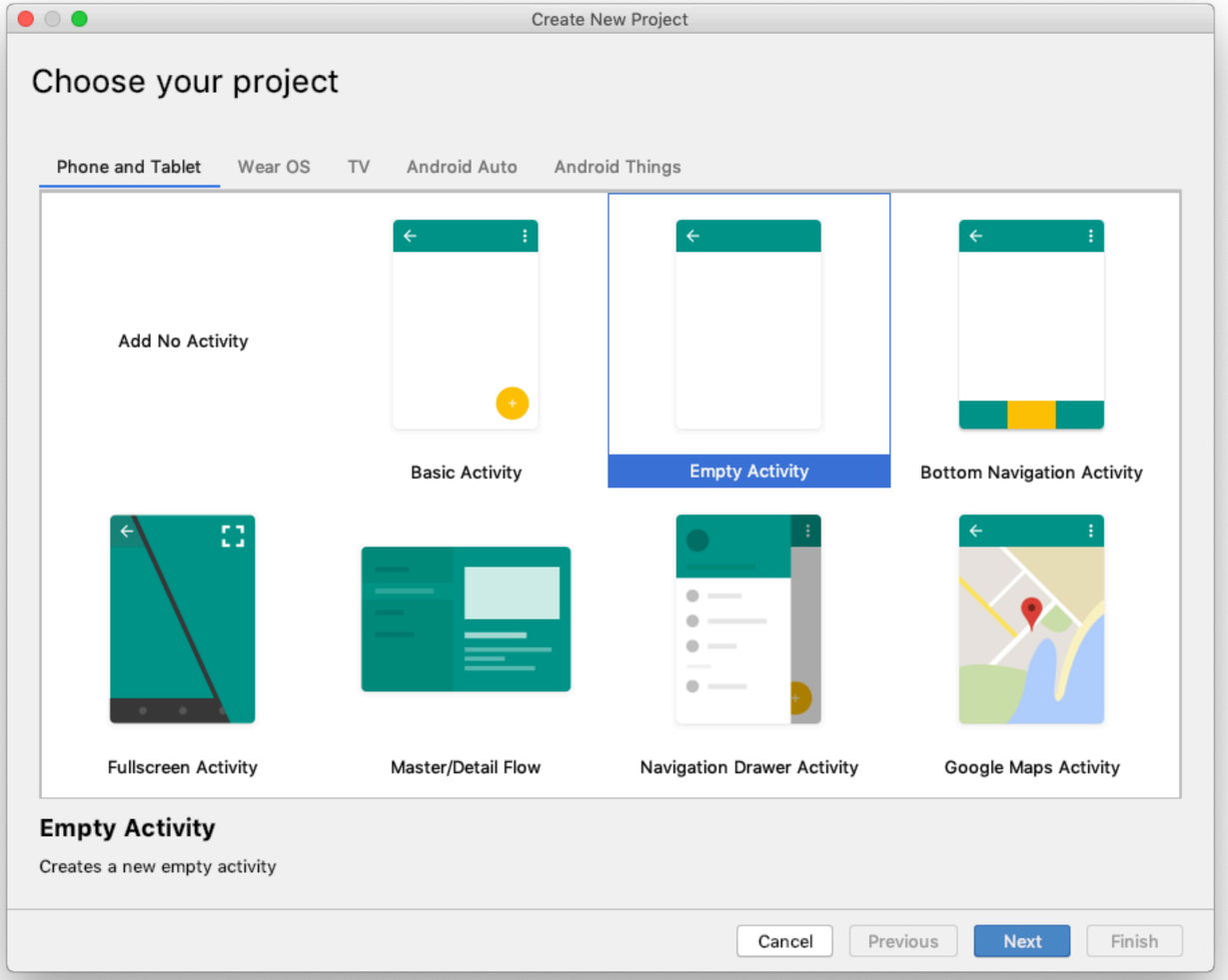
# Navigation Components

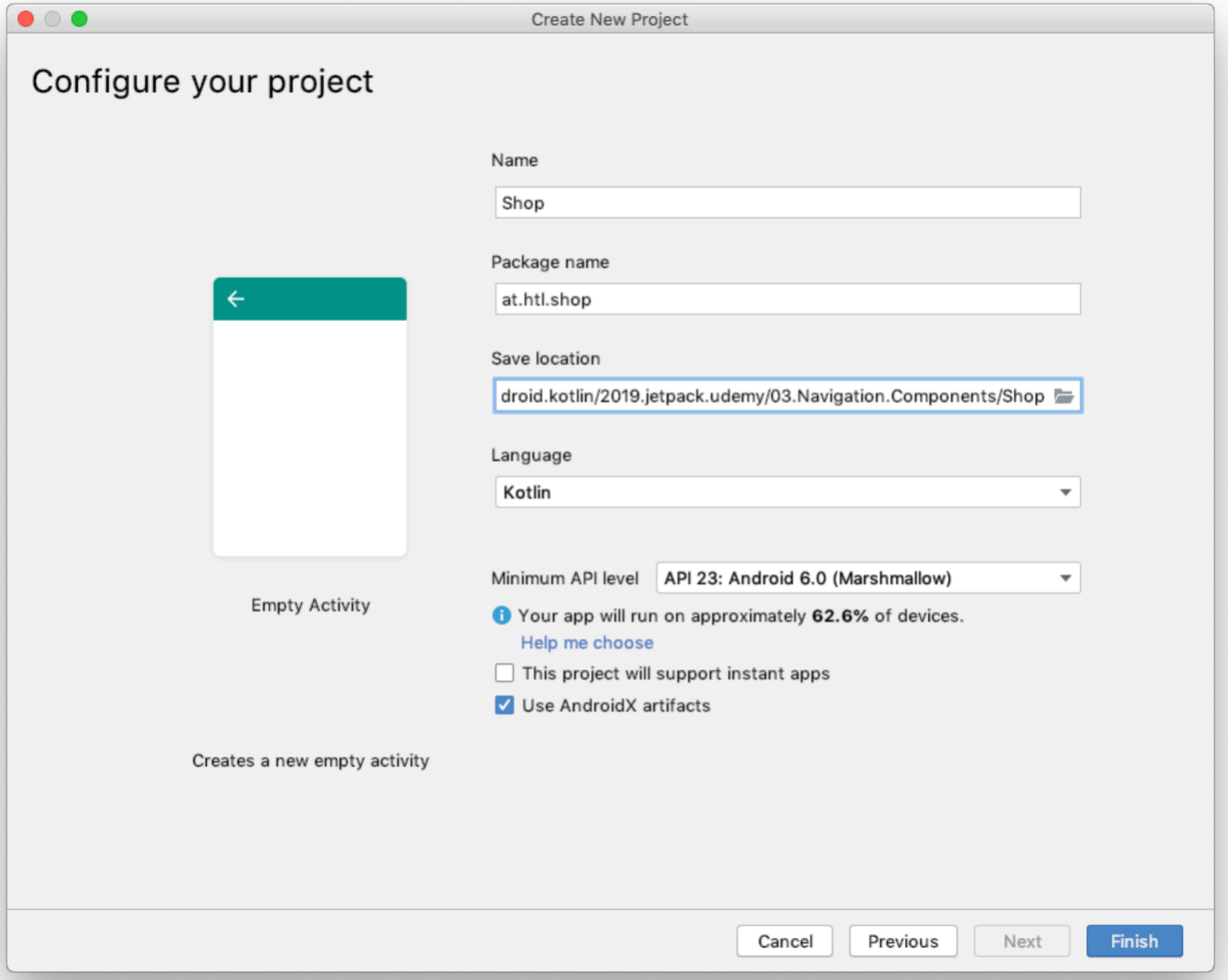
## Jetpack Architecture

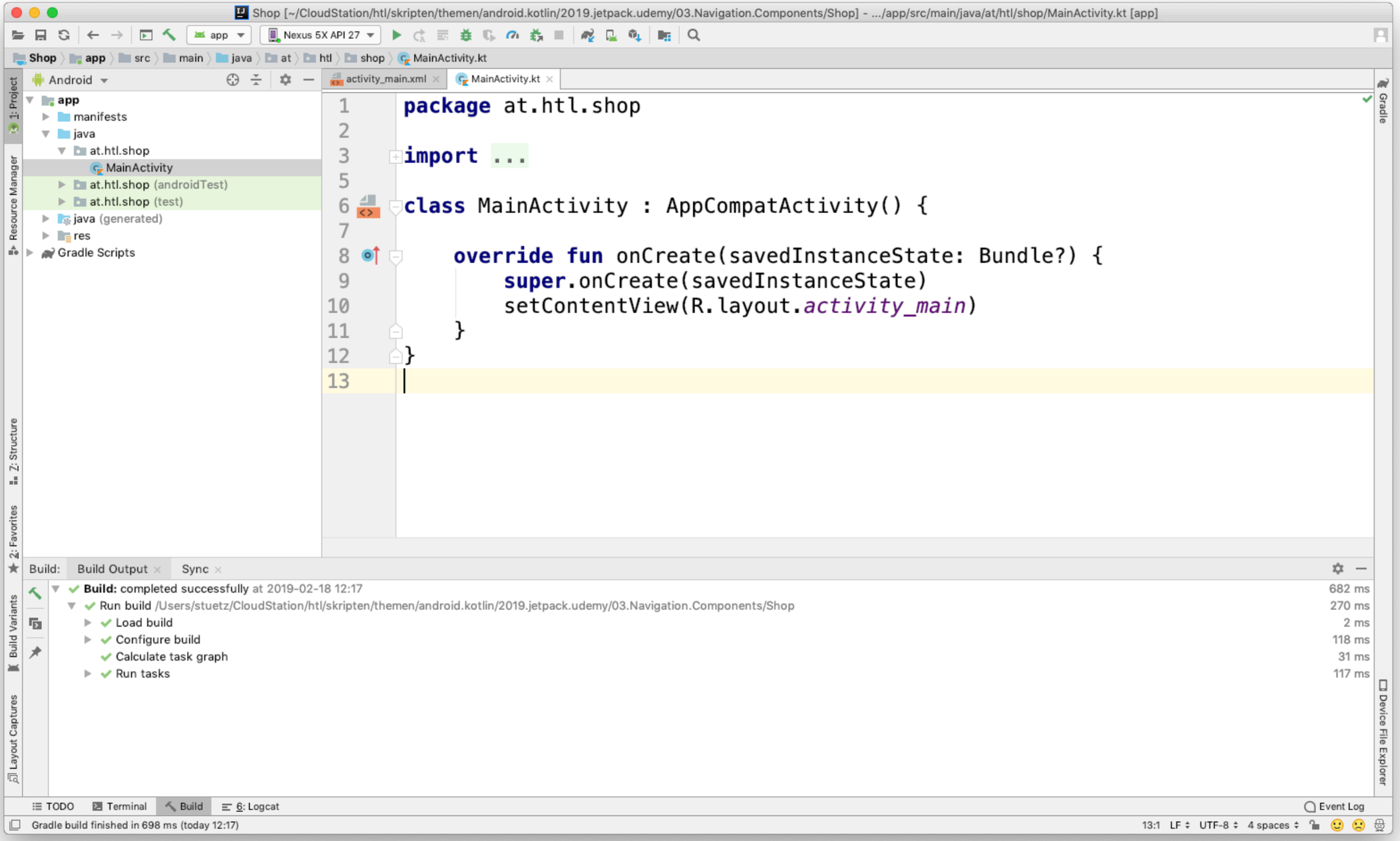
Thomas W. Stütz

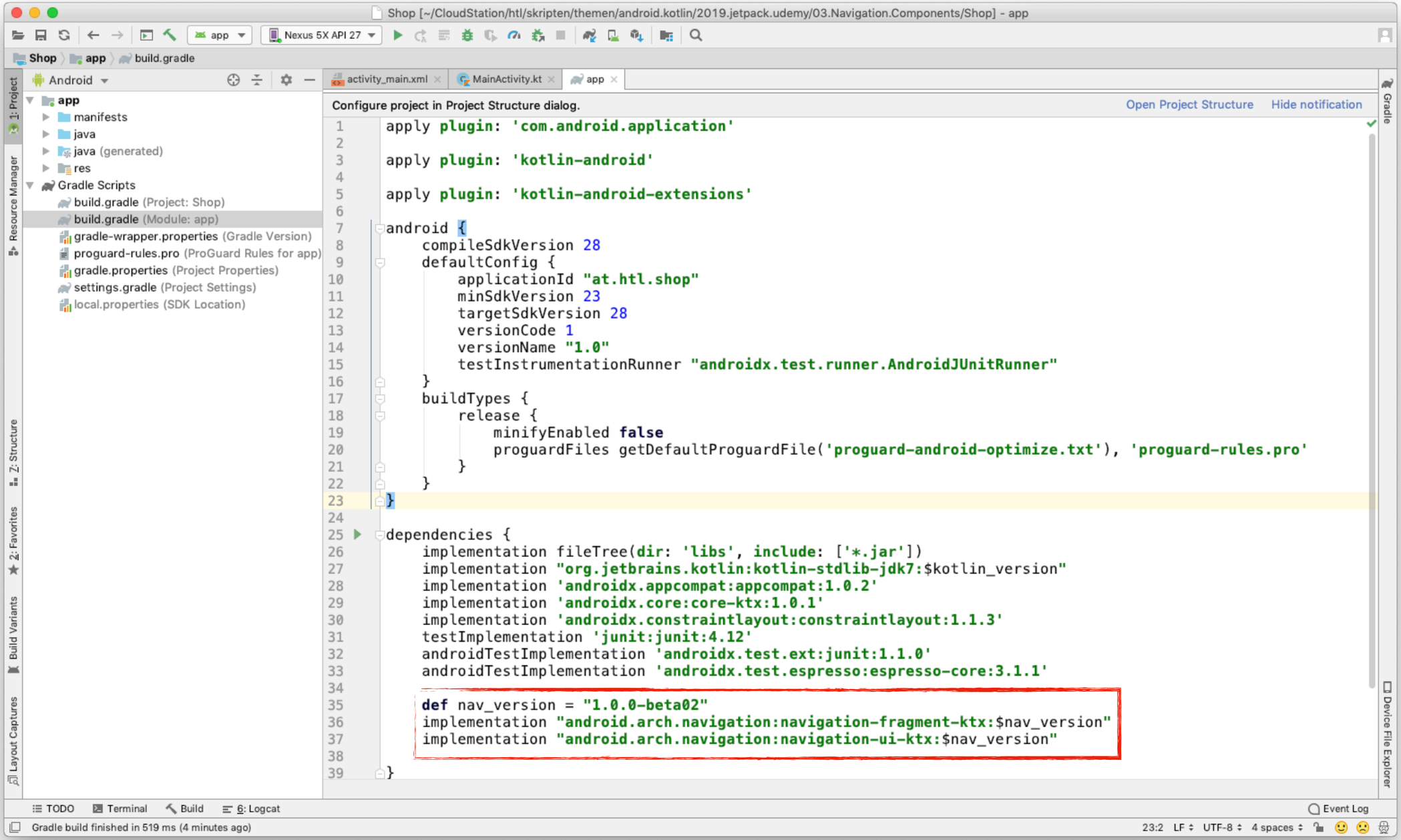
# Features of Navigation Components

- Handling fragment transaction
- Handling Up and Back actions correctly
- Providing standardized resources for animations and transitions
- Including Navigation UI patterns
- Providing type safety when passing information
- Visualizing and editing navigation from Android Studio Navigation Editor


Create New Project
Choose your project
Phone and Tablet
Wear OS
TV
Android Auto
Android Things
Add No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Fullscreen Activity
Master/Detail Flow
Navigation Drawer Activity
Google Maps Activity
Empty Activity
Creates a new empty activity
Cancel
Previous
Next
Finish


Create New Project
Configure your project
Name
Shop
Package name
at.htl.shop
Save location
droid.kotlin/2019.jetpack.udemy/03.Navigation.Components/Shop
Language
Kotlin
Minimum API level
API 23: Android 6.0 (Marshmallow)
Your app will run on approximately 62.6% of devices.
Help me choose
This project will support instant apps
Use AndroidX artifacts
Empty Activity
Creates a new empty activity
Cancel
Previous
Next
Finish

