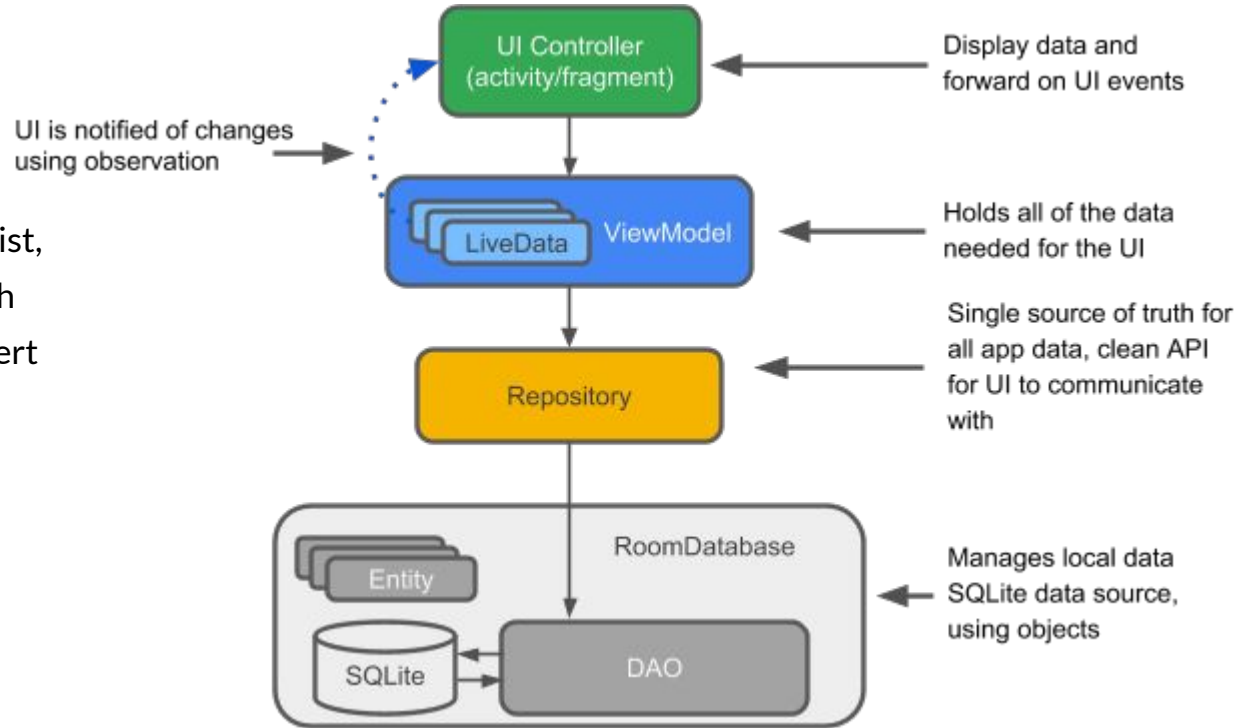




Room with Navigation in Kotlin

Architektur

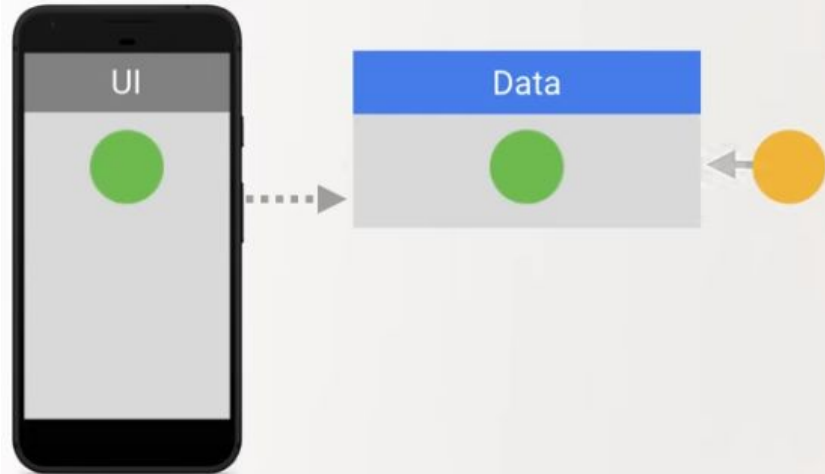
Der Vorteil des ViewModels ist, dass es die Daten behält, auch wenn die Activity neu generiert wird. (z.B.: Rotieren des Telefons)



Warum LiveData

LiveData bekommt immer eine Benachrichtigung, wenn sich was in der Datenbank ändert (vergleichbar mit einer ObservableList). Das Beispiel wäre noch mit Data Binding erweiterbar, um die Daten direkt in der App anzuzeigen.

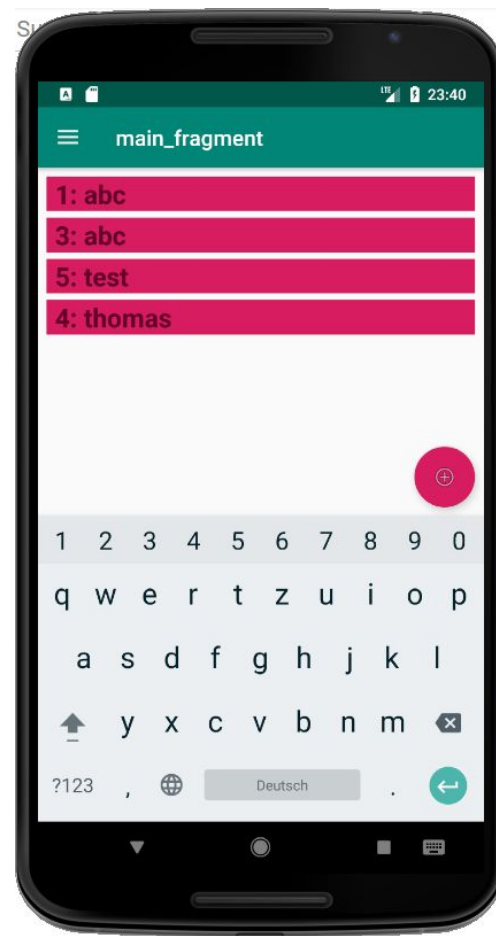
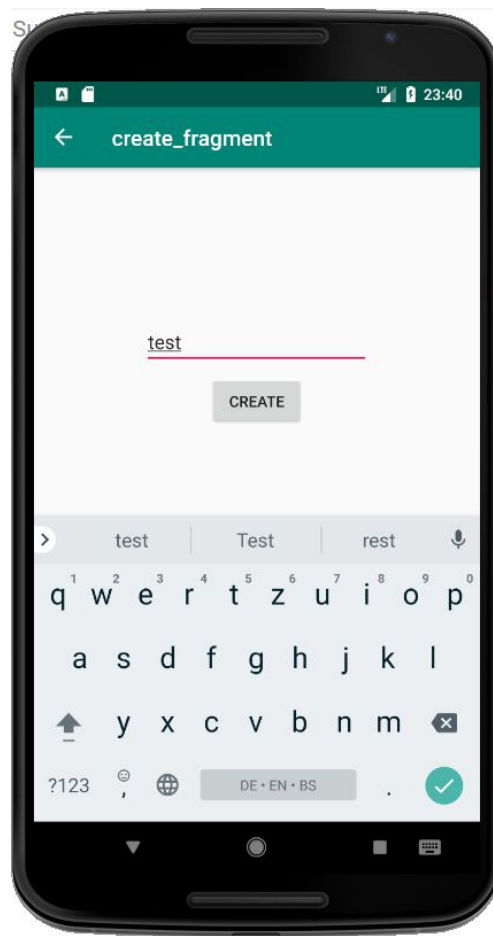
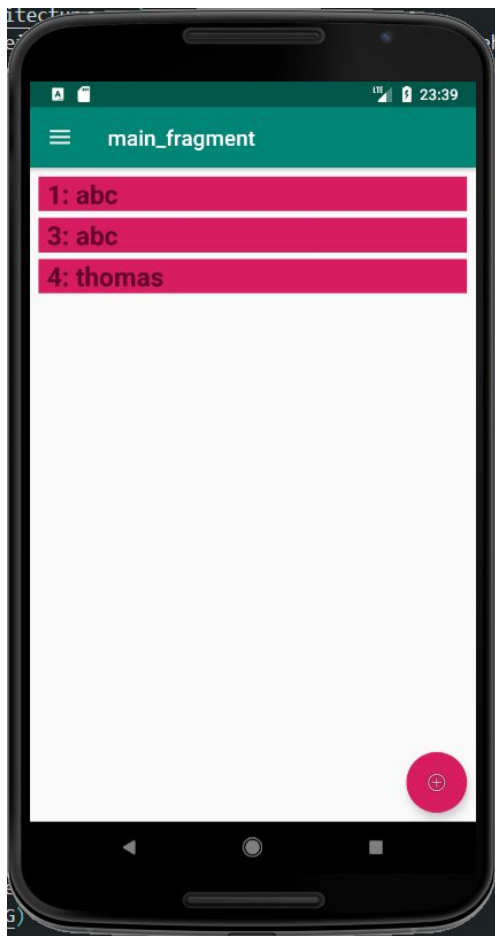
ViewModel + LiveData +
Data Binding = Reactive UI

