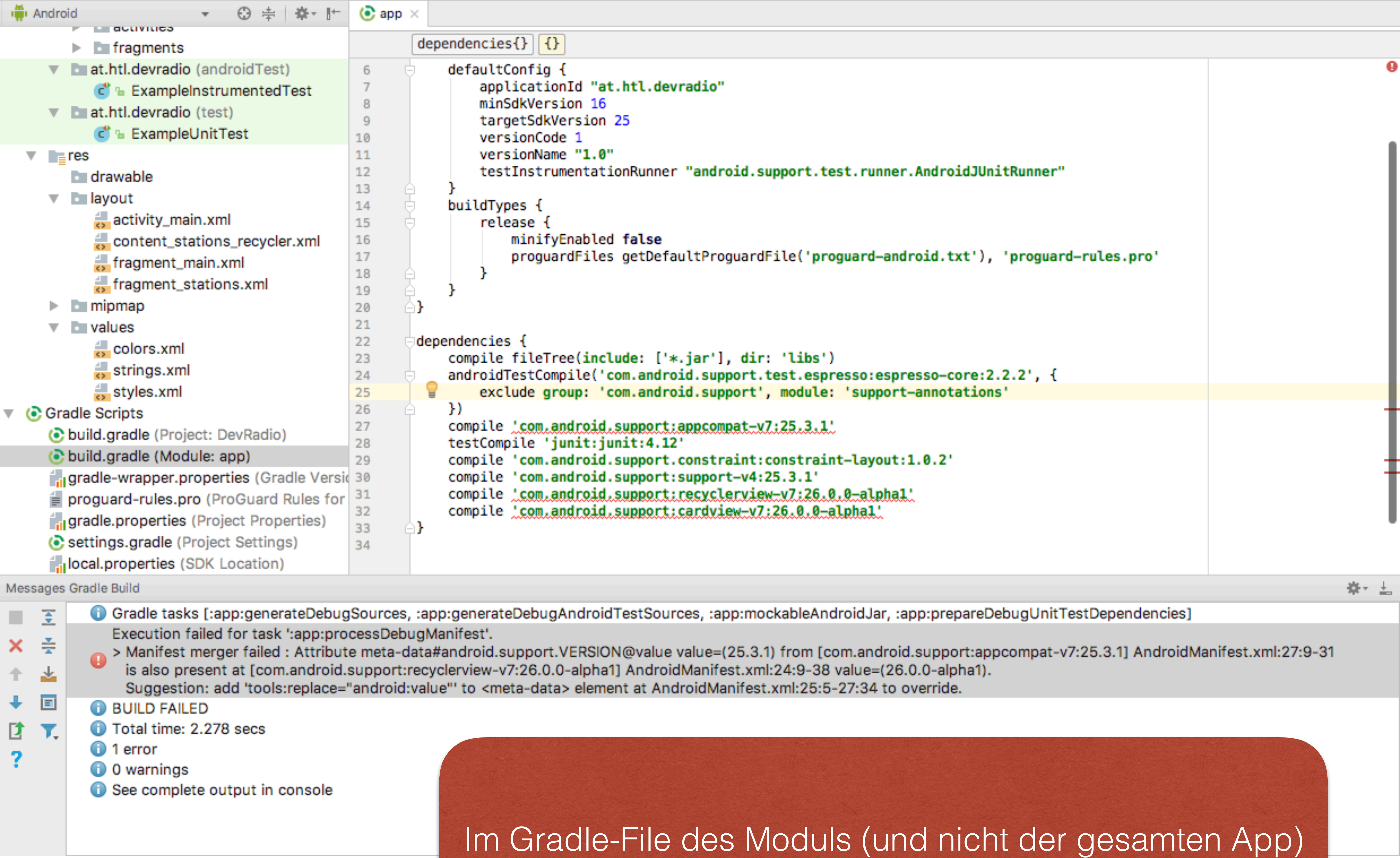
Enter terms for Maven Central search, or fully-qualified coordinates (e.g. *com.google.code.gson:gson:2.2.4*)

```
com.klinkerapps:recyclerview (com.klinkerapps:recyclerview:21.0.0)
nl.littlerobots.cupboard-tools:recyclerview (nl.littlerobots.cupboard-tools:recyclerview:0.3.1)
net.droidlabs.mvvm:recyclerview (net.droidlabs.mvvm:recyclerview:0.0.2)
com.android.support:recyclerview-v7 (com.android.support:recyclerview-v7:26.0.0-alpha1)
com.github.gabrielemariotti.cards:cardslib-recyclerview (com.github.gabrielemariotti.cards:cardslib-recyclerview:2....
me.tatarka.bindingcollectionadapter2:bindingcollectionadapter-recyclerview (me.tatarka.bindingcollectionadapter2:...
me.tatarka.bindingcollectionadapter:bindingcollectionadapter-recyclerview (me.tatarka.bindingcollectionadapter:bi...
com.jakewharton.rxbinding2:rxbinding-recyclerview-v7-kotlin (com.jakewharton.rxbinding2:rxbindi      clerview...
```



	Scope
{include=[*.jar], dir=libs}	Compile
androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {	
m com.android.support:appcompat-v7:25.3.1	Compile
m junit:junit:4.12	Test compile
m com.android.support.constraint:constraint-layout:1.0.2	Compile
m com.android.support:support-v4:25.3.1	Compile
m com.android.support:recyclerview-v7:26.0.0-alpha1	Compile
m com.android.support:cardview-v7:26.0.0-alpha1	Compile

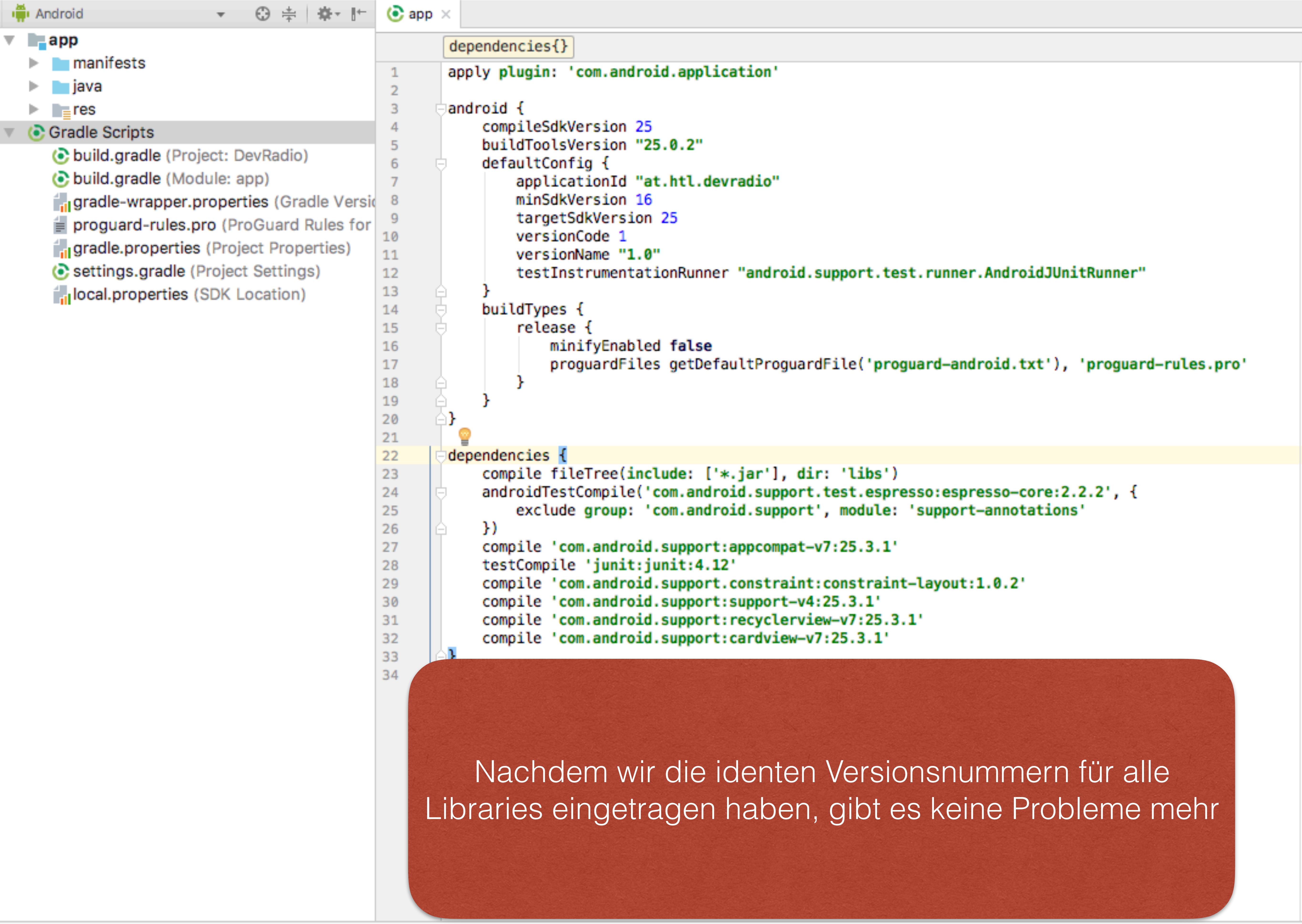
Auf diese Weise auch cardview hinzufügen


Android
app
dependencies{}
{}
activities
fragments
at.htl.devradio (androidTest)
ExampleInstrumentedTest
at.htl.devradio (test)
ExampleUnitTest
res
drawable
layout
activity_main.xml
content_stations_recycler.xml
fragment_main.xml
fragment_stations.xml
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: DevRadio)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Versi
proguard-rules.pro (ProGuard Rules for
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Messages Gradle Build

```
    defaultConfig {
        applicationId "at.htl.devradio"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'com.android.support:support-v4:25.3.1'
    compile 'com.android.support:recyclerview-v7:26.0.0-alpha1'
    compile 'com.android.support:cardview-v7:26.0.0-alpha1'
}
```

```
Gradle tasks [:app:generateDebugSources, :app:generateDebugAndroidTestSources, :app:mockableAndroidJar, :app:prepareDebugUnitTestDependencies]
Execution failed for task ':app:processDebugManifest'.
> Manifest merger failed : Attribute meta-data#android.support.VERSION@value value=(25.3.1) from [com.android.support:appcompat-v7:25.3.1] AndroidManifest.xml:27:9-31
  is also present at [com.android.support:recyclerview-v7:26.0.0-alpha1] AndroidManifest.xml:24:9-38 value=(26.0.0-alpha1).
  Suggestion: add 'tools:replace="android:value"' to <meta-data> element at AndroidManifest.xml:25:5-27:34 to override.
BUILD FAILED
Total time: 2.278 secs
1 error
0 warnings
See complete output in console
```

Im Gradle-File des Moduls (und nicht der gesamten App) werden nun die Libraries eingebunden. Allerdings erhalten wir einen Fehler.