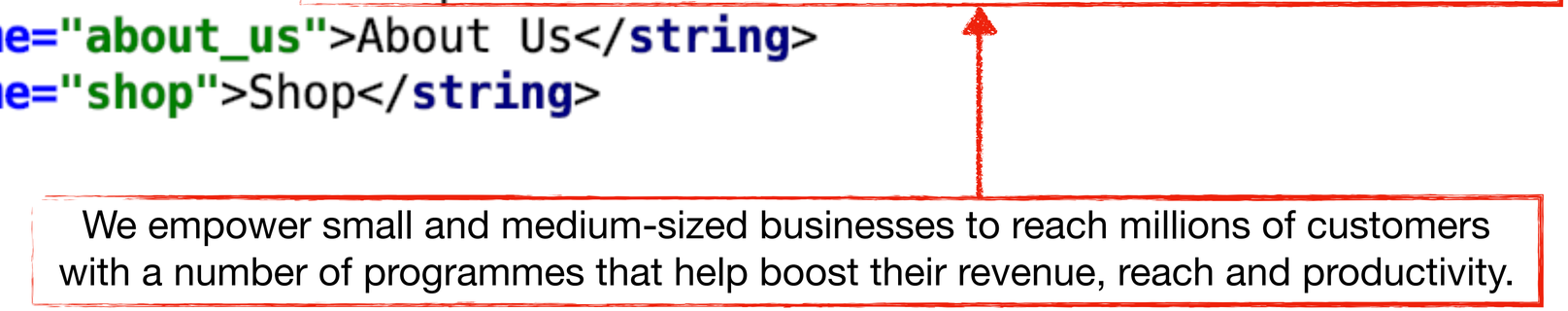

Shop [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/03.Navigation.Components/Shop] - .../app/src/main/java/at/htl/shop/MainActivity.kt [app]
app
Nexus 5X API 27
Shop
app
src
main
java
at
htl
shop
MainActivity.kt
Android
activity_main.xml
MainActivity.kt
app
manifests
java
at.htl.shop
MainActivity
at.htl.shop (androidTest)
at.htl.shop (test)
java (generated)
res
Gradle Scripts
1 package at.htl.shop
2
3 import ...
5
6 class MainActivity : AppCompatActivity() {
7
8 override fun onCreate(savedInstanceState: Bundle?) {
9 super.onCreate(savedInstanceState)
10 setContentView(R.layout.activity_main)
11 }
12 }
13
Build:
Build Output
Sync
Build: completed successfully at 2019-02-18 12:17
682 ms
Run build /Users/stuetz/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/03.Navigation.Components/Shop
270 ms
Load build
2 ms
Configure build
118 ms
Calculate task graph
31 ms
Run tasks
117 ms
TODO
Terminal
Build
6: Logcat
Event Log
Gradle build finished in 698 ms (today 12:17)
13:1
LF
UTF-8
4 spaces
1: Project
Resource Manager
7: Structure
2: Favorites
Build Variants
Layout Captures
Gradle
Device File Explorer




https://developer.android.com/jetpack/androidx/releases/navigation

# values/strings.xml

```xml
<resources>
    <string name="app_name">Shop</string>
    <string name="home">Home</string>
    <string name="cart">Cart</string>
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <string name="detail">We empower small and medium-sized businesses t
    <string name="about_us">About Us</string>
    <string name="shop">Shop</string>
</resources>
```