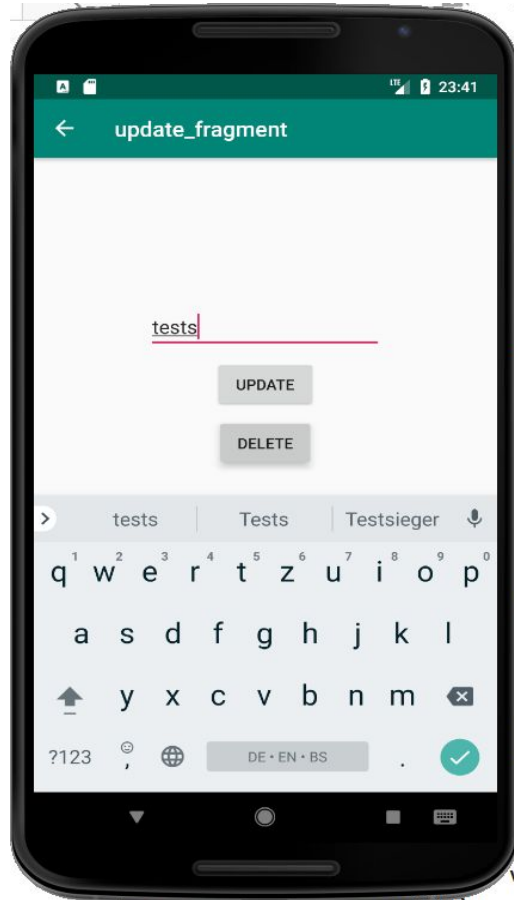
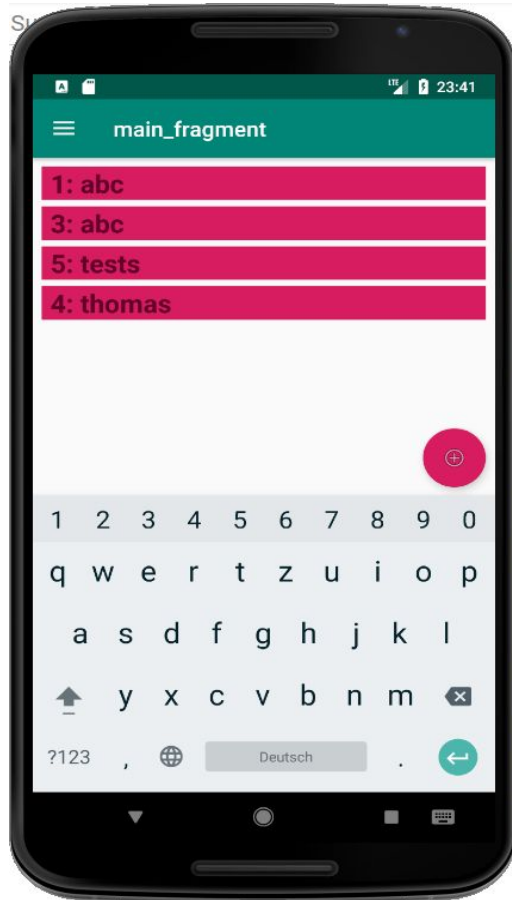
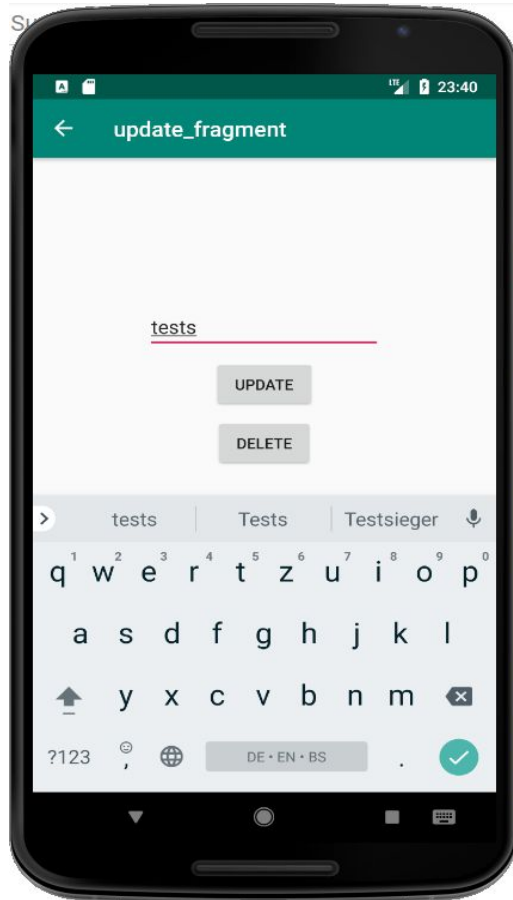


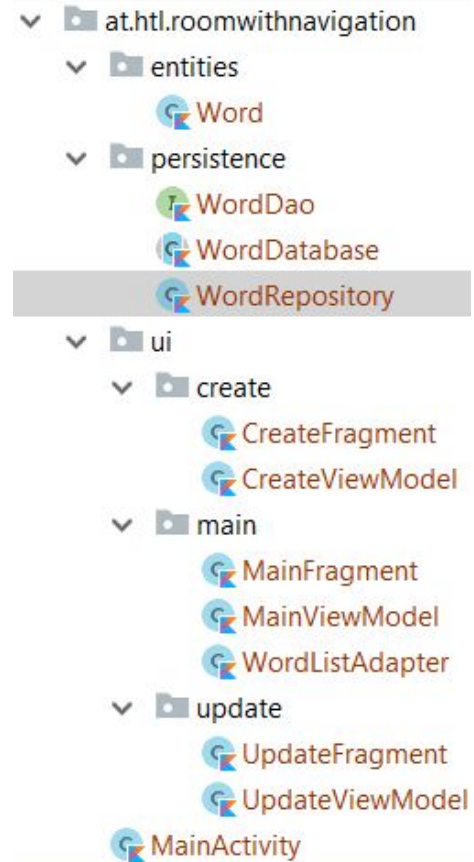
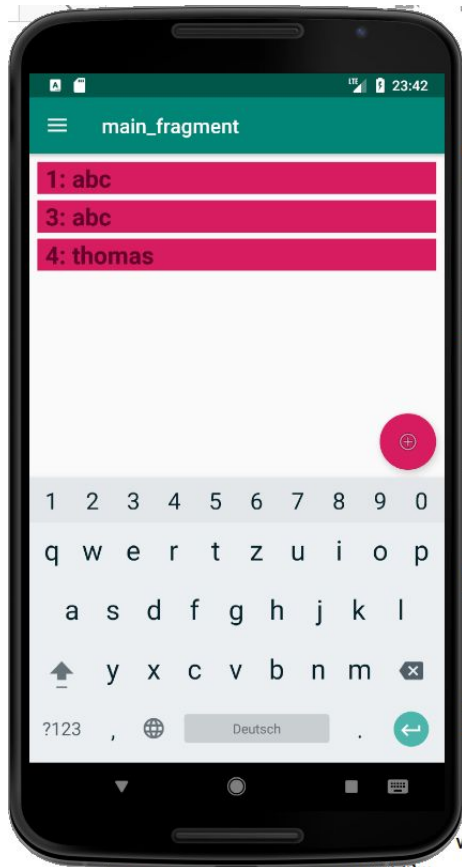


Was wir erreichen wollen

Man soll eine App entwickeln, welche mittels Room auf die SQLite-Datenbank zugreift. Darauf soll man über die CRUD-Operationen zugreifen können.









Los geht's

Create New Project

Create Android Project

Application name

RoomWithNavigation

Company domain

htl.at

Project location


D:\Schule\RoomWithNavigation\RoomWithNavigation ...

Package name

at.htl.roomwithnavigation Edit

☐ Include C++ support

☒ Include Kotlin support

 'RoomWithNavigation' already exists at the specified project location.

Previous Next Cancel Finish



Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ **Phone and Tablet**

API 27: Android 8.1 (Oreo)



By targeting **API 27 and later**, your app will run on approximately **1,1%** of devices. [Help me choose](#)

☐ Include Android Instant App support

☐ **Wear OS**

API 23: Android 6.0 (Marshmallow)



☐ **TV**

API 21: Android 5.0 (Lollipop)



☐ **Android Auto**

☐ **Android Things**

API 24: Android 7.0 (Nougat)



Previous

Next

Cancel

Finish



Add an Activity to Mobile



Basic Activity

Bottom Navigation Activity

Empty Activity

Fragment + ViewModel

Fullscreen Activity

Google AdMob Ads Activity

Google Maps Activity

Previous

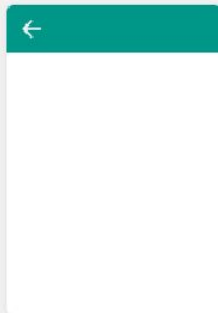
Next

Cancel

Finish



Configure Activity

**Creates a new activity and a fragment with view model**

Activity Name:

MainActivity

Activity Layout Name:

main_activity

Fragment Name:

MainFragment

Fragment Layout Name:

main_fragment

ViewModel Name:

MainViewModel

Fragment package path:

ui.main

The name of the activity class to create

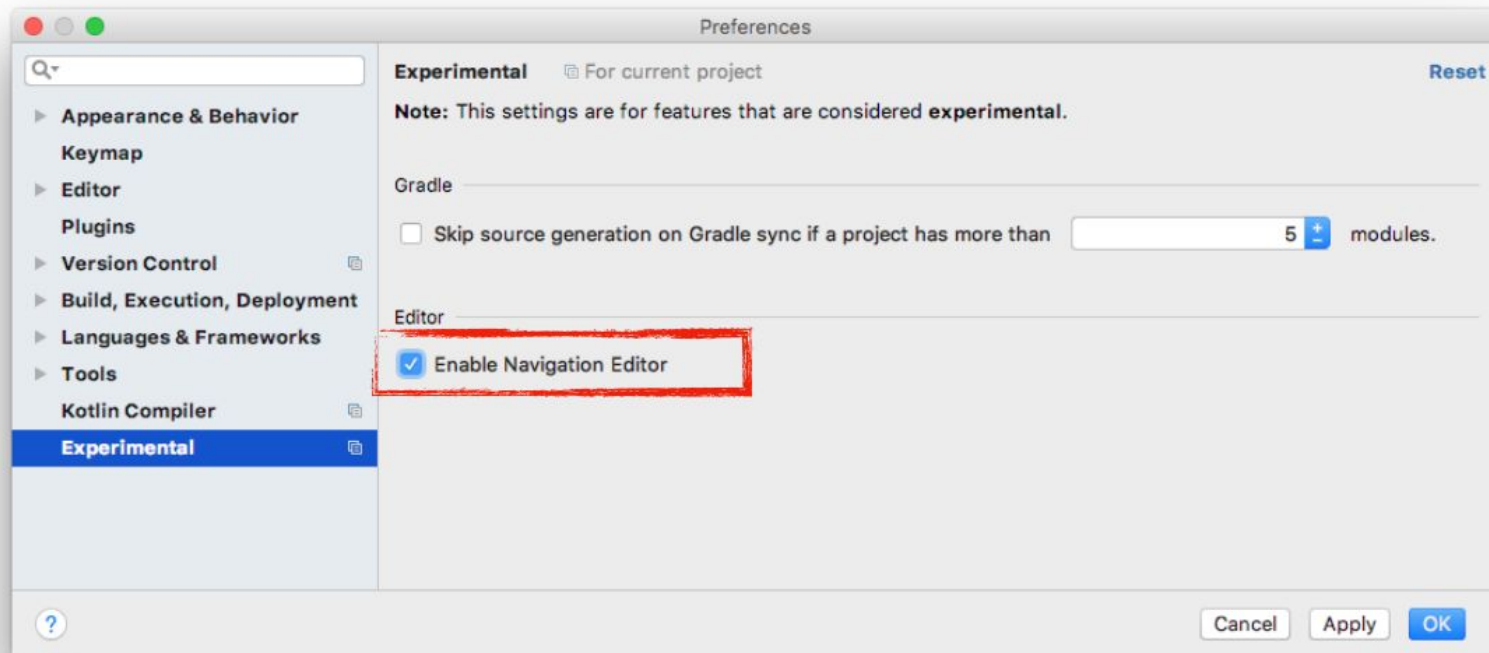
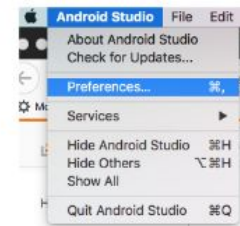
Previous

Next

Cancel

Finish

Kontrolliere ...



Im gradle build (Project: ...)

Hier zum Kopieren

```
ext {  
    roomVersion = '1.1.1'  
    archLifecycleVersion = '1.1.1'  
}
```

```
buildscript {  
    ext.kotlin_version = '1.2.70'  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.2.0'  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
  
        // NOTE: Do not place your application dependencies here; they belong  
        // in the individual module build.gradle files  
    }  
}  
  
allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}  
  
task clean(type: Delete) {  
    delete rootProject.buildDir  
}  
  
ext {  
    roomVersion = '1.1.1'  
    archLifecycleVersion = '1.1.1'  
}
```

Im gradle build (app)

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

```
apply plugin: 'kotlin-android'
```

```
apply plugin: 'kotlin-android-extensions'
```

```
apply plugin: 'kotlin-kapt'
```

```
android {  
    compileSdkVersion 27  
    defaultConfig {  
        applicationId "at.htl.roomwithnavigation"  
        minSdkVersion 27  
        targetSdkVersion 27  
        versionCode 1  
        versionName "1.0"
```

In der nächsten Folie zum Herauskopieren!

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
    implementation 'com.android.support:appcompat-v7:27.1.1'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    implementation 'android.arch.lifecycle:extensions:1.1.1'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

```
//Navigation  
def nav_version = "1.0.0-alpha06"  
  
implementation "android.arch.navigation:navigation-fragment:$nav_version" // use -ktx for Kotlin  
implementation "android.arch.navigation:navigation-ui:$nav_version" // use -ktx for Kotlin  
  
// Room components  
implementation "android.arch.persistence.room:runtime:$rootProject.roomVersion"  
kapt "android.arch.persistence.room:compiler:$rootProject.roomVersion"  
androidTestImplementation "android.arch.persistence.room:testing:$rootProject.roomVersion"  
  
// Lifecycle components  
implementation "android.arch.lifecycle:extensions:$rootProject.archLifecycleVersion"  
kapt "android.arch.lifecycle:compiler:$rootProject.archLifecycleVersion"
```

```
}
```

//Navigation

def nav_version = "1.0.0-alpha06"

implementation "android.arch.navigation:navigation-fragment:\$nav_version" *// use -ktx for Kotlin*

implementation "android.arch.navigation:navigation-ui:\$nav_version" *// use -ktx for Kotlin*

// Room components

implementation "android.arch.persistence.room:runtime:\$rootProject.roomVersion "

kapt "android.arch.persistence.room:compiler:\$rootProject.roomVersion "

androidTestImplementation "android.arch.persistence.room:testing:\$rootProject.roomVersion "

// Lifecycle components

implementation "android.arch.lifecycle:extensions:\$rootProject.archLifecycleVersion "

kapt "android.arch.lifecycle:compiler:\$rootProject.archLifecycleVersion "

ACHTUNG BEIM EINFÜGEN AUF ZEILENUMBRUCH und LEERZEILEN

FAILURE: Build failed with an exception.

* Where:

Build file '[D:\RoomWithNavigation\app\build.gradle](#)' line: 40

* What went wrong:

A problem occurred evaluating project ':app'.

> Could not get unknown property 'Kotlin' for object of type org.gradle.api.internal.artifacts.dsl.dependencies.DefaultDependencyHandler.

* Try:

Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at <https://help.gradle.org>

CONFIGURE FAILED in 0s

Could not get unknown property 'Kotlin' for object of type org.gradle.api.internal.artifacts.dsl.dependencies.DefaultDependencyHandler.

[Open File](#)

```
//Navigation
```

```
def nav_version = "1.0.0-alpha06"
```

```
implementation "android.arch.navigation:navigation-fragment:$nav_version" // use -ktx for  
Kotlin
```

```
implementation "android.arch.navigation:navigation-ui:$nav_version" // use -ktx for Kotlin
```

```
// Room components
```

```
implementation "android.arch.persistence.room:runtime:$rootProject.roomVersion"
```

```
kapt "android.arch.persistence.room:compiler:$rootProject.roomVersion"
```

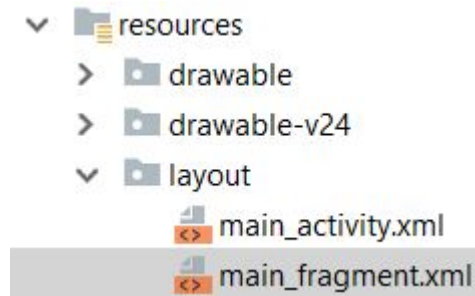
```
androidTestImplementation "android.arch.persistence.room:testing:$rootProject.roomVersion"
```

```
// Lifecycle components
```

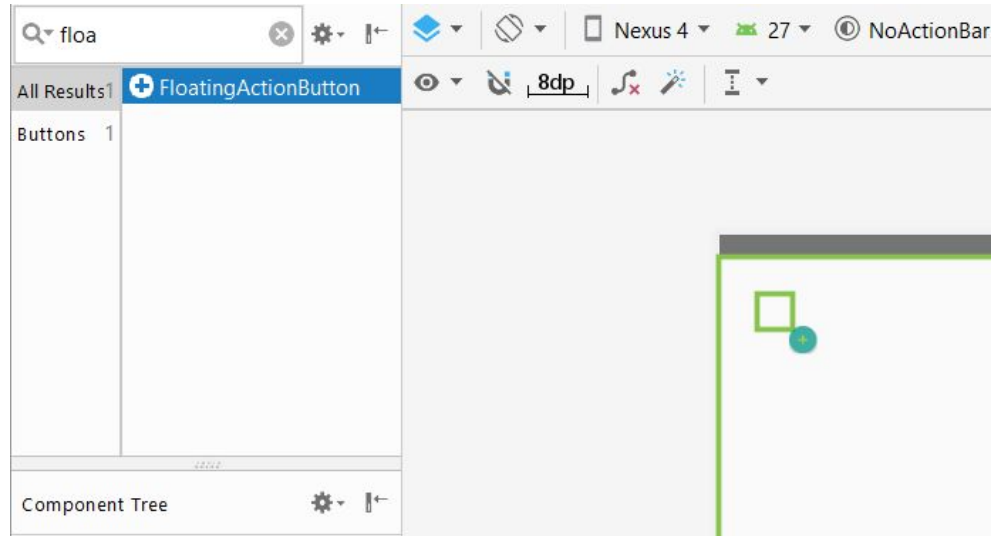
```
implementation "android.arch.lifecycle:extensions:$rootProject.archLifecycleVersion"
```