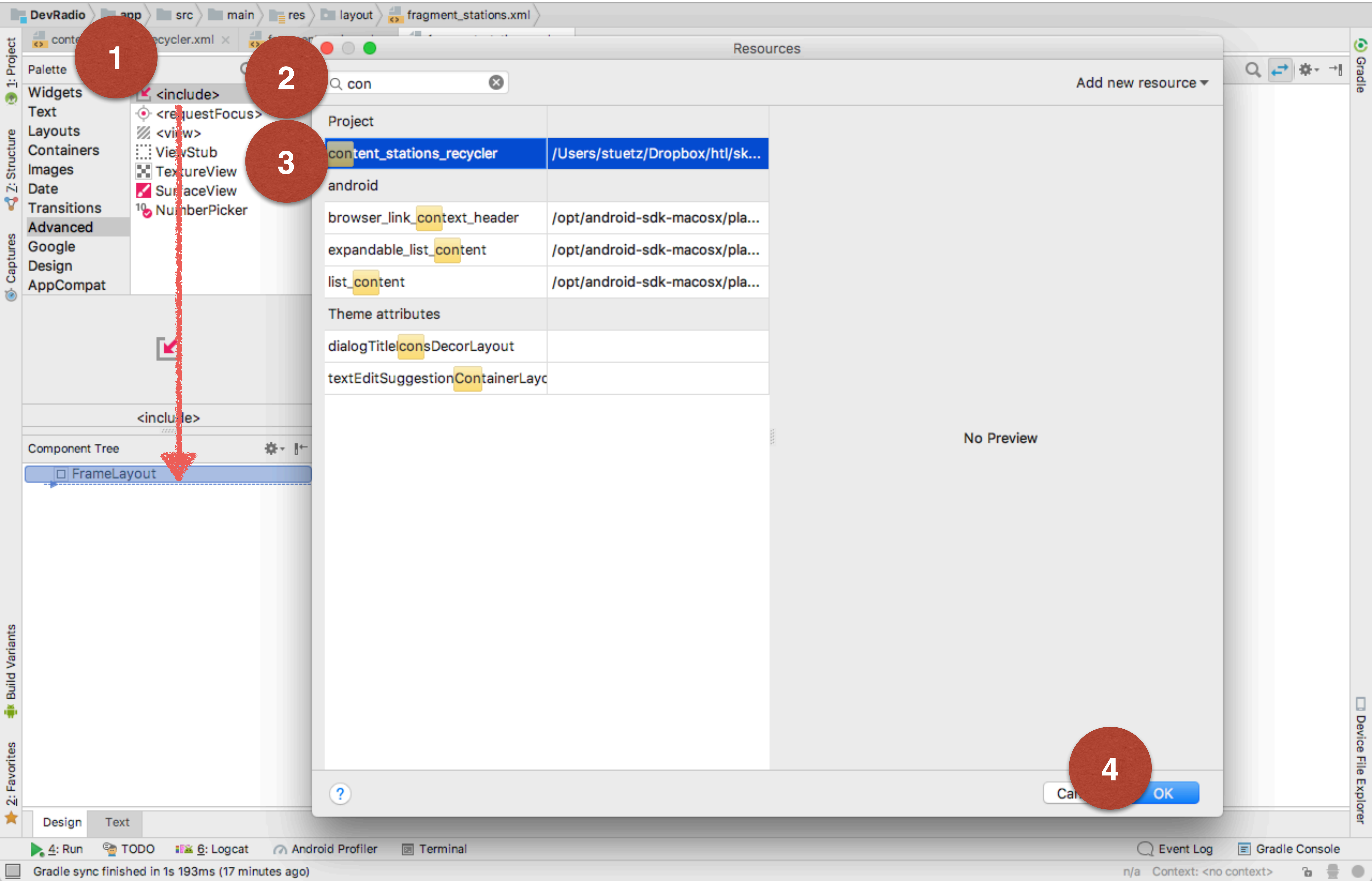

Android
app
app
manifests
java
res
Gradle Scripts
build.gradle (Project: DevRadio)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Versi
proguard-rules.pro (ProGuard Rules for
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
dependencies{}

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "at.htl.devradio"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'com.android.support:support-v4:25.3.1'
    compile 'com.android.support:recyclerview-v7:25.3.1'
    compile 'com.android.support:cardview-v7:25.3.1'
}
```

Nachdem wir die identen Versionsnummern für alle Libraries eingetragen haben, gibt es keine Probleme mehr

Vorgehensweise

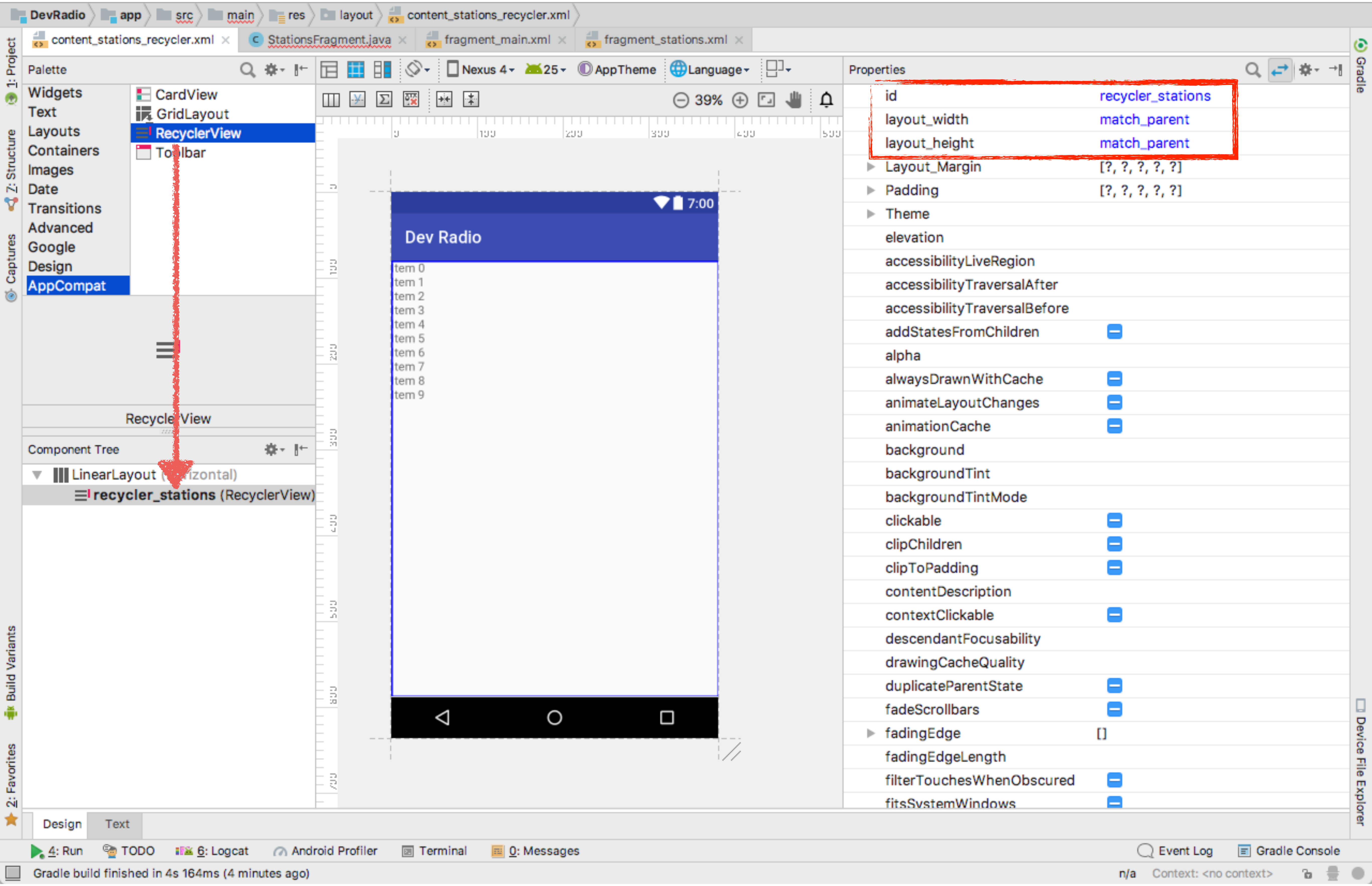
- Wir haben nun das Layout „fragment_stations.xml“
- In dieses Layout wollen wir nun die RecyclerView („content_stations_recycler.xml“) einbinden.
- Dazu verwenden wir „include“



Search: con

Project	
content_stations_recycler	/Users/stuetz/Dropbox/htl/sk...
android	
browser_link_context_header	/opt/android-sdk-macosx/pla...
expandable_list_content	/opt/android-sdk-macosx/pla...
list_content	/opt/android-sdk-macosx/pla...
Theme attributes	
dialogTitleIconsDecorLayout	
textEditSuggestionContainerLayo	

Gradle sync finished in 1s 193ms (17 minutes ago)



id	recycler_stations
layout_width	match_parent
layout_height	match_parent
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
addStatesFromChildren	▭
alpha	
alwaysDrawnWithCache	▭
animateLayoutChanges	▭
animationCache	▭
background	
backgroundTint	
backgroundTintMode	
clickable	▭
clipChildren	▭
clipToPadding	▭
contentDescription	
contextClickable	▭
descendantFocusability	
drawingCacheQuality	
duplicateParentState	▭
fadeScrollbars	▭
fadingEdge	[]
fadingEdgeLength	
filterTouchesWhenObscured	▭
fitsSystemWindows	▭

Gradle build finished in 4s 164ms (4 minutes ago)

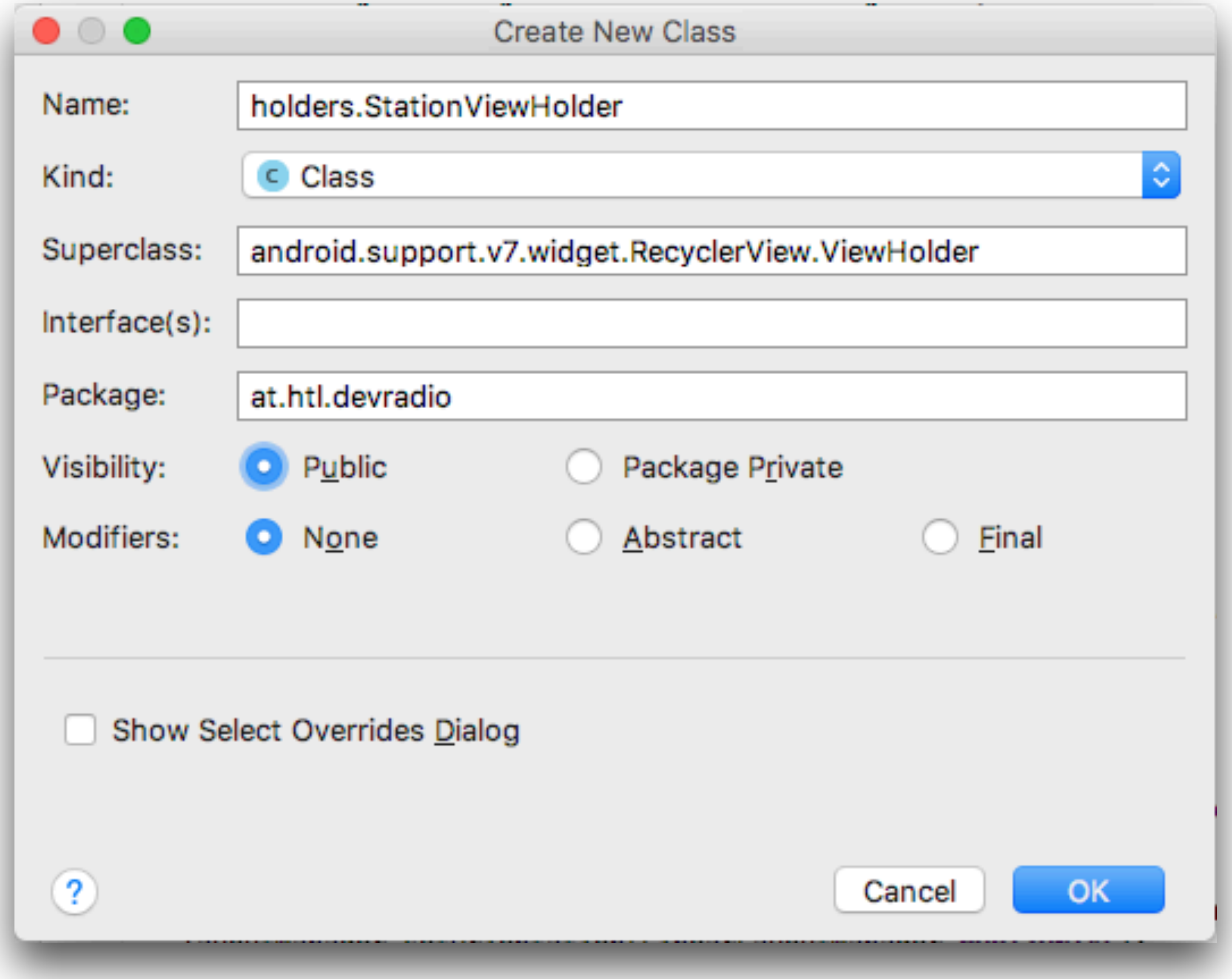
StationsFragment.java

```java
public class StationsFragment extends Fragment {

        ...