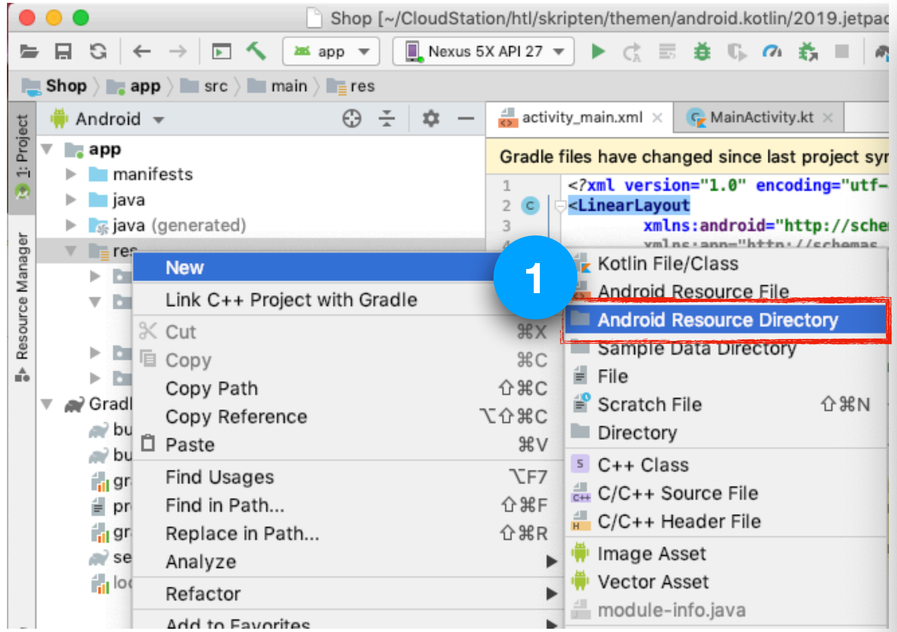
# values/styles.xml

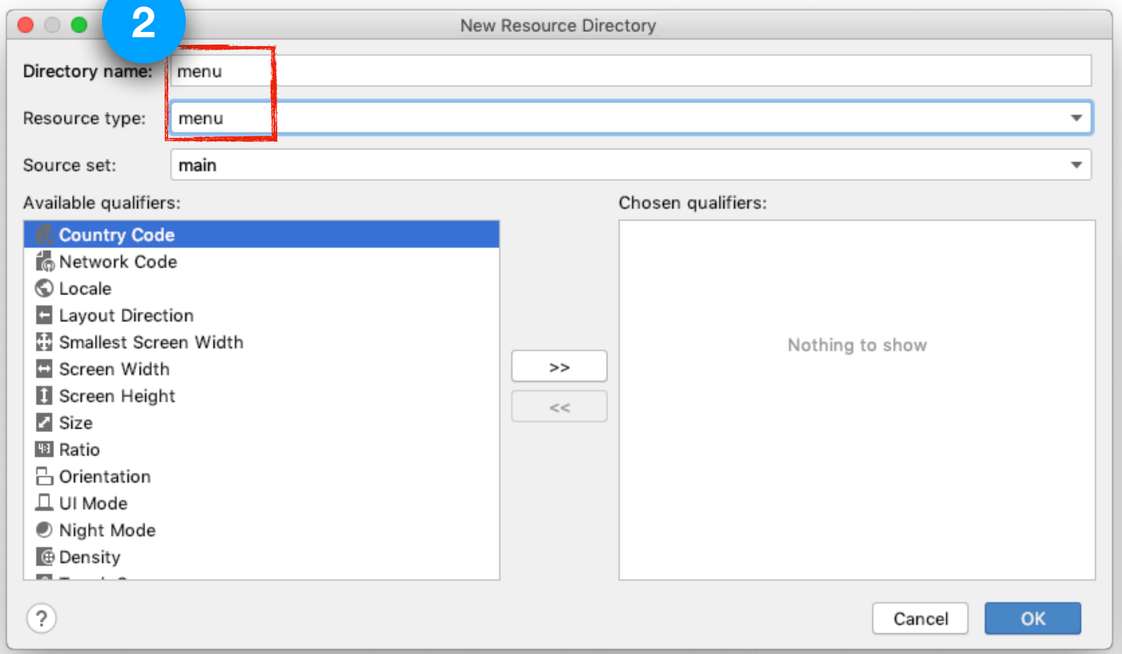
```xml
<resources>

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

</resources>
```

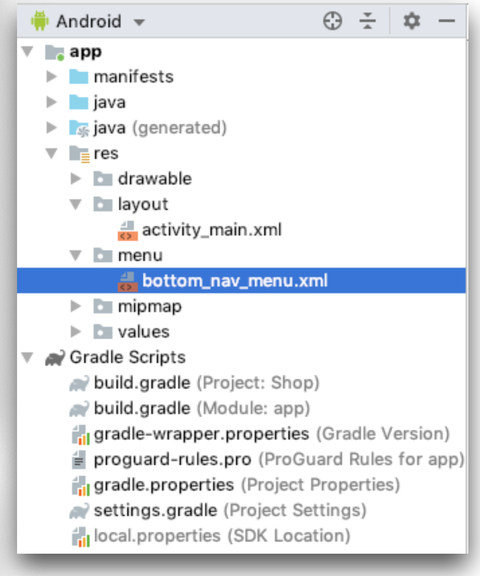
# values/colors.xml

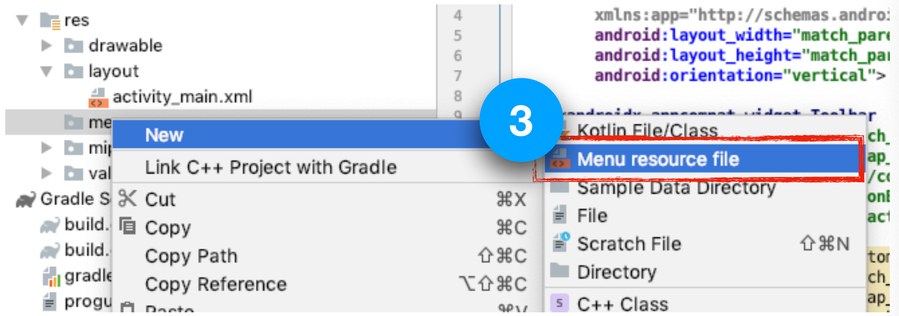
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

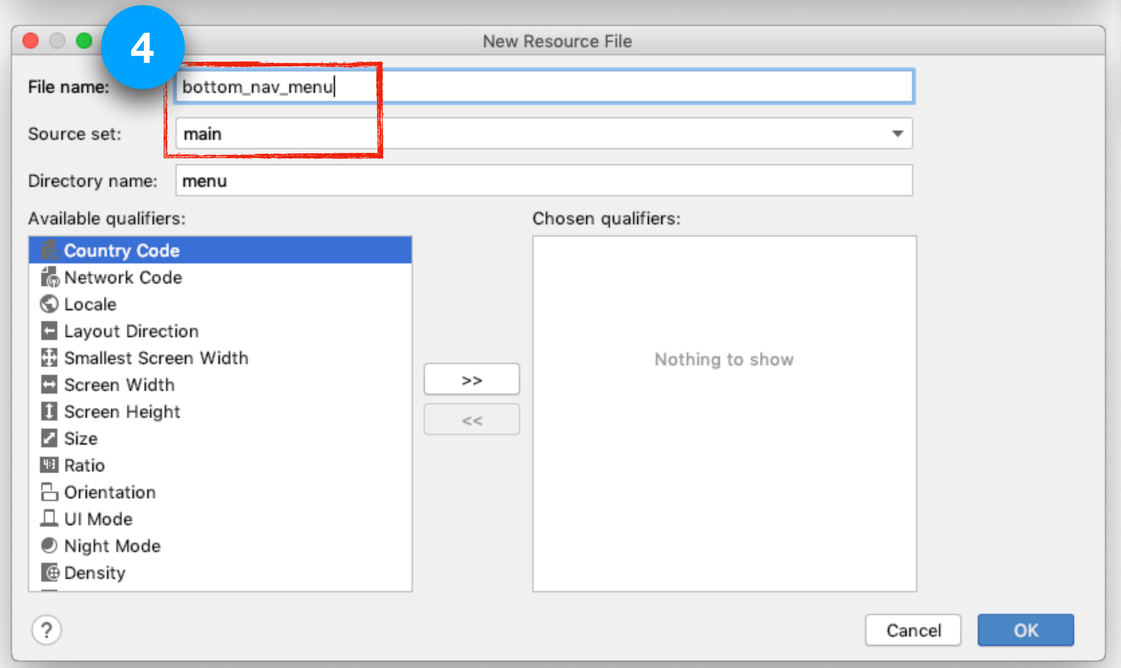
# Erstellen des Menü-Ressourcenordners und eines Menüs