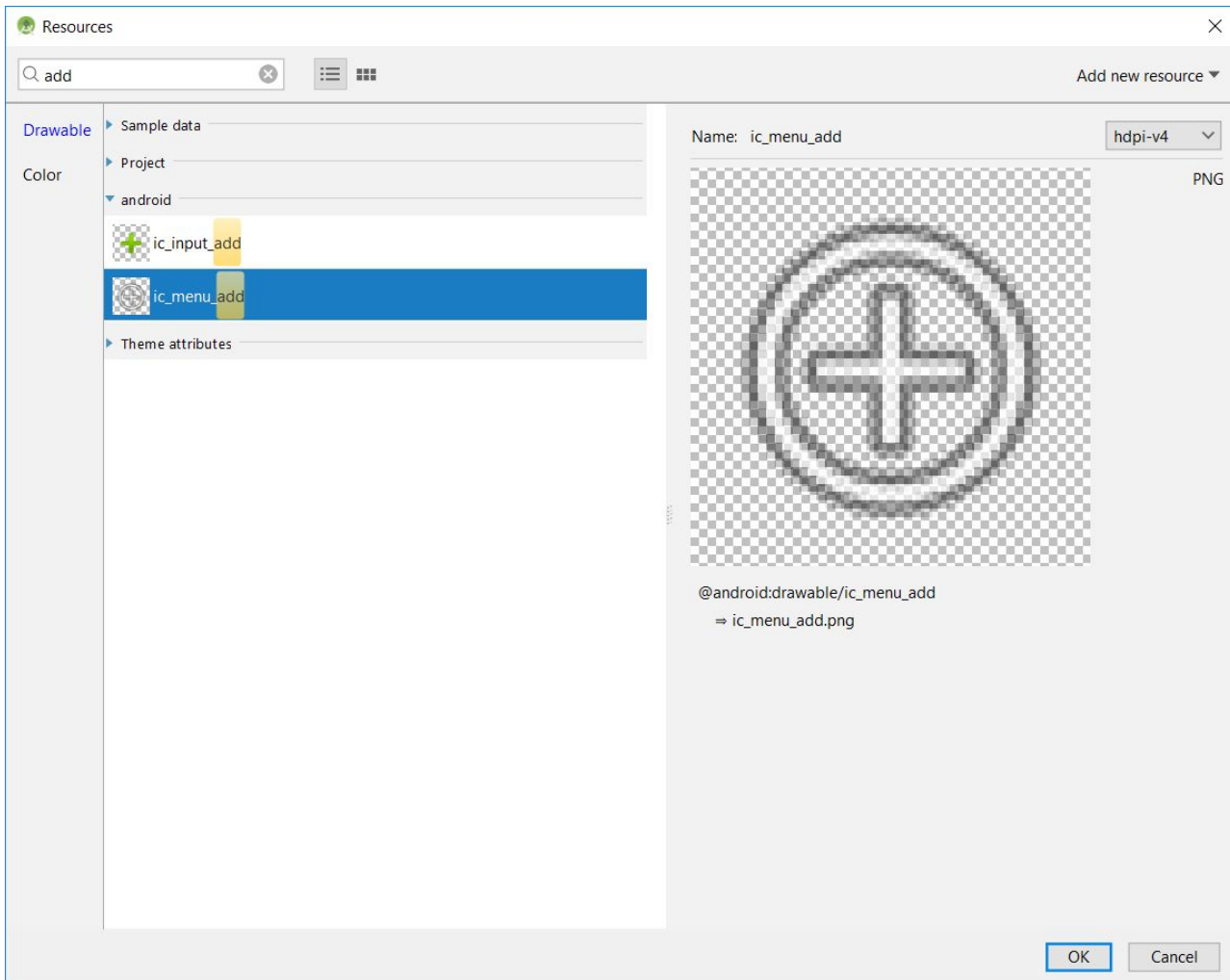
```
kapt "android.arch.lifecycle:compiler:$rootProject.archLifecycleVersion"
```

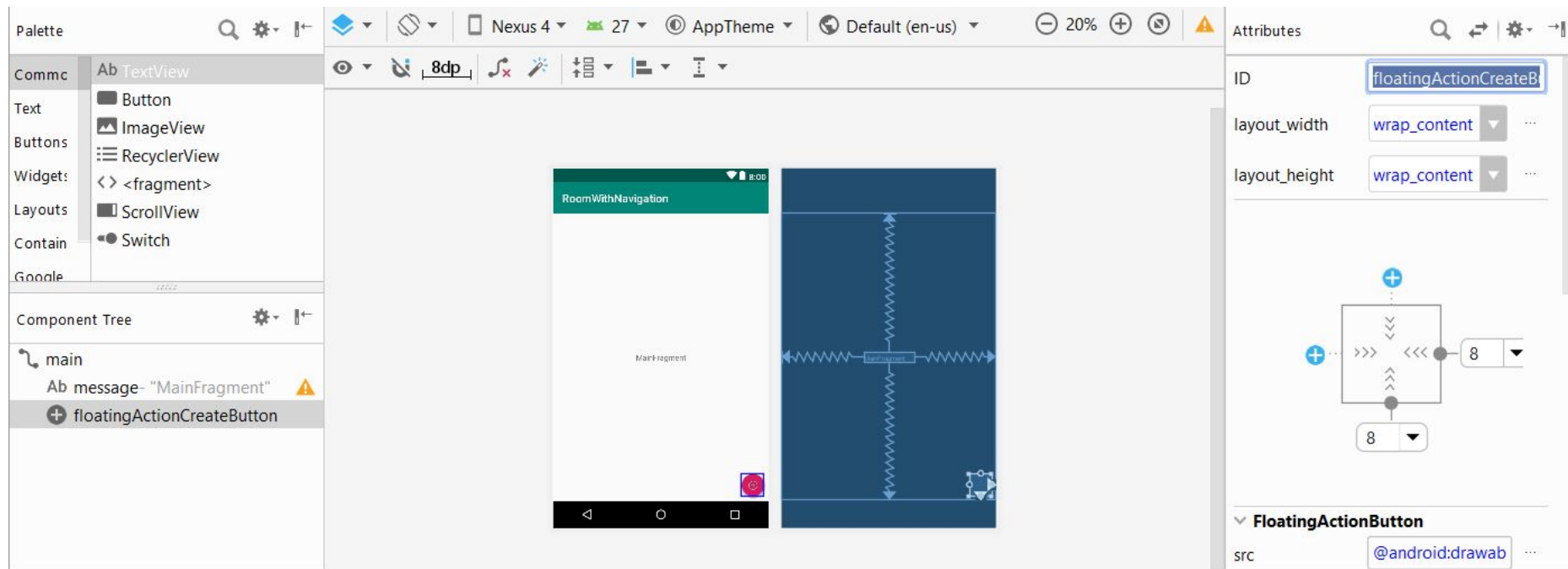
In resource > layout > main_fragment suchen nach floatingActionButton