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_stations, container, false);

        RecyclerView recyclerView = (RecyclerView) v.findViewById(R.id.recycler_stations);
        recyclerView.setHasFixedSize(true);

        LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
        layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
        recyclerView.setLayoutManager(layoutManager);

        return v;
    }
}
```

setHasFixedSize(true) bedeutet, dass die einzelnen Elemente eine gleiche Größe haben (unabhängig vom Inhalt). Daher kann der Compiler Optimierungen durchführen und die Performance der Liste wird besser

Um die Orientierung der Liste auf HORIZONTAL einzustellen, muss man einen LayoutManager erstellen und diesen dann der Liste zuweisen

Erstellen des ViewHolder


Create New Class

Name: holders.StationViewHolder

Kind: Class

Superclass: android.support.v7.widget.RecyclerView.ViewHolder

Interface(s):

Package: at.htl.devradio

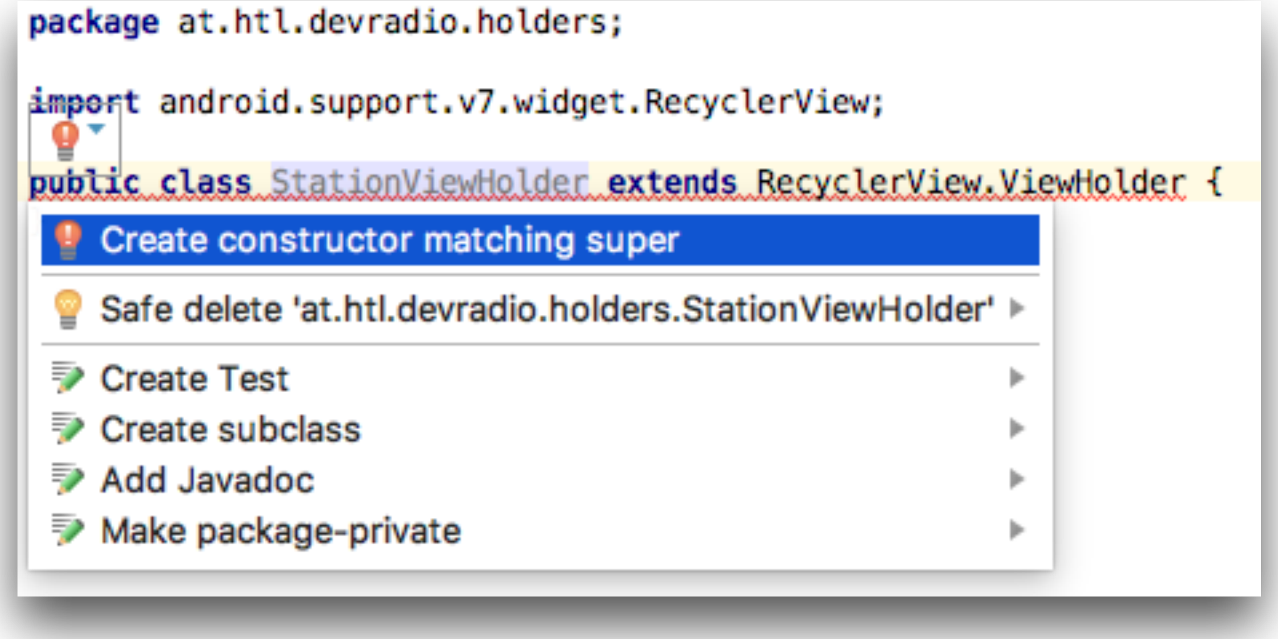
Visibility: ● Public ○ Package Private

Modifiers: ● None ○ Abstract ○ Final

□ Show Select Overrides Dialog

Ein ViewHolder - Objekt speichert die einzelnen Views jedes elements der Liste. Diese Liste wird im tag-field des Adapters gespeichert.

SationViewHolder.java

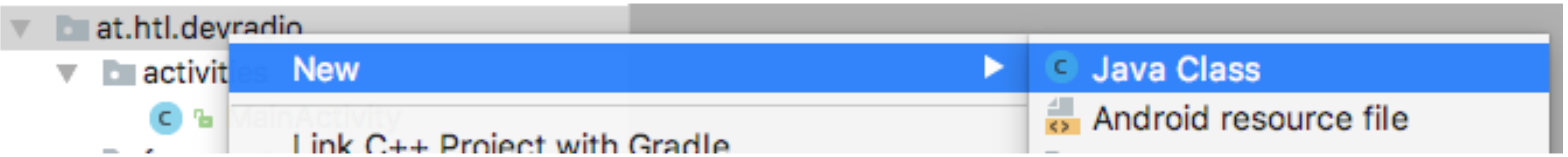


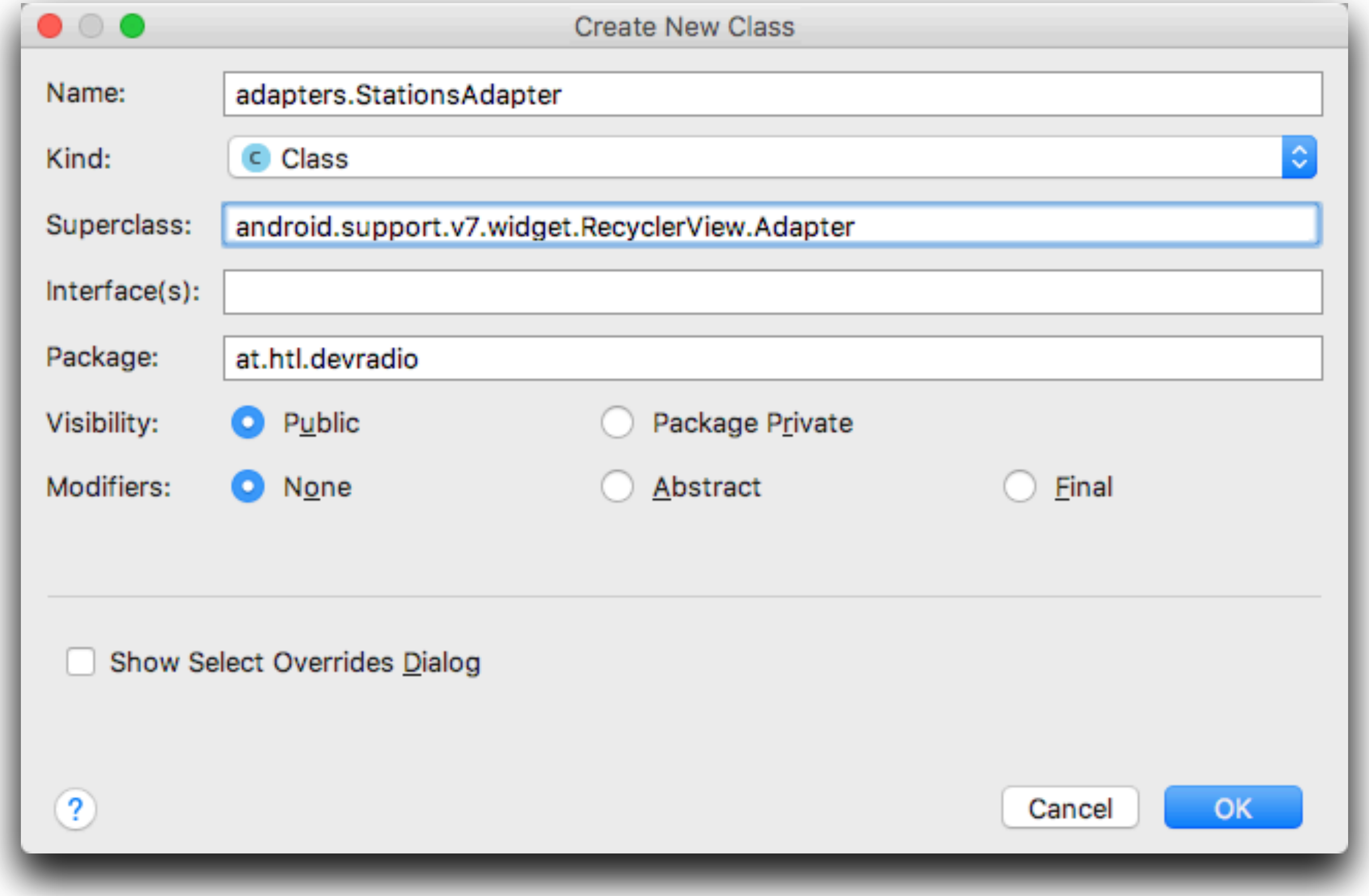
```
package at.htl.devradio.holders;

import android.support.v7.widget.RecyclerView;

public class StationViewHolder extends RecyclerView.ViewHolder {
```

Create constructor matching super
Safe delete 'at.htl.devradio.holders.StationViewHolder'
Create Test
Create subclass
Add Javadoc
Make package-private

```
package at.htl.devradio.holders;

import android.support.v7.widget.RecyclerView;
import android.view.View;

public class StationViewHolder extends RecyclerView.ViewHolder {

    public StationViewHolder(View itemView) {
        super(itemView);
    }

}
```

StationsAdapter




Create New Class

Name: adapters.StationsAdapter

Kind: Class

Superclass: android.support.v7.widget.RecyclerView.Adapter

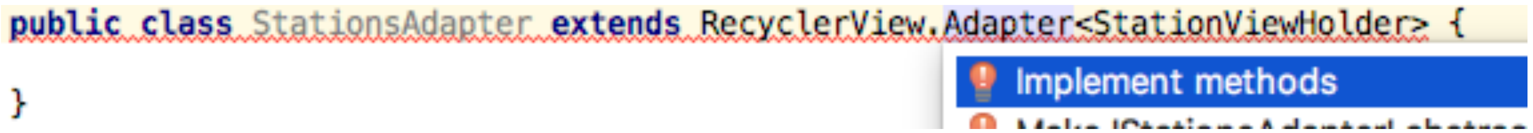
Interface(s):

Package: at.htl.devradio

Visibility: ● Public ○ Package Private

Modifiers: ● None ○ Abstract ○ Final

□ Show Select Overrides Dialog