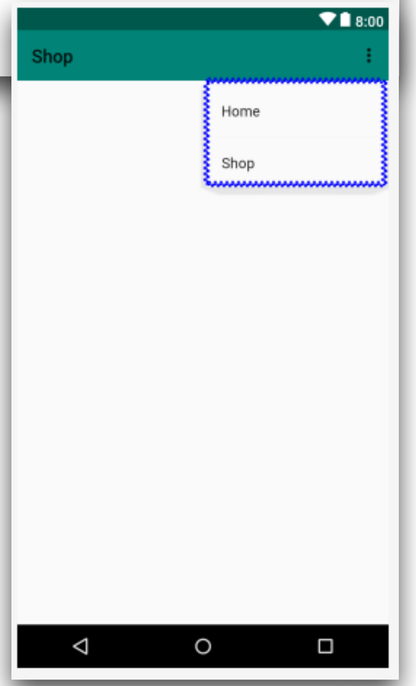



# Menüs

bottom_nav_menu.xml

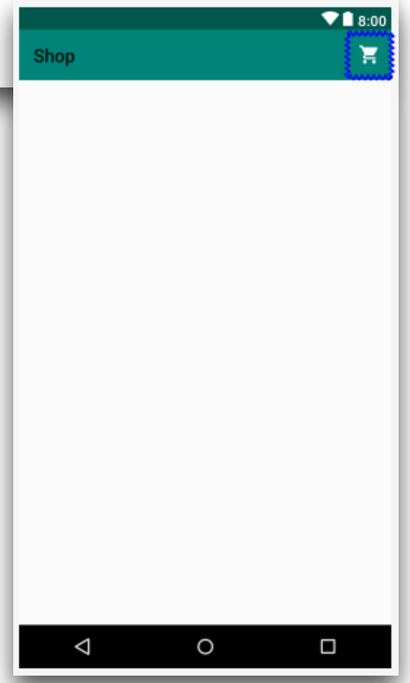
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
            android:id="@+id/home_destination"
            android:title="Home"
            android:icon="@drawable/ic_home"
    />

    <item
            android:id="@+id/shop_destination"
            android:title="Shop"
            android:icon="@drawable/ic_shop"
    />
</menu>
```



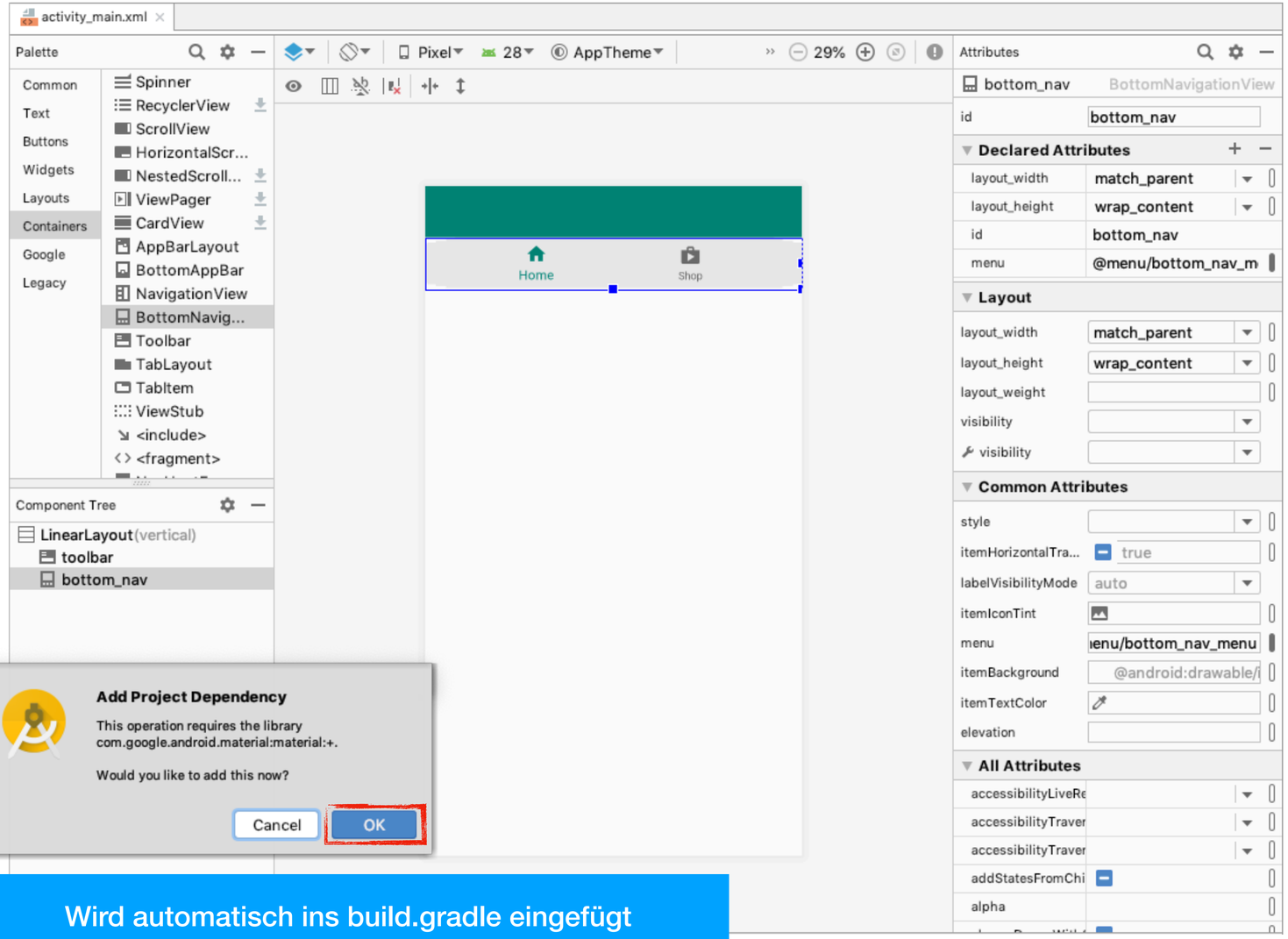

Erstellen Sie ein weiteres Menü: toolbar_menu.xml

toolbar_menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
            android:id="@+id/cart_destination"
            android:icon="@drawable/ic_shopping_cart"
            android:title="Cart"
            app:showAsAction="ifRoom"
    />
</menu>
```