Einfach den floatingActionButton
in das Fragment ziehen

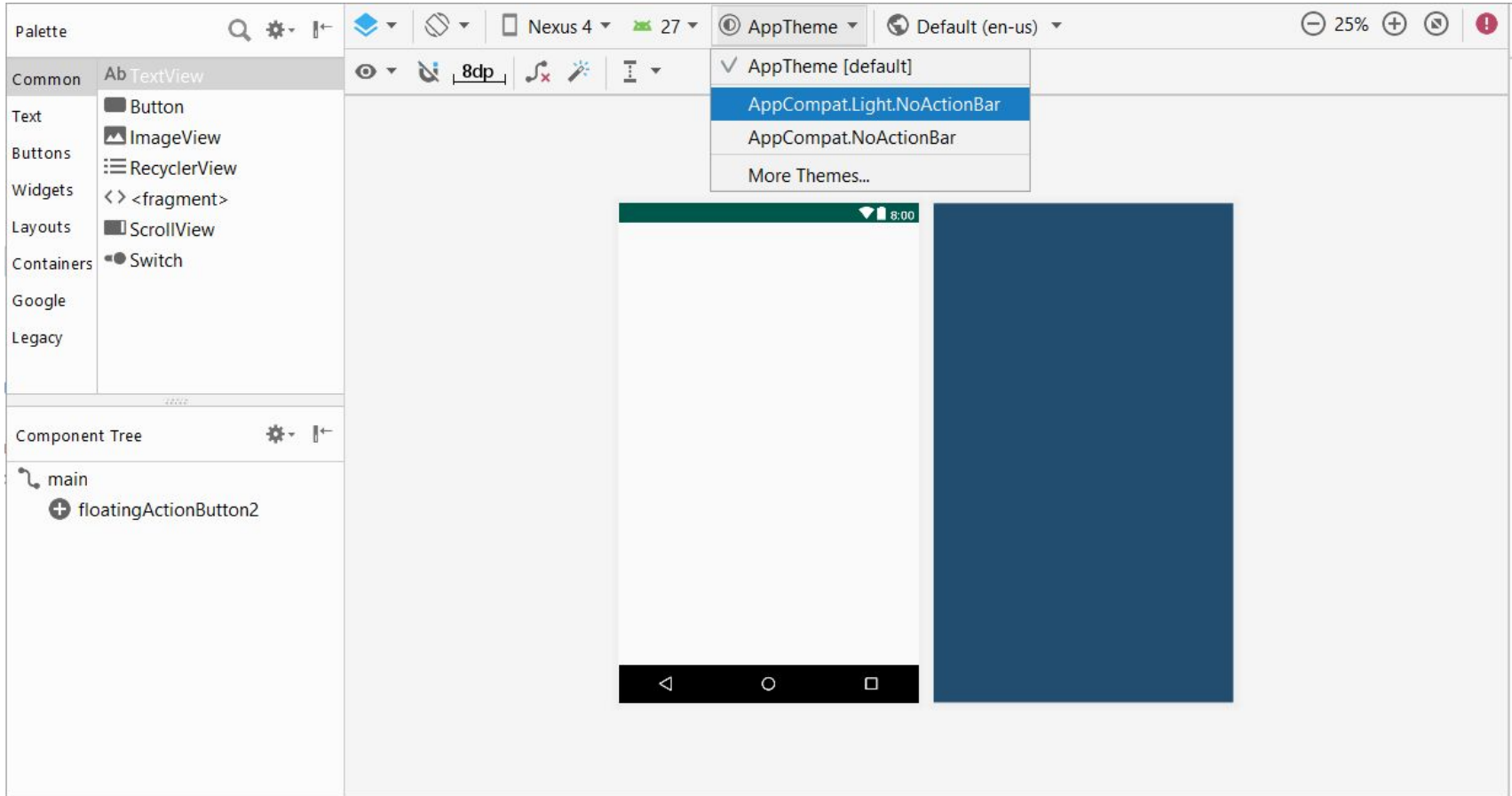




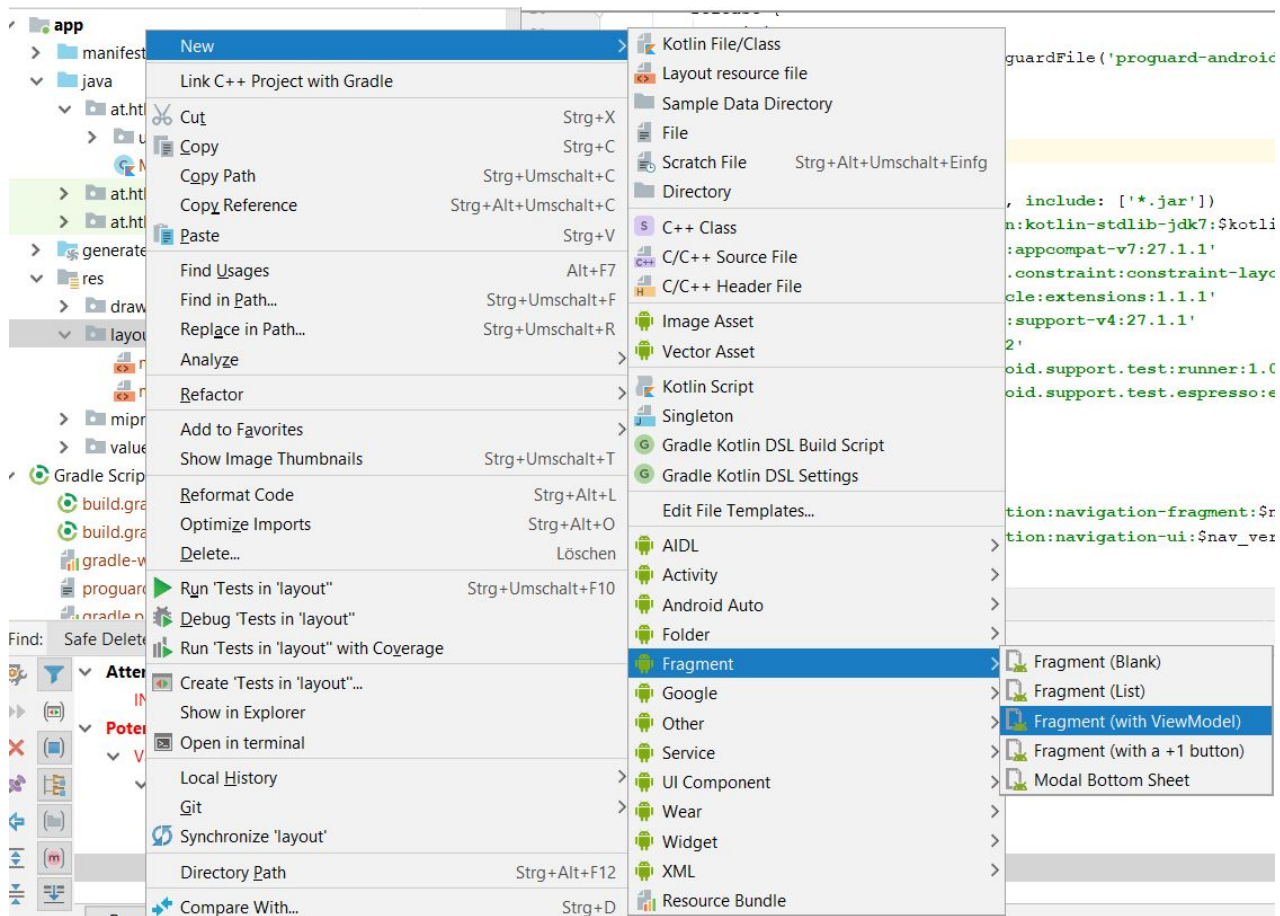


Falls der Button nicht angezeigt wird siehe nächste Folie

Wenn der Button nicht angezeigt wird, hat es geholfen das Theme zu ändern



layout > new > fragment > fragment (with viewmodel)





Configure Component

Android Studio

Creates a Fragment with a ViewModel.

Fragment Name:

Fragment Layout Name:

ViewModel Name:

Source Language:



The name of the fragment class to create

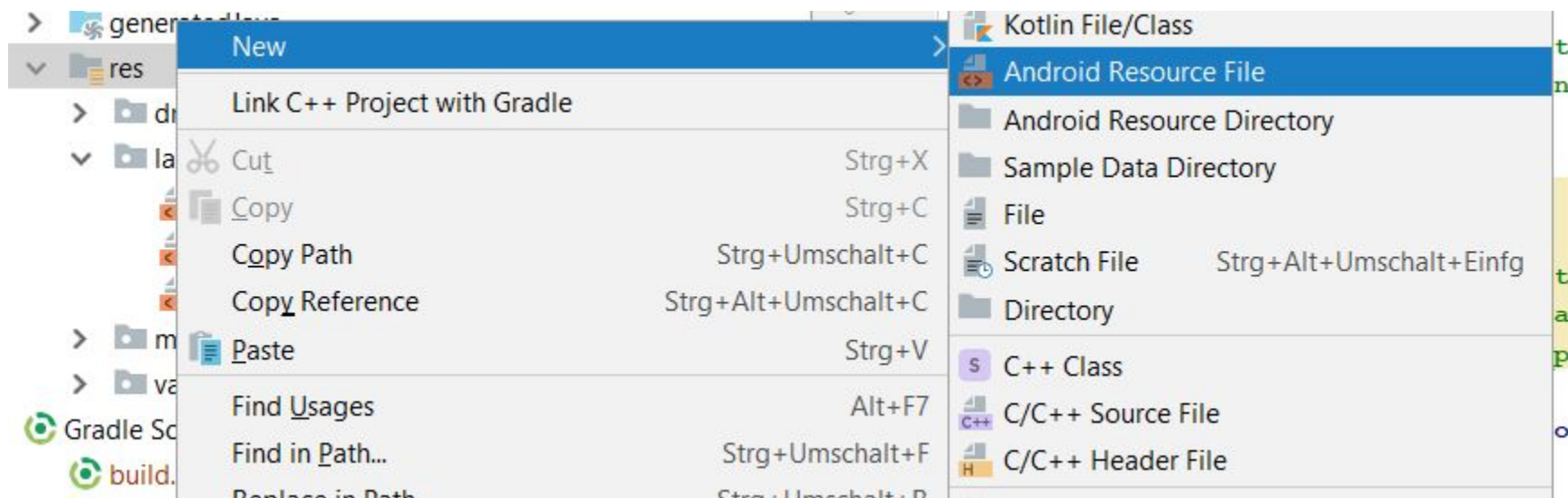
Jetzt noch das Fragment in die Ordnerstruktur bringen





Hinzufügen des Navigation Graphs

res > new > android resource file



New Resource File



File name:



Resource type:

Root element:

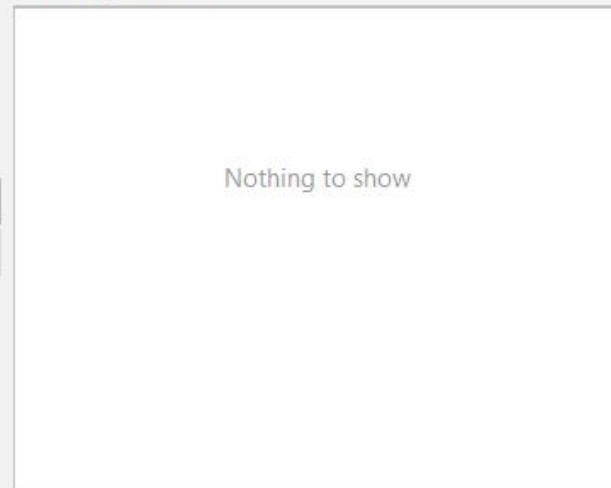
Source set:

Directory name:

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode

Chosen qualifiers:



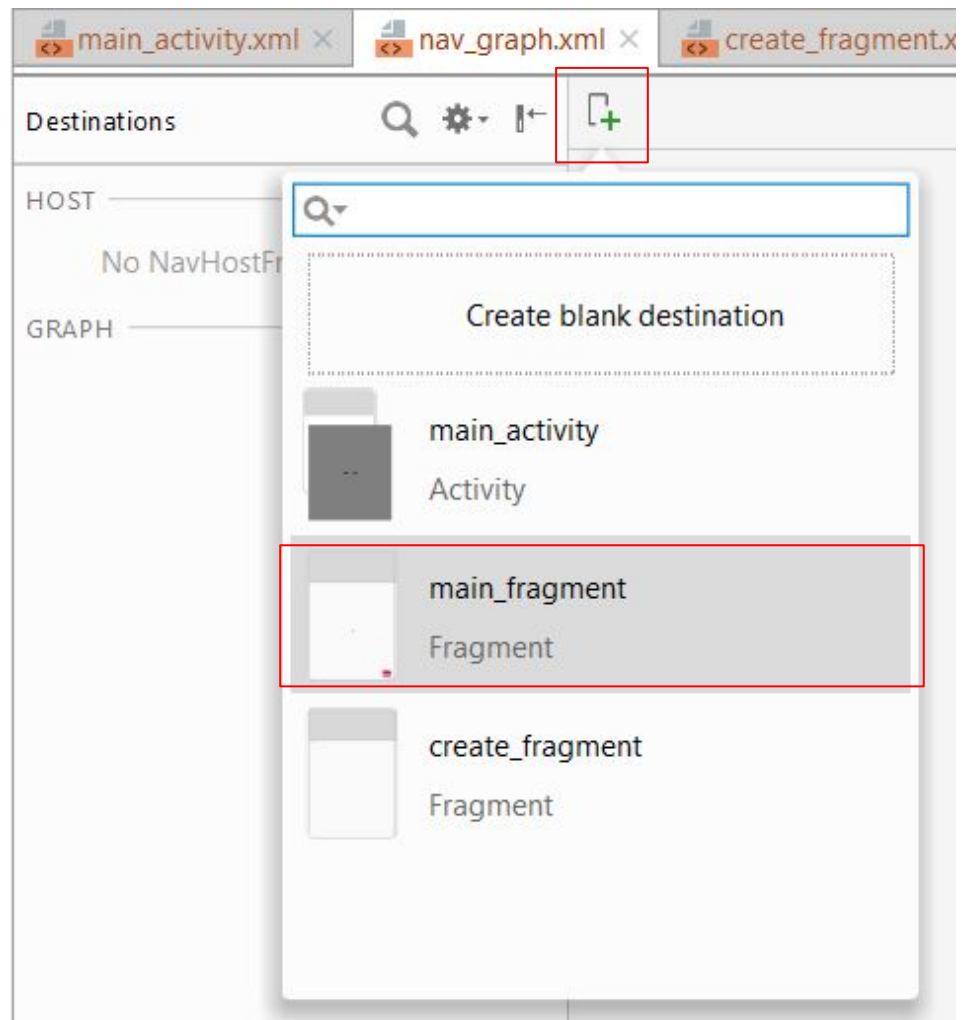
Nothing to show



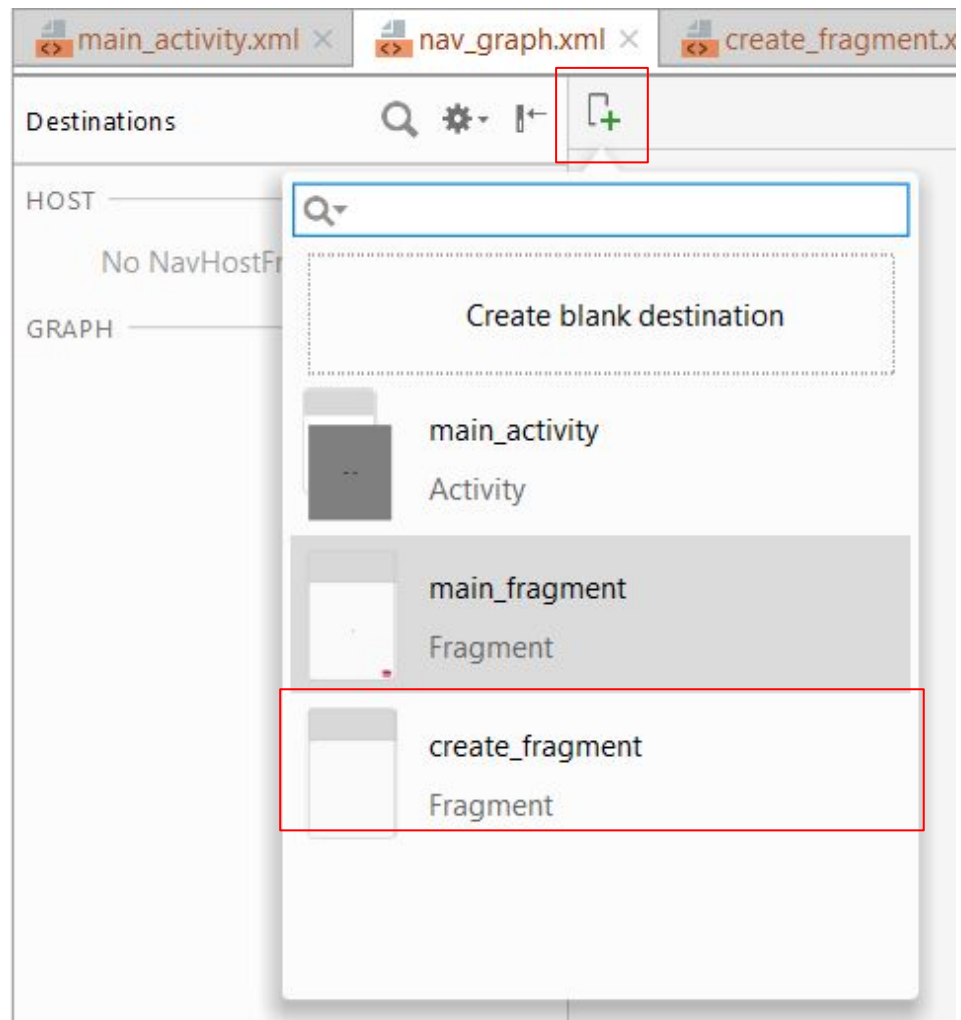
OK

Cancel

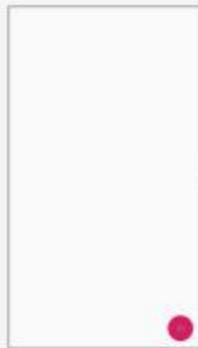
Im Designer



Im Designer



mainFragment



createFragment



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/container"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   tools:context=".MainActivity">
9
10   <fragment
11     android:id="@+id/my_fragment"
12     android:name="androidx.navigation.fragment.NavHostFragment"
13     android:layout_width="match_parent"
14     android:layout_height="match_parent"
15     app:defaultNavHost="true"
16     app:navGraph="@navigation/nav_graph"></fragment>
17
18 </android.support.v4.widget.DrawerLayout>
```

```
class MainActivity : AppCompatActivity() {
```

```
    lateinit var drawer: DrawerLayout
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.main_activity)
```

```
        val navController = (my_fragment as NavHostFragment).navController
```

```
        drawer = container
```

```
        NavigationUI.setupActionBarWithNavController(activity: this, navController, drawer)
```

```
    }
```

```
    override fun onSupportNavigateUp(): Boolean {
```

```
        return NavigationUI.navigateUp(drawer, Navigation.findNavController(activity: this, my_fragment.id))
```

```
    }
```

```
}
```


In **MainFragment.kt** fgt man noch diese Methode ein
Damit setzt man den onClickListener auf den FloatingActionButton

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {  
    super.onCreateView(view, savedInstanceState)  
  
    floatingActionButton.setOnClickListener { it: View!  
        Navigation.findNavController(it).navigate(createFragment)  
    }  
}
```

Jetzt kann man vom main- zum create-Fragment wechseln

Nun geht es zu der Erstellung der nötigen Files für Room

Zuerst die Word Entität

▼ at.htl.roomwithnavigation

▼ entities

Word.kt

```
import android.arch.persistence.room.Entity
import android.arch.persistence.room.PrimaryKey

@Entity(tableName = "word_table")
data class Word(
    @PrimaryKey(autoGenerate = true) val id: Long,
    var word: String)
|
```

- ▼ entities
 - Word
- ▼ persistence
 - WordDao.kt
 - WordDatabase.kt
 - WordRepository.kt

```
@Dao
interface WordDao {
    @Insert
    fun insert(word: Word)

    @Update
    fun update(word: Word)

    @Query( value: "SELECT * from word_table ORDER BY id ASC")
    fun getAllLive(): LiveData<List<Word>>

    @Query( value: "DELETE FROM word_table")
    fun deleteAll()

    @Delete
    fun delete(word: Word)
}
```

In dem entities gibt man die Entitäten an welche in der Datenbank gespeichert werden

Die Version gibt an in welcher Version sich die Datenbank befindet und wenn man die Struktur ändert kann man eine Migrationsfunktion programmieren, sodass die aktuellen Apps auch auf die aktuelle Struktur geändert werden

```
import android.arch.persistence.room.Database
import android.arch.persistence.room.RoomDatabase
import at.htl.roomwithnavigation.entities.Word

@Database(entities = [Word::class], version = 1)
abstract class WordDatabase : RoomDatabase() {
    |
}
```

```
@Database(entities = [Word::class], version = 1)
abstract class WordDatabase : RoomDatabase() {

    abstract fun wordDao(): WordDao

    fun getWordDao(): WordDao = wordDao()

    companion object {
        private var INSTANCE: WordDatabase? = null

        fun getInstance(ctx: Context): WordDatabase {
            if (INSTANCE == null) {
                INSTANCE = Room.databaseBuilder(ctx,
                    WordDatabase::class.java, name: "word_database")
                    .build()
            }
            return INSTANCE as WordDatabase
        }
    }
}
```

```
class WordRepository(application: Application) {  
    private val wordDatabase: WordDatabase = WordDatabase.getInstance(application)  
    private val wordDao: WordDao = wordDatabase.getWordDao()  
  
    fun insert(word: Word) {  
        thread {  
            wordDao.insert(word)  
        }  
    }  
  
    fun update(word: Word) {  
        thread {  
            wordDao.update(word)  
        }  
    }  
  
    fun delete(word: Word) {  
        thread {  
            wordDao.delete(word)  
        }  
    }  
  
    fun getAllLive(): LiveData<List<Word>> = wordDao.getAllLive()  
}
```

Im main-fragment löscht man zuerst die Textview

```
<TextView
    android:id="@+id/message"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="MainFragment"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

im main-fragment