```
public class StationsAdapter extends RecyclerView.Adapter<StationViewHolder> {

}
```

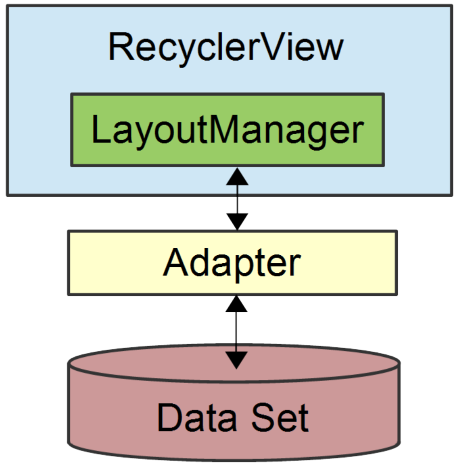
SationsAdapter.java

```
package at.htl.devradio.adapters;

import android.support.v7.widget.RecyclerView;
import android.view.ViewGroup;

import at.htl.devradio.holders.StationViewHolder;

public class StationsAdapter extends RecyclerView.Adapter<StationViewHolder> {

    @Override
    public StationViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(StationViewHolder holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }
}
```

StationsFragment.java

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {


    View v = inflater.inflate(R.layout.fragment_stations, container, false);

    RecyclerView recyclerView = (RecyclerView) v.findViewById(R.id.recycler_stations);
    recyclerView.setHasFixedSize(true);

    StationsAdapter adapter = new StationsAdapter();
    recyclerView.setAdapter(adapter);

    LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
    layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
    recyclerView.setLayoutManager(layoutManager);

    return v;
}
```

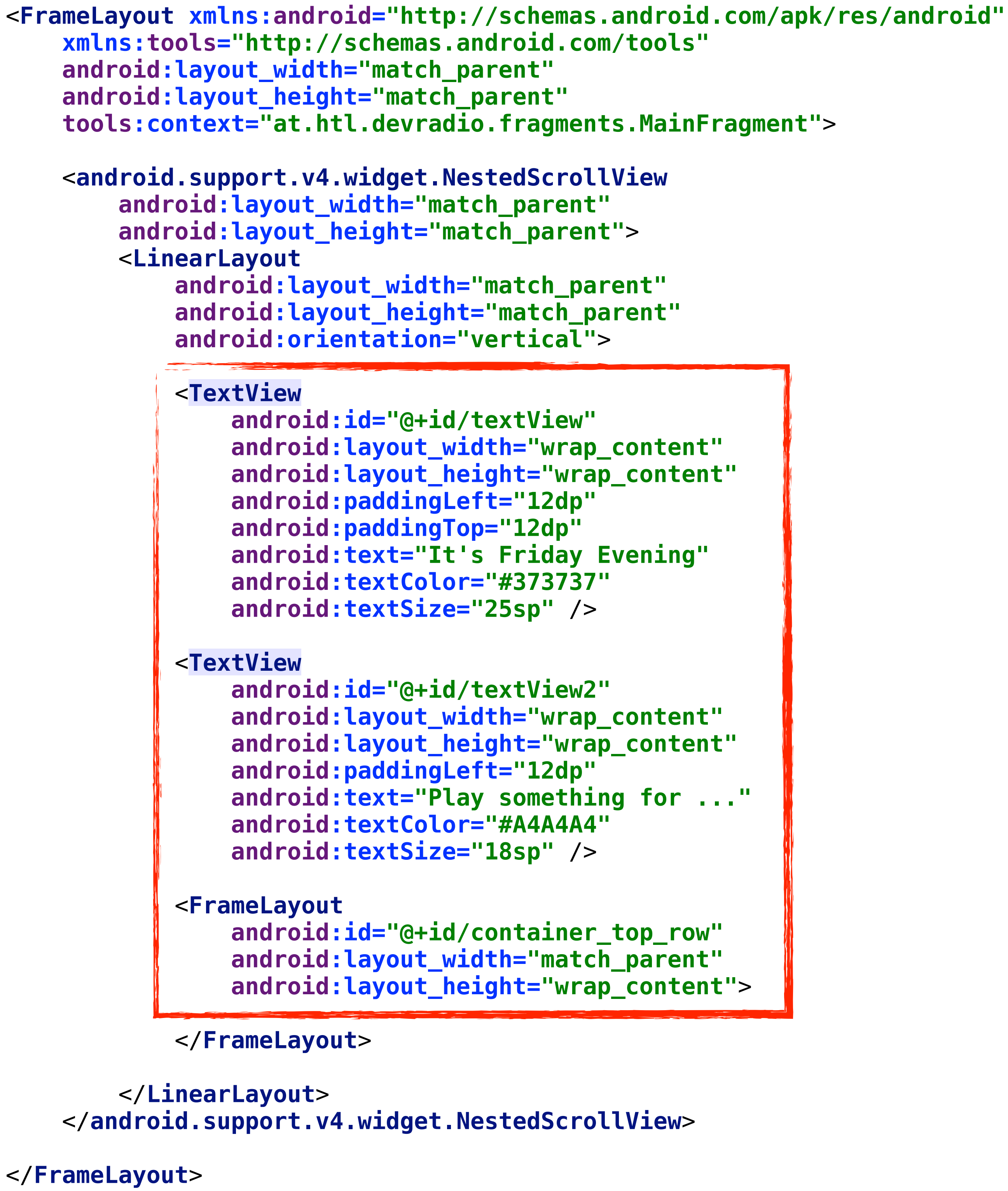
Nun wird noch das Adapter-Objekt erstellt.



Was fehlt noch?

- Wir müssen das StationsFragment im MainFragment einbinden

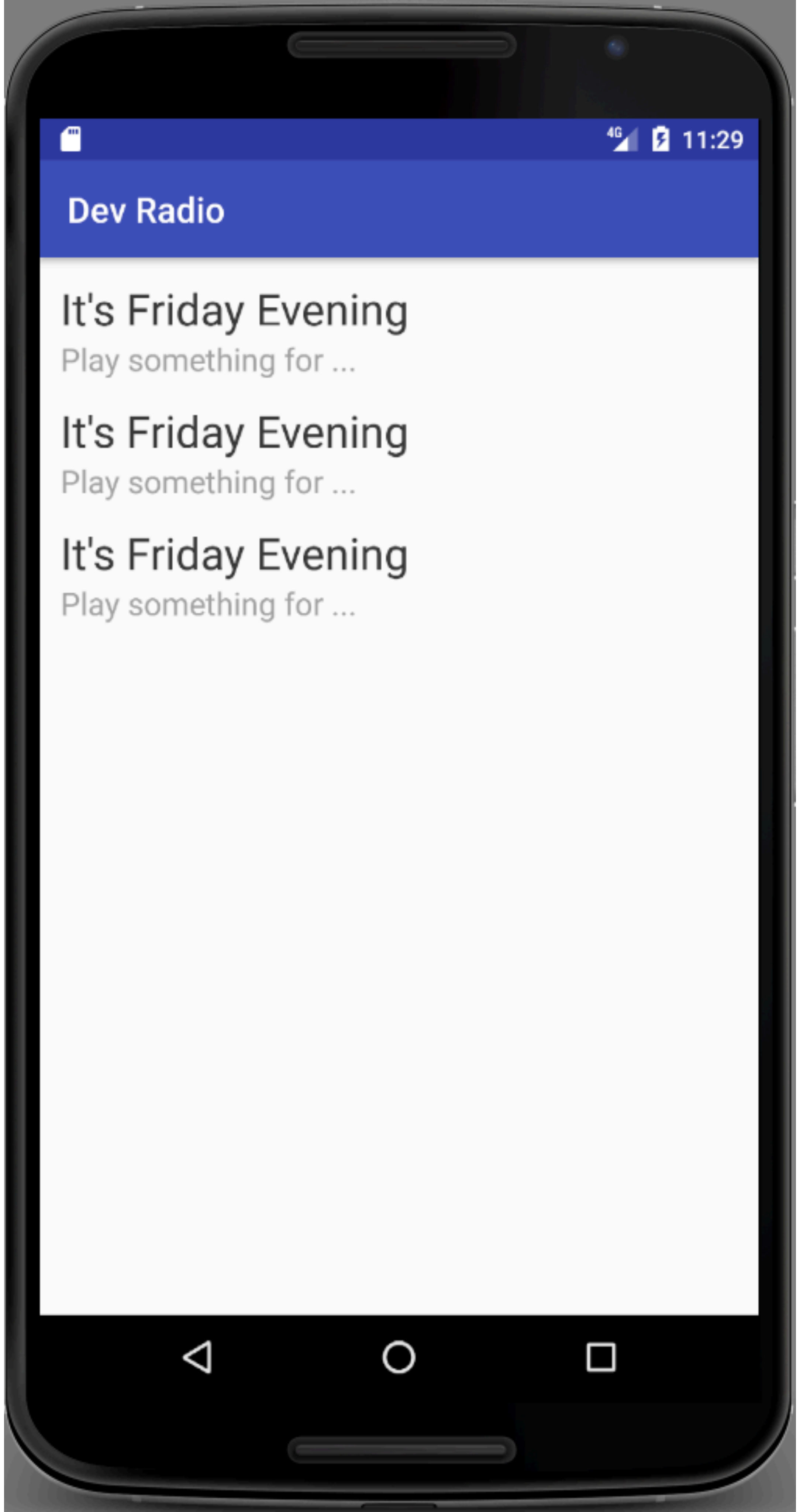
fragment_main.xml



```xml
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="at.htl.devradio.fragments.MainFragment">

    <android.support.v4.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:id="@+id/textView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingLeft="12dp"
                android:paddingTop="12dp"
                android:text="It's Friday Evening"
                android:textColor="#373737"
                android:textSize="25sp" />

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingLeft="12dp"
                android:text="Play something for ..."
                android:textColor="#A4A4A4"
                android:textSize="18sp" />

            <FrameLayout
                android:id="@+id/container_top_row"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

            </FrameLayout>

        </LinearLayout>
    </android.support.v4.widget.NestedScrollView>

</FrameLayout>
```

Jetzt kopieren wir diesen Teil noch 2 x und ändern die id auf container_middle_row und container_bottom_row

Anschließend starten wir die App



Dev Radio

It's Friday Evening
Play something for ...

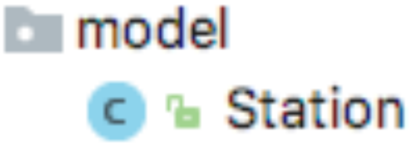
It's Friday Evening
Play something for ...

It's Friday Evening
Play something for ...

MainFragment.java

```java
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {

    View v = inflater.inflate(R.layout.fragment_main, container, false);

    FragmentManager fm = getActivity().getSupportFragmentManager();
    StationsFragment stationsFragment1;
    StationsFragment stationsFragment2;
    StationsFragment stationsFragment3;

    stationsFragment1 = StationsFragment.newInstance("blah", "blah");
    fm.beginTransaction().add(R.id.container_top_row, stationsFragment1).commit();

    stationsFragment2 = StationsFragment.newInstance("blah", "blah");
    fm.beginTransaction().add(R.id.container_middle_row, stationsFragment2).commit();

    stationsFragment3 = StationsFragment.newInstance("blah", "blah");
    fm.beginTransaction().add(R.id.container_bottom_row, stationsFragment3).commit();

    return v;
}
```

Die drei FrameLayouts in fragment_main.xml werden durch die erstellten stationFragments ersetzt.