# activity_main.xml

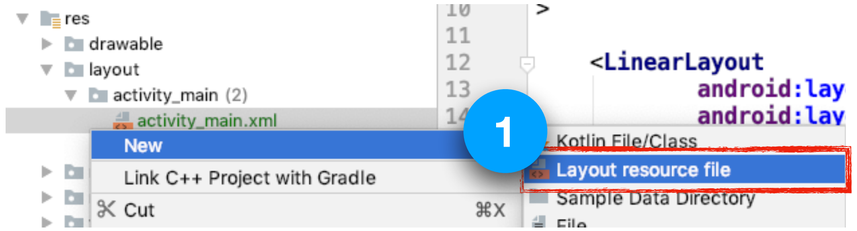


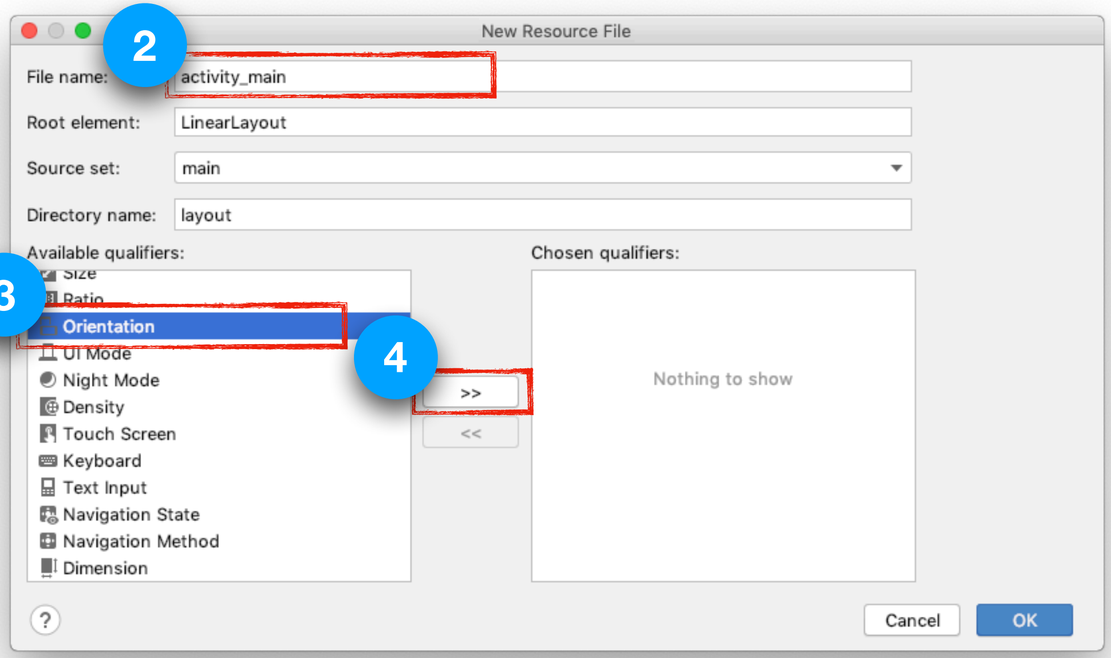

Wird automatisch ins build.gradle eingefügt

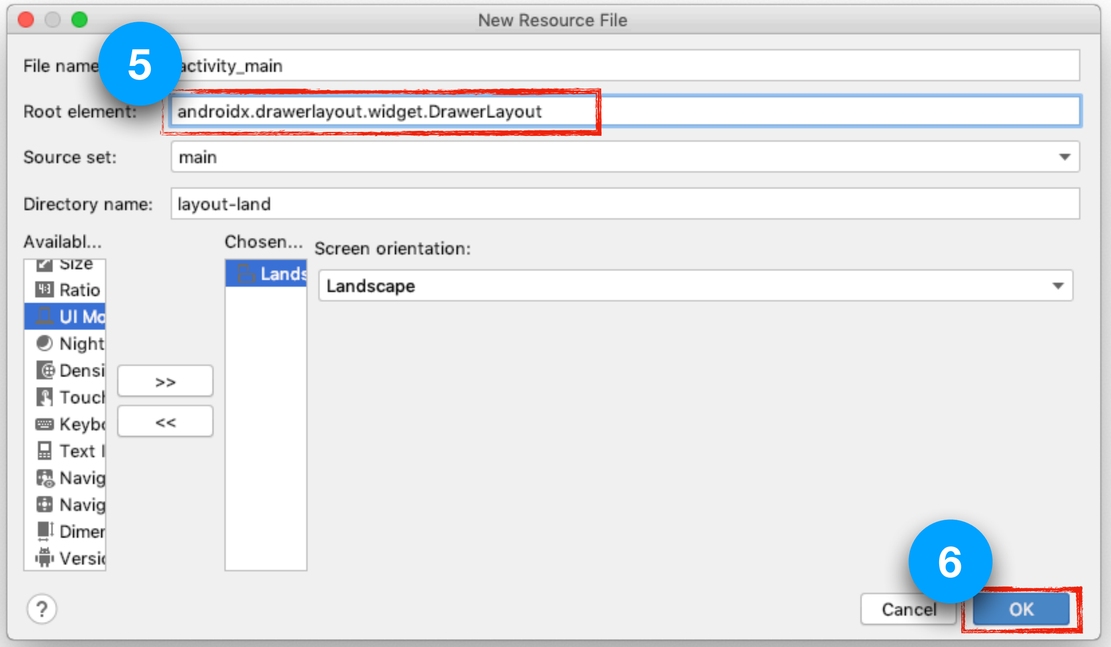
```
implementation 'com.google.android.material:material:1.0.0'
```