```
<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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.main.MainActivity">
```

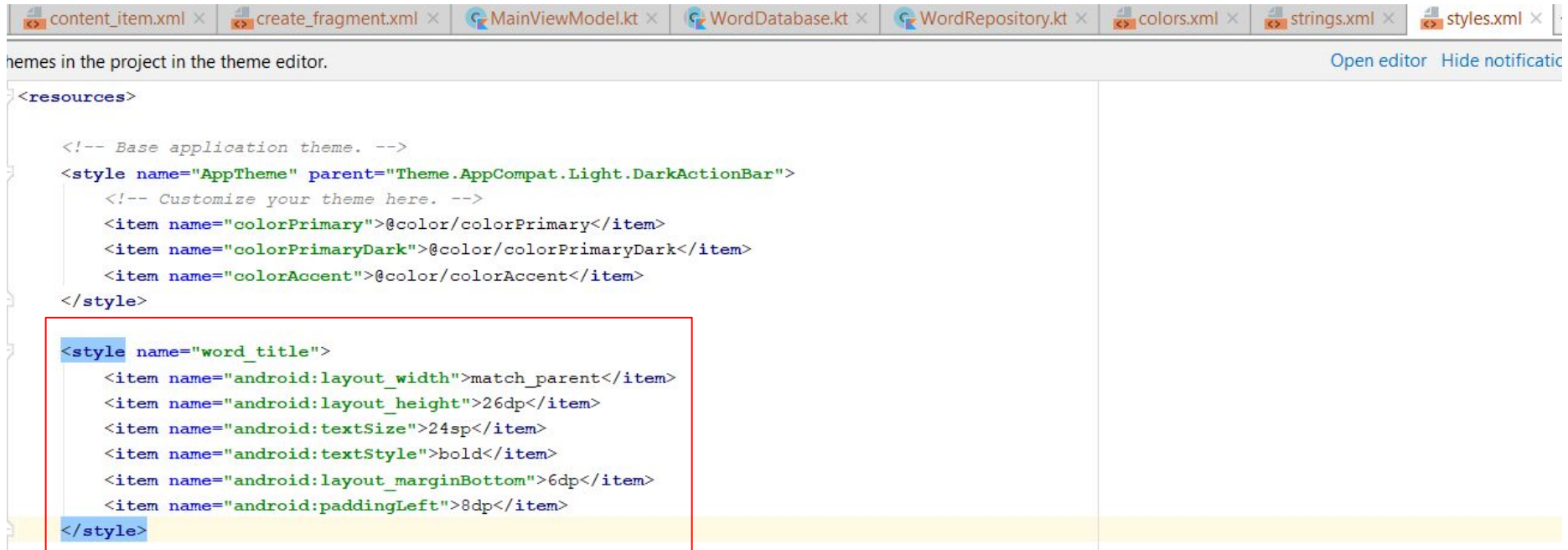
```
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/floatingActionCreateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:clickable="true"
        android:src="@android:drawable/ic_menu_add"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
```

```
    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:listitem="@layout/content_item" />
```

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

Keine Sorge wegen des
content_items. Das
erstellen wir jetzt gleich

im res > values > styles xml fügen wir einen neuen style für unser content_item den kann man ähnlich wie eine css-klasse verwenden



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

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

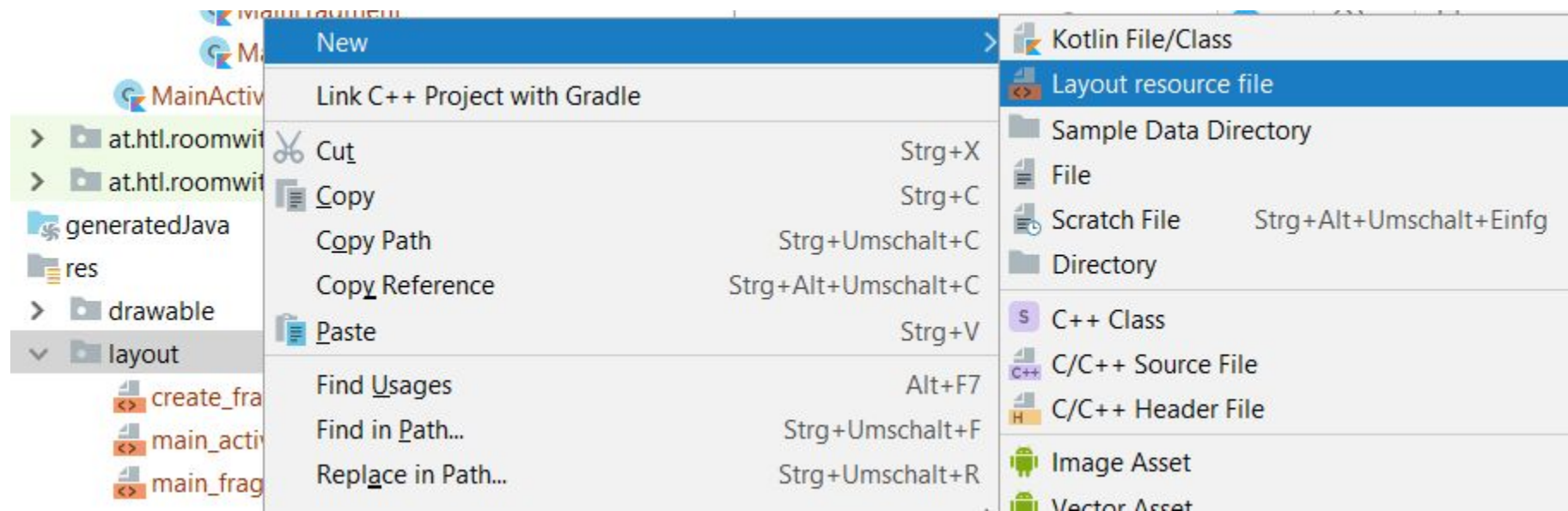
    <style name="word_title">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">26dp</item>
        <item name="android:textSize">24sp</item>
        <item name="android:textStyle">bold</item>
        <item name="android:layout_marginBottom">6dp</item>
        <item name="android:paddingLeft">8dp</item>
    </style>


```

Sollte es so aussehen kann man es ignorieren und weiterarbeiten. Es funktioniert genauso.

```
<item name="android:layout_width">match_parent</item>  
<item name="android:layout_height">26dp</item>  
<item name="android:textSize">24sp</item>  
<item name="android:textStyle">bold</item>  
<item name="android:layout_marginBottom">6dp</item>  
<item name="android:paddingLeft">8dp</item>
```

layout > new > layout resource file





New Resource File



File name:

Root element:

Source set:



Directory name:

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode
- Density

Chosen qualifiers:

Nothing to show

>>

<<



OK

Cancel

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


    <TextView
        android:id="@+id/textView"
        style="@style/word_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorAccent"/>

</LinearLayout>
```

Hier setzt man dann den style

layout > new > fragment > fragment (with viewmodel)


New Android Component



Configure Component

Android Studio

Creates a Fragment with a ViewModel.



Fragment Name:

UpdateFragment

Fragment Layout Name:

update_fragment

ViewModel Name:

UpdateViewModel

Source Language:

Kotlin











The name of the fragment class to create

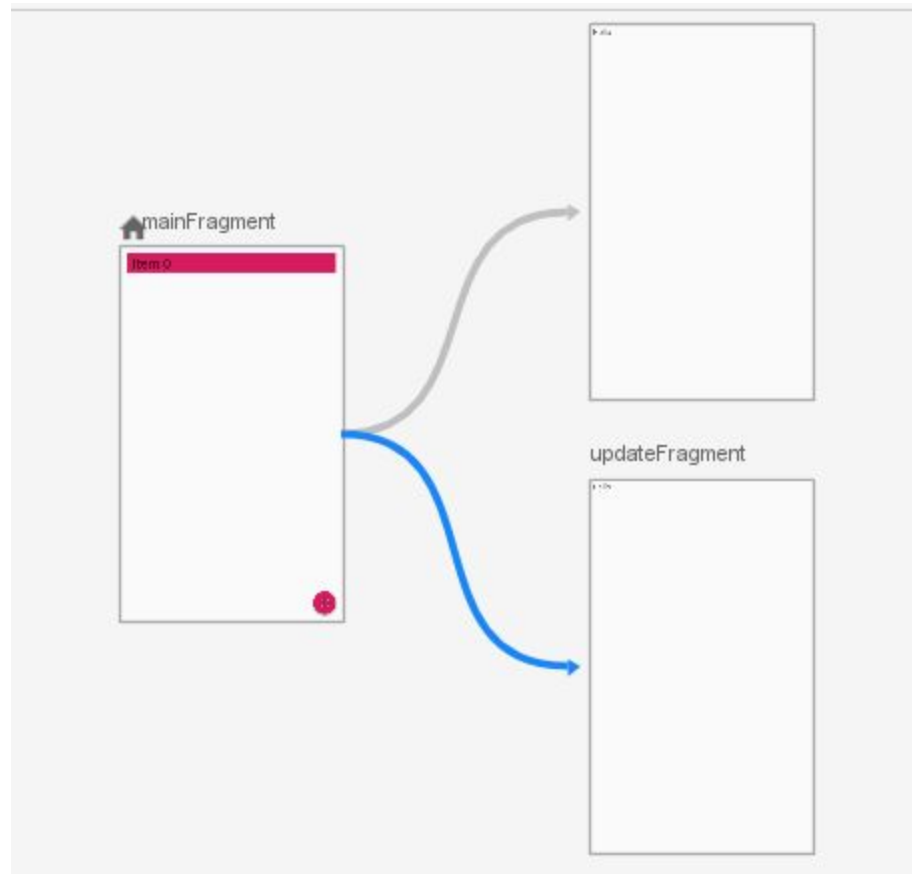
Previous

Next

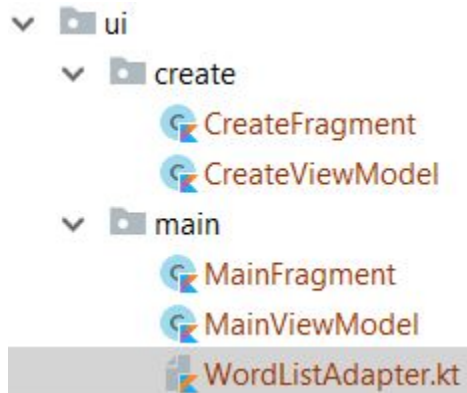
Cancel

Finish

- ▼  ui
 - ▼  create
 -  CreateFragment
 -  CreateViewModel
 - ▼  main
 -  MainFragment
 -  MainViewModel
 - ▼  update
 -  UpdateFragment
 -  UpdateViewModel



Jetzt müssen wir einen Adapter für die RecyclerView erstellen