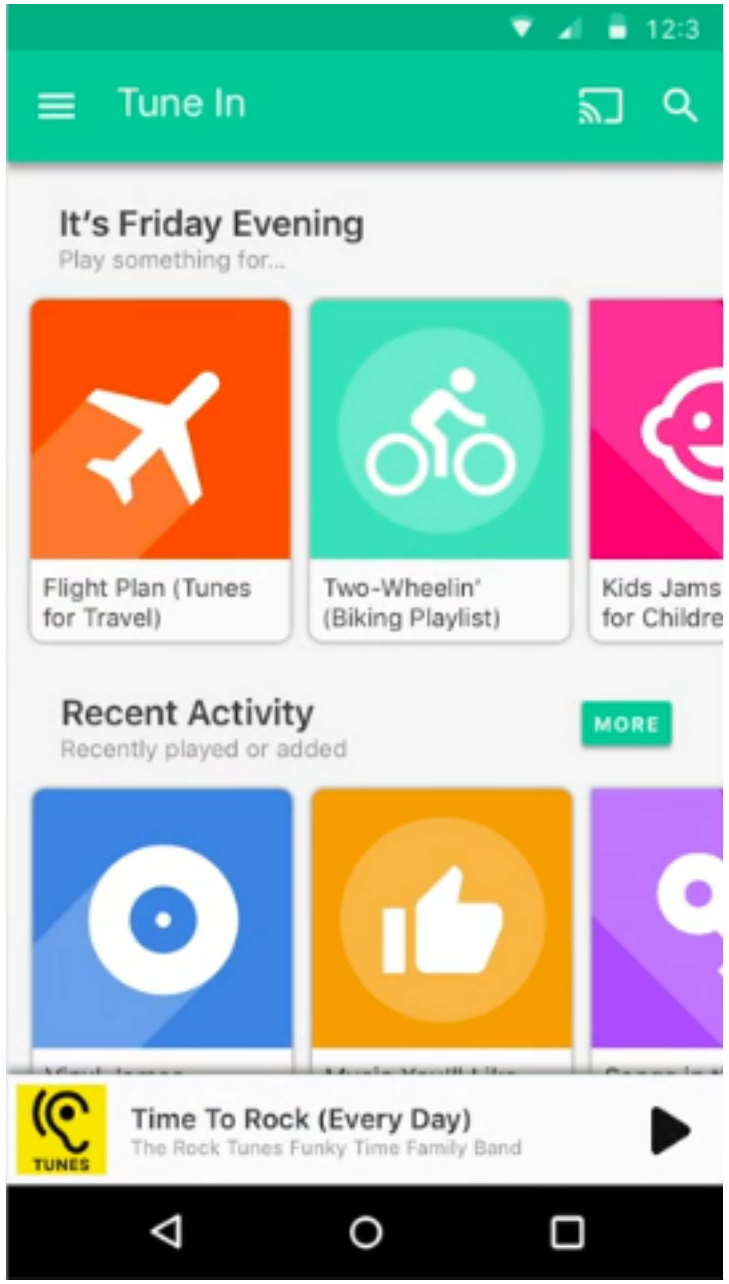


model

Station

Station.java

```java
package at.htl.devradio.model;

public class Station {

    final String DRAWABLE = "drawable/";

    private String stationTitle;
    private String imgUri;

    public Station(String stationTitle, String imgUri) {
        this.stationTitle = stationTitle;
        this.imgUri = imgUri;
    }


    public String getStationTitle() {
        return stationTitle;
    }


    public String getImgUri() {
        return imgUri;
    }

}
```



StationsFragment 1v2

```java
public class StationsFragment extends Fragment {

    private static final String ARG_STATION_TYPE = "station_type";

    public static final int STATION_TYPE_FEATURED = 0;
    public static final int STATION_TYPE_RECENT = 1;
    public static final int STATION_TYPE_PARTY = 2;

    private int stationType;

    public StationsFragment() {
        // Required empty public constructor
    }

    public static StationsFragment newInstance(int stationType) {
        StationsFragment fragment = new StationsFragment();
        Bundle args = new Bundle();
        args.putInt(ARG_STATION_TYPE, stationType);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            stationType = getArguments().getInt(ARG_STATION_TYPE);
        }
    }
```

StationsFragment 2v2

```
  @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {


        View v = inflater.inflate(R.layout.fragment_stations, container, false);

        RecyclerView recyclerView = (RecyclerView) v.findViewById(R.id.recycler_stations);
        recyclerView.setHasFixedSize(true);

        StationsAdapter adapter;

        if (stationType == STATION_TYPE_FEATURED) {

        } else if (stationType == STATION_TYPE_RECENT) {

        } else {

        }
        recyclerView.setAdapter(adapter);

        LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
        layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
        recyclerView.setLayoutManager(layoutManager);

        return v;
    }

}
```

Wir sind hier noch nicht fertig, aber zunächst müssen wir noch den Adapter fertigstellen

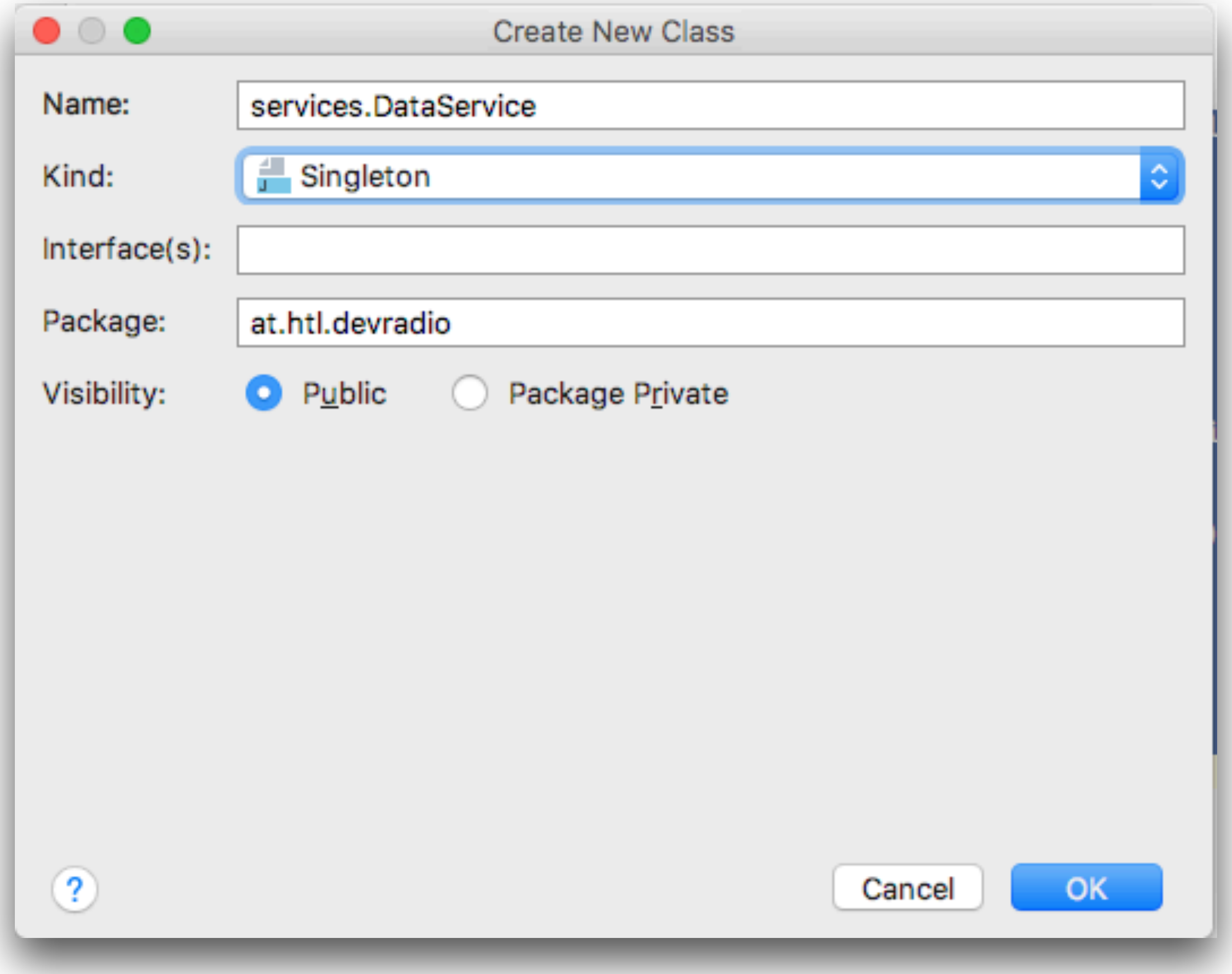
MainFragment.java

```java
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {

    View v = inflater.inflate(R.layout.fragment_main, container, false);

    FragmentManager fm = getActivity().getSupportFragmentManager();
    StationsFragment stationsFragment1;
    StationsFragment stationsFragment2;
    StationsFragment stationsFragment3;

    stationsFragment1 = StationsFragment.newInstance(StationsFragment.STATION_TYPE_FEATURED);
    fm.beginTransaction().add(R.id.container_top_row, stationsFragment1).commit();

    stationsFragment2 = StationsFragment.newInstance(StationsFragment.STATION_TYPE_RECENT);
    fm.beginTransaction().add(R.id.container_middle_row, stationsFragment2).commit();

    stationsFragment3 = StationsFragment.newInstance(StationsFragment.STATION_TYPE_PARTY);
    fm.beginTransaction().add(R.id.container_bottom_row, stationsFragment3).commit();

    return v;
}
```

StationsAdapter.java

```java
public class StationsAdapter extends RecyclerView.Adapter<StationViewHolder> {

    private ArrayList<Station> stations;

    public StationsAdapter(ArrayList<Station> stations) {
        this.stations = stations;
    }

    @Override
    public StationViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(StationViewHolder holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }
}
```

DataService.java



Create New Class

Name: services.DataService

Kind: Singleton

Interface(s):

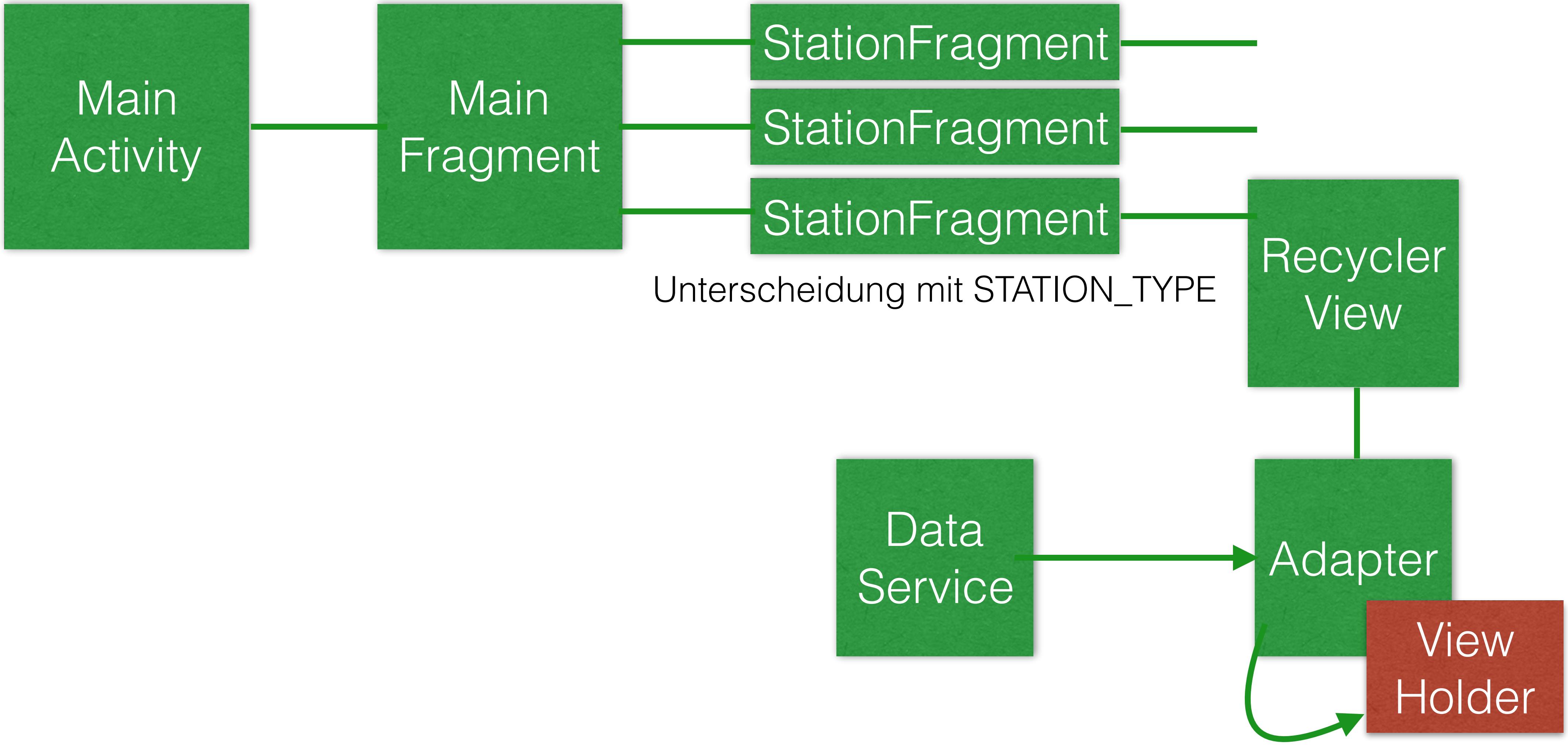
Package: at.htl.devradio

Visibility: ● Public ○ Package Private

DataService.java

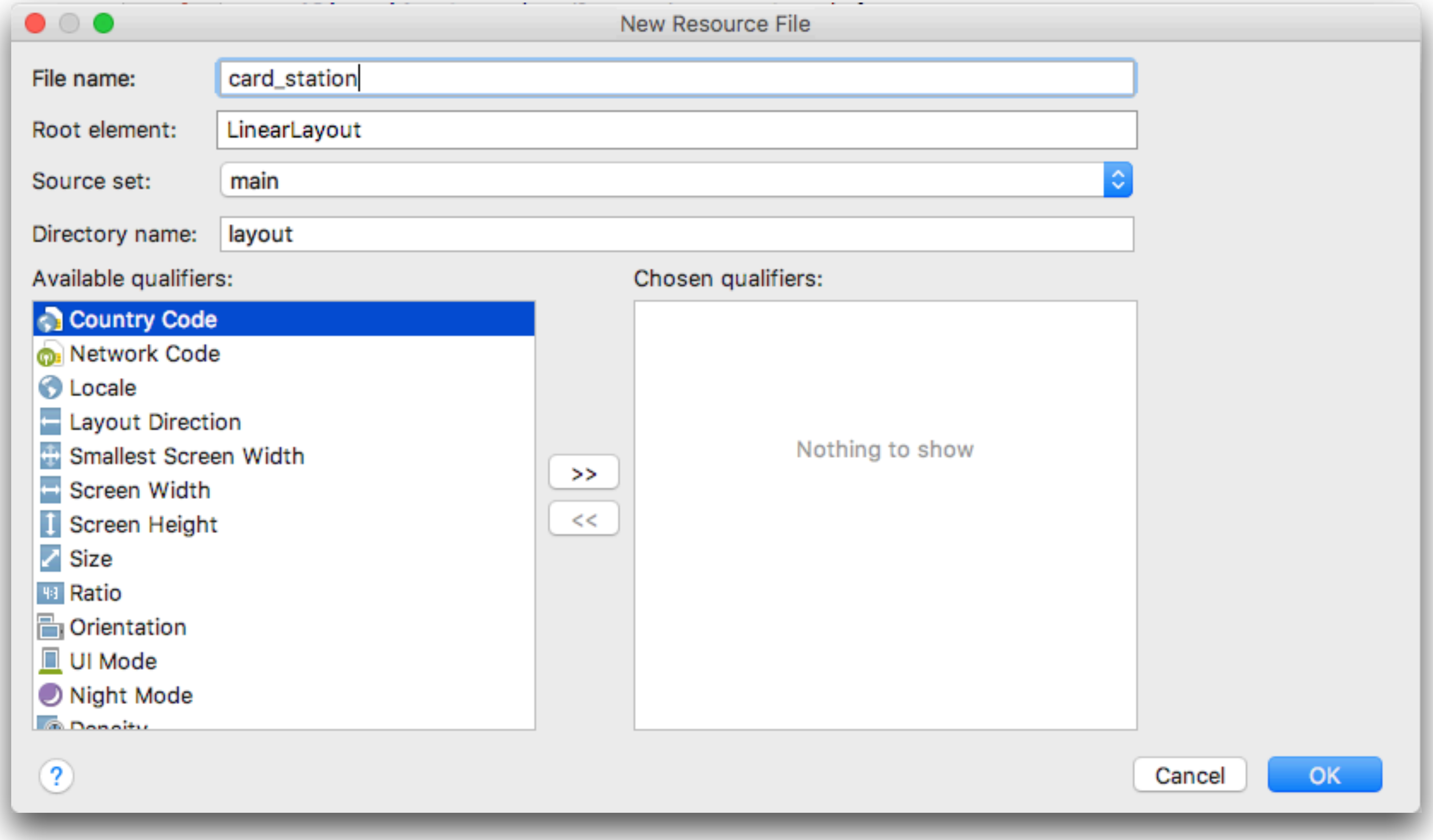
```java
public class DataService {
    private static final DataService ourInstance = new DataService();

    public static DataService getInstance() {
        return ourInstance;
    }

    private DataService() {
    }

    public ArrayList<Station> getFeaturedStations() {
        // Pretend we just downloaded featured stations from the Internet

        ArrayList<Station> list = new ArrayList<>();
        list.add(new Station("flight Plan (Tunes for Travel)", "flightplanmusic"));
        list.add(new Station("Two-Wheelin' (Biking Playlist)", "bicyclemusic"));
        list.add(new Station("Kids Jams (Music for Children", "kidsmusic"));

        return list;
    }

    public ArrayList<Station> getRecentStations() {
        // Pretend we just downloaded featured stations from the Internet

        ArrayList<Station> list = new ArrayList<>();
        return list;
    }

    public ArrayList<Station> getPartyStations() {
        // Pretend we just downloaded featured stations from the Internet

        ArrayList<Station> list = new ArrayList<>();
        return list;
    }
}
```

Zwischenstand



Die ViewHolder-Objekte, welche die Daten anzeigen, müssen noch erstellt werden.

Neues Resource-File erstellen



File name: card_station

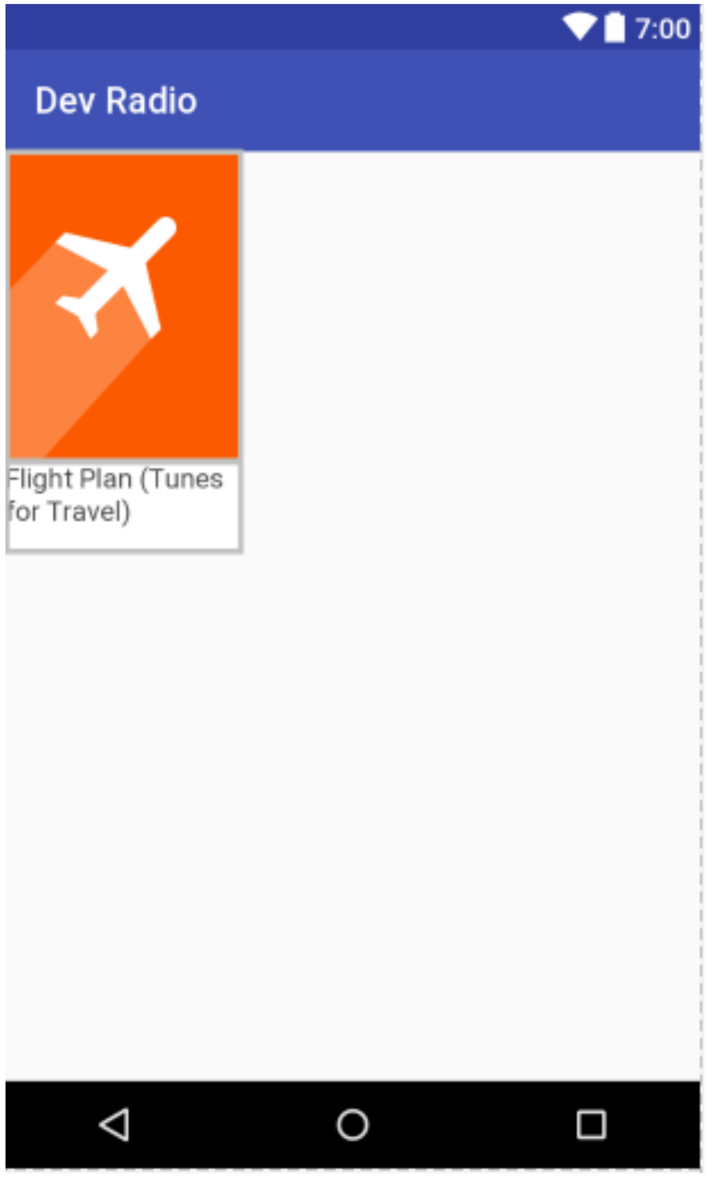
Root element: LinearLayout

Source set: main

Directory name: layout

card_station.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/card_station"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content">


    <LinearLayout
        android:id="@+id/main_image"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/main_image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/flightplanmusic" />