```
class WordListAdapter(var list: List<Word> = listOf()) : RecyclerView.Adapter<WordListAdapter.WordViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): WordViewHolder {
        val view = LayoutInflater
            .from(parent.context)
            .inflate(R.layout.content_item, parent, attachToRoot: false)
        return WordViewHolder(view)
    }

    override fun getItemCount(): Int = list.size

    override fun onBindViewHolder(holder: WordViewHolder, position: Int) {
        val current: Word = list[position]
        holder.view.textView.text = "${current.id}: ${current.word}"
        holder.view.setOnClickListener { it: View!
            var bundle = Bundle()
            bundle.putLong("Id", current.id)
            bundle.putString("Word", current.word)
            Navigation.findNavController(it).navigate(action_mainFragment_to_updateFragment, bundle)
        }
    }

    class WordViewHolder(val view: View) : RecyclerView.ViewHolder(view)
}
```

```
class MainViewModel(application: Application) : AndroidViewModel(application) {  
  
    private val mRepository: WordRepository = WordRepository(application)  
    private var mAllWords: LiveData<List<Word>> = mRepository.getAllLive()  
  
    fun getAllWords(): LiveData<List<Word>> = mAllWords  
}
```

```
class MainFragment : Fragment() {

    companion object {
        fun newInstance() = MainFragment()
    }

    private lateinit var viewModel: MainViewModel

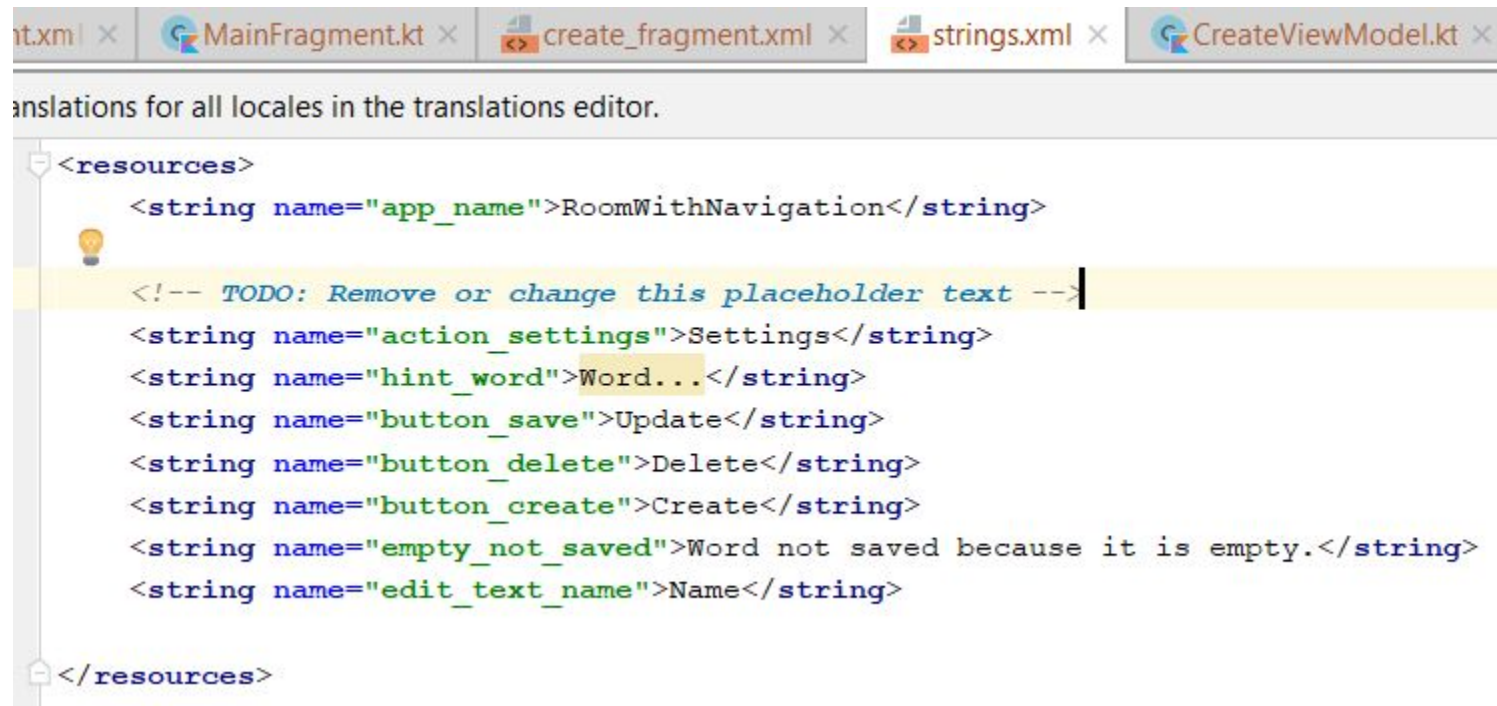
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View {
        return inflater.inflate(R.layout.main_fragment, container, attachToRoot: false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProviders.of(fragment: this).get(MainViewModel::class.java)

        var adapter = WordListAdapter()
        recyclerview.adapter = adapter
        recyclerview.layoutManager = LinearLayoutManager(this.context)

        viewModel.getAllWords().observe(owner: this, Observer<List<Word>> { it: List<Word>?
            adapter.list = it!!
            adapter.notifyDataSetChanged()
        })
    }
}
```

In das string xml file fügen wir die notwendigen Strings ein



The screenshot shows an IDE window with several tabs: 'nt.xml', 'MainFragment.kt', 'create_fragment.xml', 'strings.xml', and 'CreateViewModel.kt'. The 'strings.xml' tab is active, displaying the XML content for translations. A light gray banner at the top of the editor area reads 'Translations for all locales in the translations editor.' The XML code is as follows:

```
<resources>
  <string name="app_name">RoomWithNavigation</string>
  <!-- TODO: Remove or change this placeholder text -->
  <string name="action_settings">Settings</string>
  <string name="hint_word">Word...</string>
  <string name="button_save">Update</string>
  <string name="button_delete">Delete</string>
  <string name="button_create">Create</string>
  <string name="empty_not_saved">Word not saved because it is empty.</string>
  <string name="edit_text_name">Name</string>
</resources>
```

Auf den nächsten beiden Seiten
zum Kopieren

```
agument.xml x MainFragment.kl x create_fragment.xml x strings.xml

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".ui.create.CreateFragment">

<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:hint="@string/edit_text_name"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_create"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
</android.support.constraint.ConstraintLayout>
```


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

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:hint="@string/edit_text_name"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_create"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />

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



```
class CreateViewModel(application: Application) : AndroidViewModel(application) {  
      
    private val mRepository = WordRepository(application)  
  
    fun insert(word: Word) {  
        mRepository.insert(word)  
    }  
}
```

```

class CreateFragment : Fragment() {

    companion object {
        fun newInstance() = CreateFragment()
    }

    private lateinit var viewModel: CreateViewModel

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.create_fragment, container, attachToRoot: false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProviders.of(fragment: this).get(CreateViewModel::class.java)
    }

```

```

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        button.setOnClickListener { it: View!
            viewModel.insert(Word( id: 0, editText.text.toString()))
            Navigation.findNavController(it).popBackStack()
        }
    }

```

Im update_fragment

Auf den Seiten
danach zum
Herauskopieren

```
<?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=".ui.update.UpdateFragment">
```

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:hint="@string/edit_text_name"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button_update"
    android:layout_width="91dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_save"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
```

```
<Button
    android:id="@+id/button_delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_delete"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_update" />
</android.support.constraint.ConstraintLayout>
```

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

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp"
        android:ems="10"
        android:hint="@string/edit_text_name"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button_update"
    android:layout_width="91dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_save"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
```

```
<Button
    android:id="@+id/button_delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/button_delete"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_update" />
```

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

```
class UpdateViewModel(application: Application) : AndroidViewModel(application) {

    private var mRepository = WordRepository(application)

    fun update(word: Word) {
        | mRepository.update(word)
    }

    fun delete(word: Word) {
        | mRepository.delete(word)
    }
}
```

```
class UpdateFragment : Fragment() {

    companion object {
        fun newInstance() = UpdateFragment()
    }

    private lateinit var viewModel: UpdateViewModel

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        return inflater.inflate(R.layout.update_fragment, container, attachToRoot: false)
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        viewModel = ViewModelProviders.of(fragment: this).get(UpdateViewModel::class.java)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        val id = arguments?.getLong( key: "Id")!!
        editText.setText(arguments?.getString( key: "Word").toString())

        button_update.setOnClickListener { it: View!
            viewModel.update(Word(id, editText.text.toString()))
            Navigation.findNavController(it).popBackStack()
        }

        button_delete.setOnClickListener { it: View!
            viewModel.delete(Word(id, editText.text.toString()))
            Navigation.findNavController(it).popBackStack()
        }
    }
}
```




**Damit ist es auch schon
geschafft :)**



Quellen

<https://www.youtube.com/watch?v=5qlIPTDE274>

<https://codelabs.developers.google.com/codelabs/android-room-with-a-view/#0>

Github:

<https://github.com/ThomasKaar/RoomWithNavigation>