        <TextView
            android:id="@+id/main_text"
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:lines=„2"
           android:layout_marginLeft="5dp"
           android:layout_marginRight="5dp"
            android:text="Flight Plan (Tunes for Travel)"
            android:textColor="#4A4A4A"
            android:textSize="15sp" />
    </LinearLayout>
</android.support.v7.widget.CardView>
```



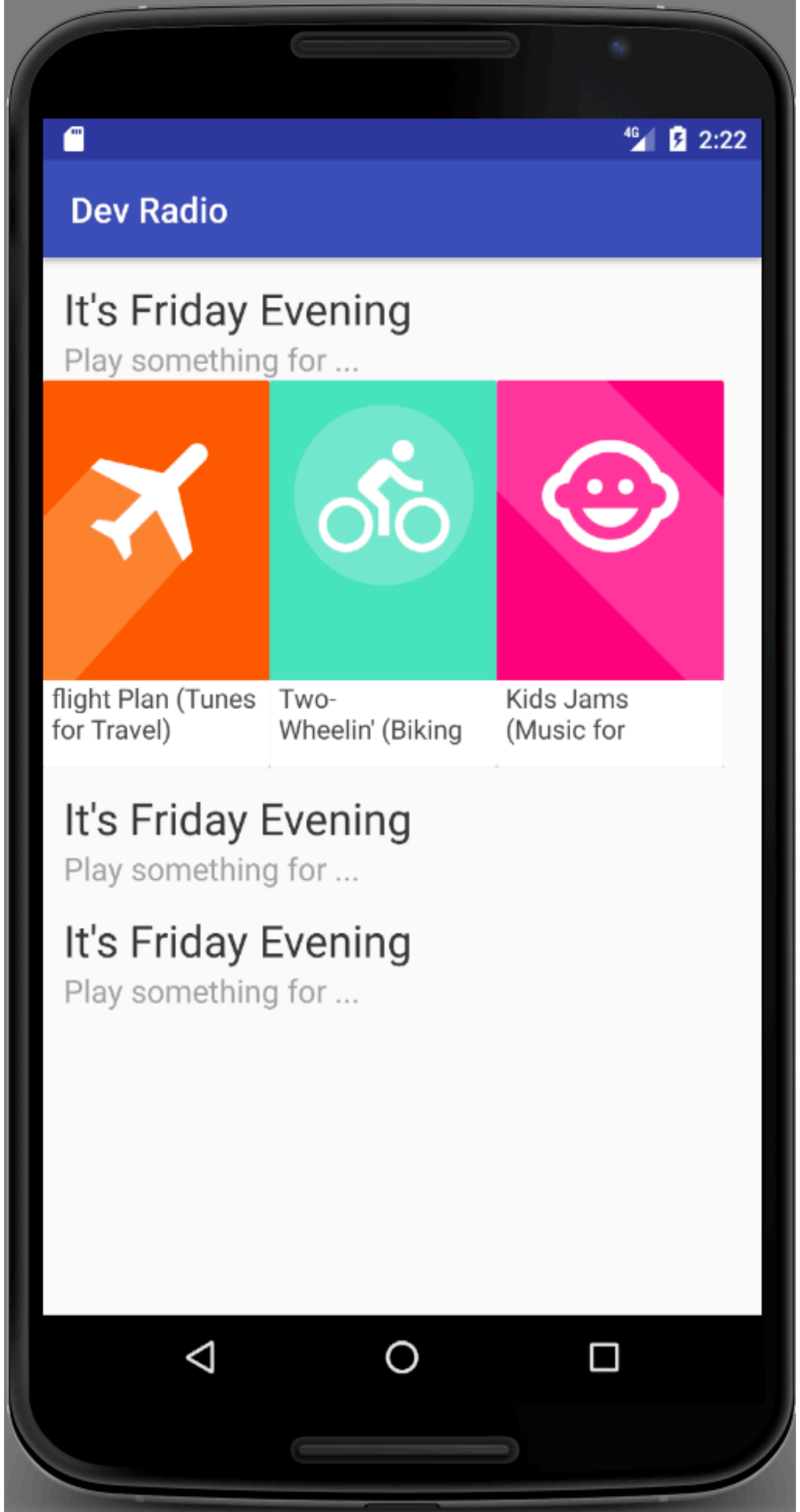
StationViewHolder.java

```java
package at.htl.devradio.holders;

import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import at.htl.devradio.R;
import at.htl.devradio.model.Station;

public class StationViewHolder extends RecyclerView.ViewHolder {

    private ImageView mainImage;
    private TextView titleTextView;

    public StationViewHolder(View itemView) {
        super(itemView);

        this.mainImage = (ImageView) itemView.findViewById(R.id.main_image);
        this.titleTextView = (TextView) itemView.findViewById(R.id.main_text);
    }

    public void updateUI(Station station) {
        String uri = station.getImgUri();
        int resource = mainImage
                .getResources()
                .getIdentifier(uri, null, mainImage.getContext().getPackageName());
        mainImage.setImageResource(resource);

        titleTextView.setText(station.getStationTitle());
    }

}
```

StationsAdapter.java

```java
public class StationsAdapter extends RecyclerView.Adapter<StationViewHolder> {

    private ArrayList<Station> stations;

    public StationsAdapter(ArrayList<Station> stations) {
        this.stations = stations;
    }

    @Override
    public StationViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View stationCard = LayoutInflater
                .from(parent.getContext())
                .inflate(R.layout.card_station, parent, false);
        return new StationViewHolder(stationCard);
    }

    @Override
    public void onBindViewHolder(StationViewHolder holder, int position) {
        Station station = stations.get(position);
        holder.updateUI(station);
    }

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

Nun wollen wir Zwischenräume zwischen den Cards erhalten



Dev Radio

It's Friday Evening

Play something for ...

It's Friday Evening

Play something for ...

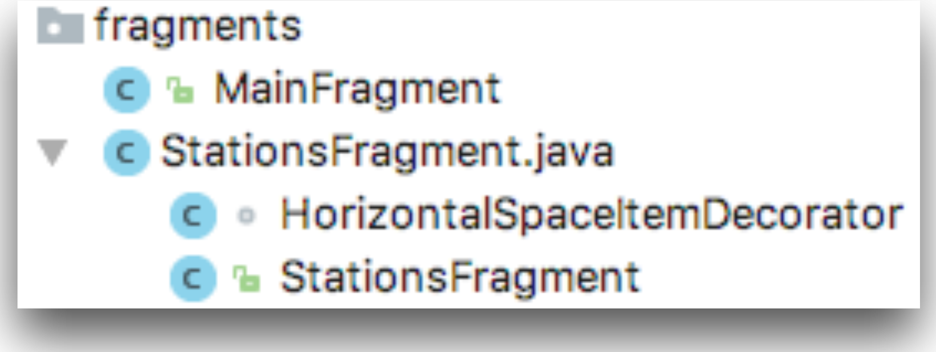
It's Friday Evening

Play something for ...

ev. die drei icon-Files aus den drawables löschen

StationsFragment.java

```
public class StationsFragment extends Fragment {

  ...

}
```



fragments
MainFragment
StationsFragment.java
HorizontalSpaceItemDecorator
StationsFragment

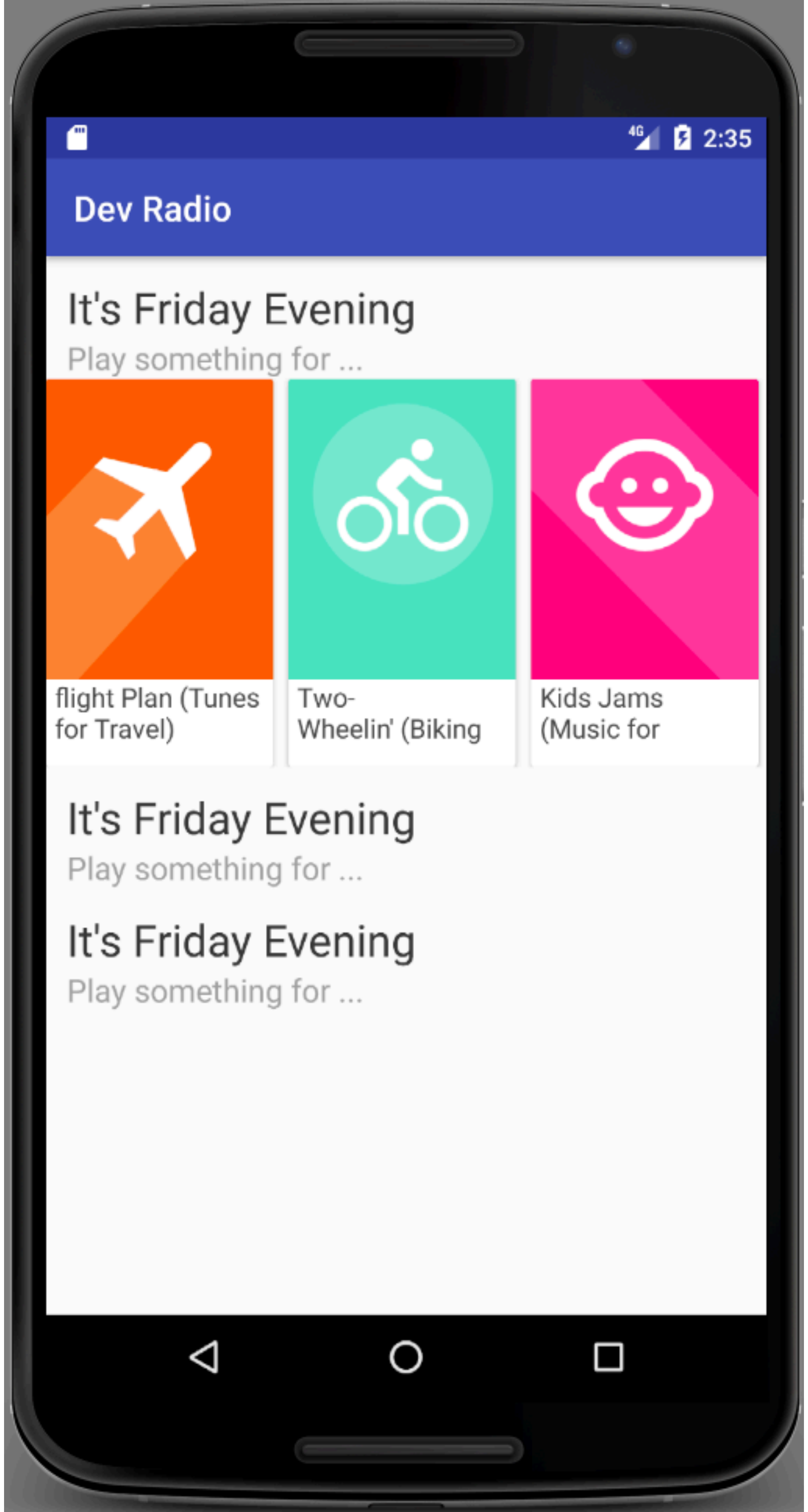
```
class HorizontalSpaceItemDecorator extends RecyclerView.ItemDecoration {

    private final int spacer;

    public HorizontalSpaceItemDecorator(int spacer) {
        this.spacer = spacer;
    }

    @Override
    public void getItemOffsets(Rect outRect, View view, RecyclerView parent, RecyclerView.State 
state) {
        super.getItemOffsets(outRect, view, parent, state);
        outRect.right = spacer;
    }
}
```

mit ^O getItemOffsets() überschreiben

StationsFragment.java

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {


    View v = inflater.inflate(R.layout.fragment_stations, container, false);

    RecyclerView recyclerView = (RecyclerView) v.findViewById(R.id.recycler_stations);
    recyclerView.setHasFixedSize(true);

    StationsAdapter adapter;

    if (stationType == STATION_TYPE_FEATURED) {
        adapter = new StationsAdapter(DataService.getInstance().getFeaturedStations());
    } else if (stationType == STATION_TYPE_RECENT) {
        adapter = new StationsAdapter(DataService.getInstance().getRecentStations());
    } else {
        adapter = new StationsAdapter(DataService.getInstance().getPartyStations());
    }

    recyclerView.addItemDecoration(new HorizontalSpaceItemDecorator(30));

    recyclerView.setAdapter(adapter);

    LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
    layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
    recyclerView.setLayoutManager(layoutManager);

    return v;
}
```

Die Zwischenräume sind nun verfügbar



Dev Radio

It's Friday Evening

Play something for ...

flight Plan (Tunes for Travel)

Two-Wheelin' (Biking

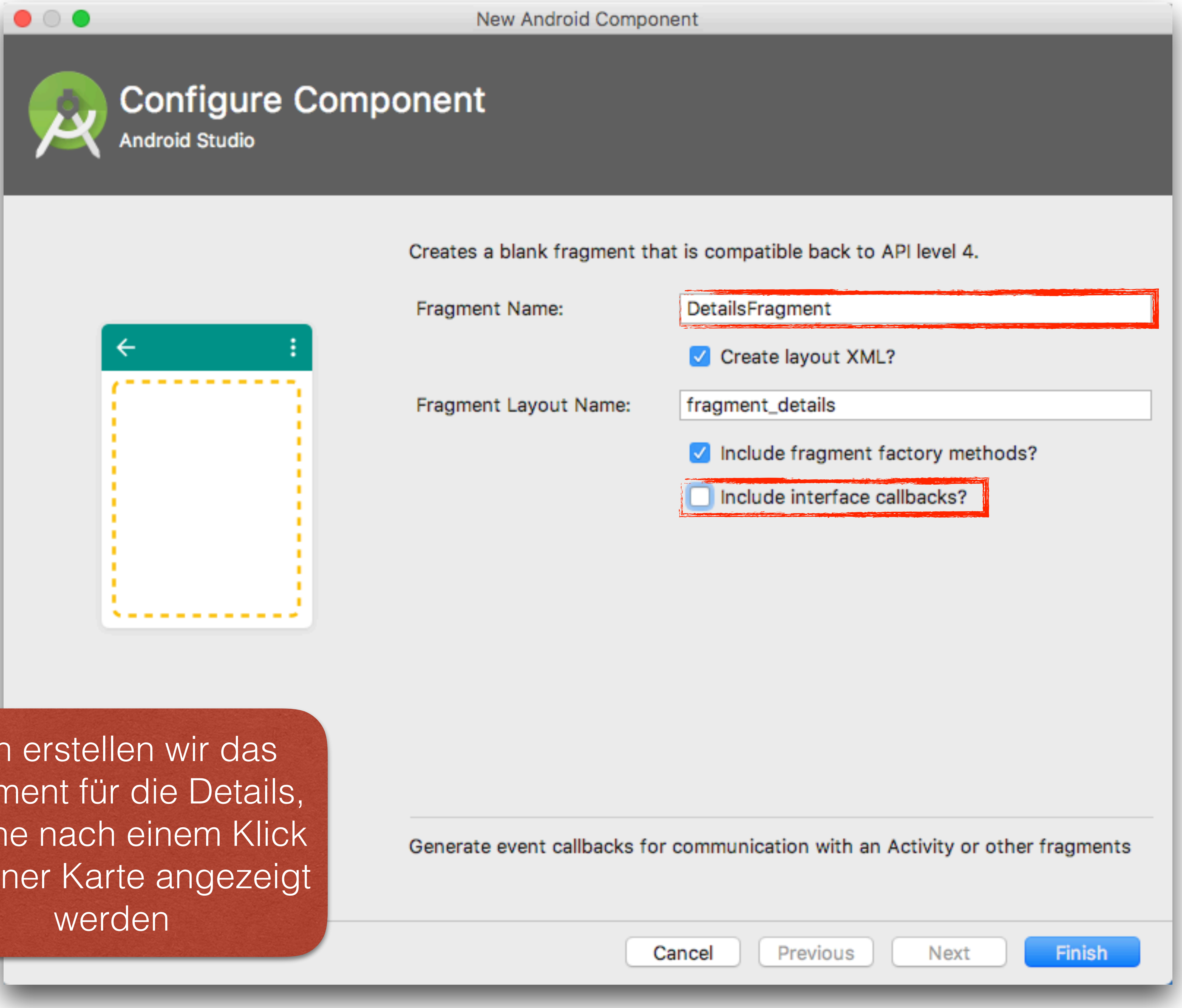
Kids Jams (Music for

It's Friday Evening

Play something for ...

It's Friday Evening

Play something for ...



New Android Component

Configure Component

Android Studio

Creates a blank fragment that is compatible back to API level 4.

Fragment Name: DetailsFragment

☑ Create layout XML?

Fragment Layout Name: fragment_details

☑ Include fragment factory methods?

□ Include interface callbacks?

Generate event callbacks for communication with an Activity or other fragments

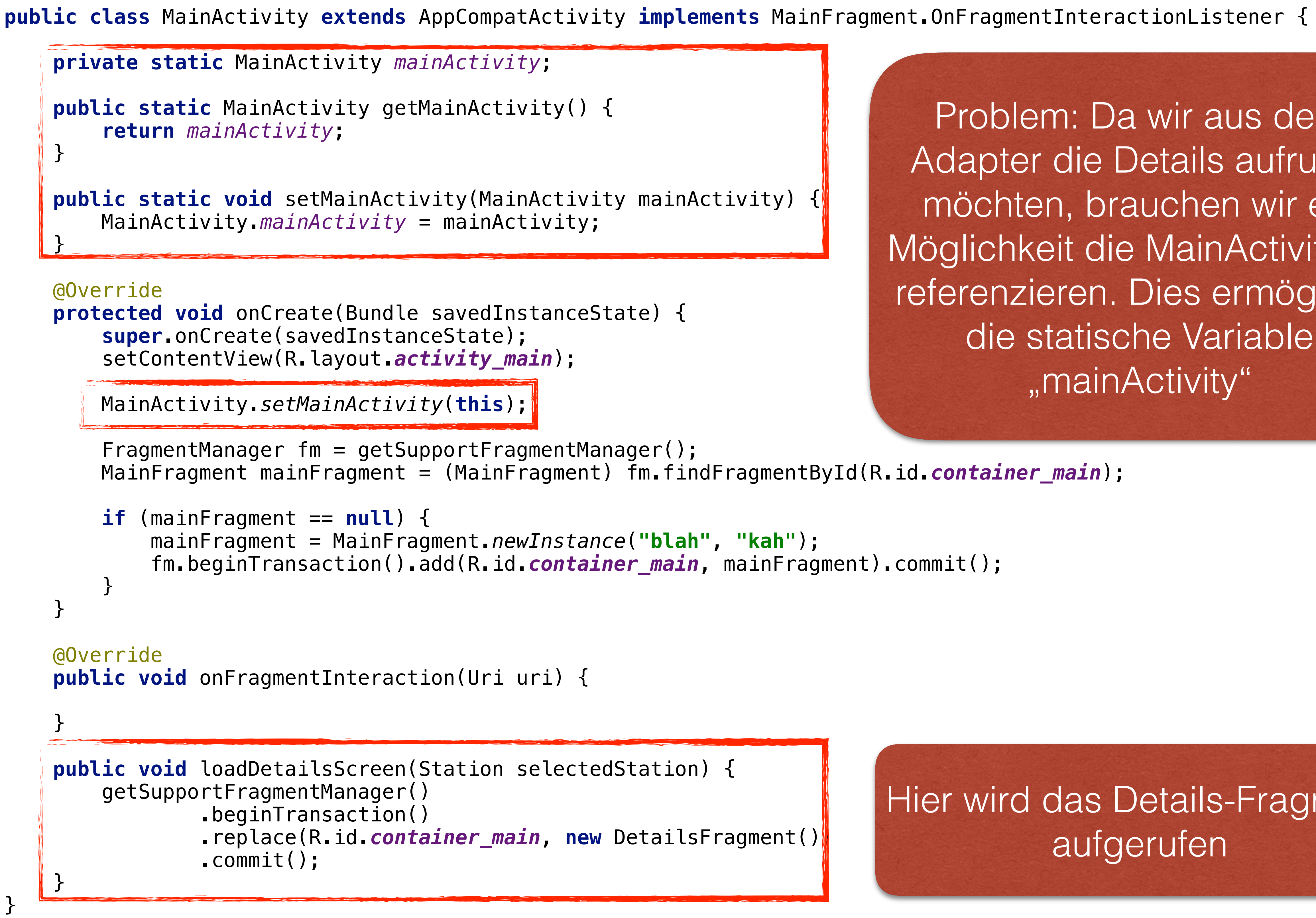
Nun erstellen wir das Fragment für die Details, welche nach einem Klick auf einer Karte angezeigt werden

StationsAdapter.java

```java
public class StationsAdapter extends RecyclerView.Adapter<StationViewHolder> {

    private ArrayList<Station> stations;

    public StationsAdapter(ArrayList<Station> stations) {
        this.stations = stations;
    }

    @Override
    public StationViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View stationCard = LayoutInflater
                .from(parent.getContext())
                .inflate(R.layout.card_station, parent, false);
        return new StationViewHolder(stationCard);
    }

    @Override
    public void onBindViewHolder(StationViewHolder holder, final int position) {
        Station station = stations.get(position);
        holder.updateUI(station);

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // load the detail screen
            }
        });
    }

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

Wir fangen einen Klick auf die Karte ab und öffnen die Details.
Problem: Wir können nicht auf die Fragments zugreifen, da wir uns im Adapter befinden, der nichts von der UI wissen darf!!!

MainActivity.java



```
public class MainActivity extends AppCompatActivity implements MainFragment.OnFragmentInteractionListener {

    private static MainActivity mainActivity;

    public static MainActivity getMainActivity() {
        return mainActivity;
    }

    public static void setMainActivity(MainActivity mainActivity) {
        MainActivity.mainActivity = mainActivity;
    }

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

        MainActivity.setMainActivity(this);

        FragmentManager fm = getSupportFragmentManager();
        MainFragment mainFragment = (MainFragment) fm.findFragmentById(R.id.container_main);

        if (mainFragment == null) {
            mainFragment = MainFragment.newInstance("blah", "kah");
            fm.beginTransaction().add(R.id.container_main, mainFragment).commit();
        }
    }

    @Override
    public void onFragmentInteraction(Uri uri) {

    }

    public void loadDetailsScreen(Station selectedStation) {
        getSupportFragmentManager()
                .beginTransaction()
                .replace(R.id.container_main, new DetailsFragment())
                .commit();
    }
}
```

Problem: Da wir aus dem Adapter die Details aufrufen möchten, brauchen wir ein Möglichkeit die MainActivity zu referenzieren. Dies ermöglicht die statische Variable „mainActivity“

Hier wird das Details-Fragment aufgerufen

StationsAdapter.java

Die station muss final sein, damit der onClickListener darauf zugreifen kann

```
@Override
public void onBindViewHolder(StationViewHolder holder, final int position) {
    final Station station = stations.get(position);
    holder.updateUI(station);

    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // load the detail screen
            MainActivity.getMainActivity().loadDetailsScreen(station);
        }
    });
}
```

Aufgabe

- Wie muss beiliegender Code verändert werden, damit man durch Drücken der Back-Taste wieder zum vorherigen Fragment zurückkommt?

```
getSupportFragmentManager()
             .beginTransaction()
             .replace(R.id.container_main, new DetailsFragment())
             .commit();
```

Lösung

```
getSupportFragmentManager()
        .beginTransaction()
        .replace(R.id.container_main, new DetailsFragment())
        .addToBackStack(null)
        .commit();
```

Noch Fragen?