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

    <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary"
            android:theme="@style/ThemeOverlay.AppCompat.Dark"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
            android:id="@+id/bottom_nav"
            app:menu="@menu/bottom_nav_menu"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

</LinearLayout>
```

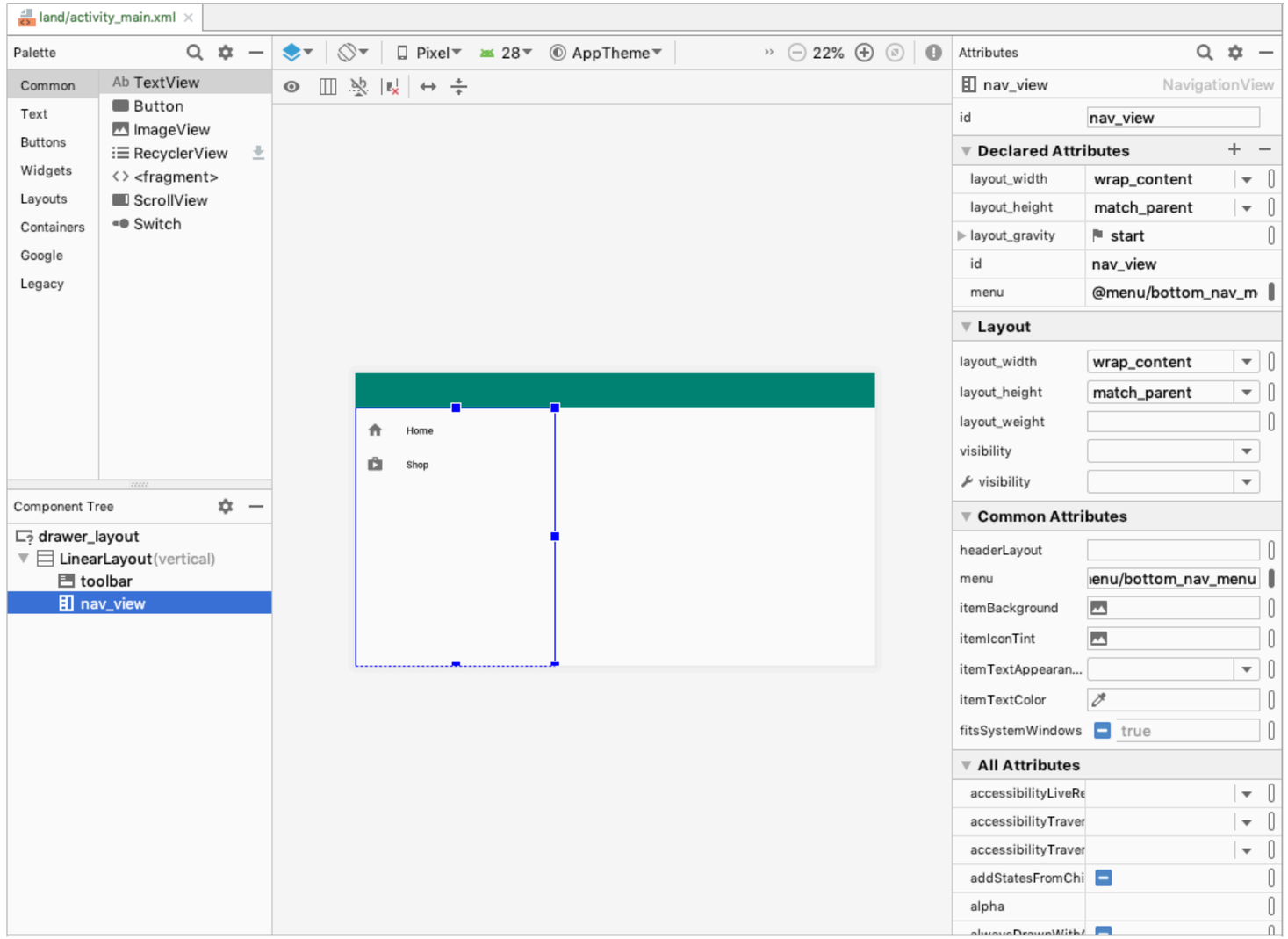
# activity_main für Landscape erstellen



# land/activity_main.xml

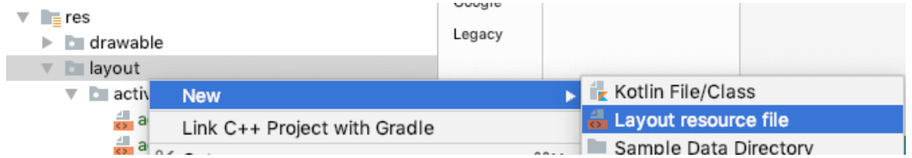


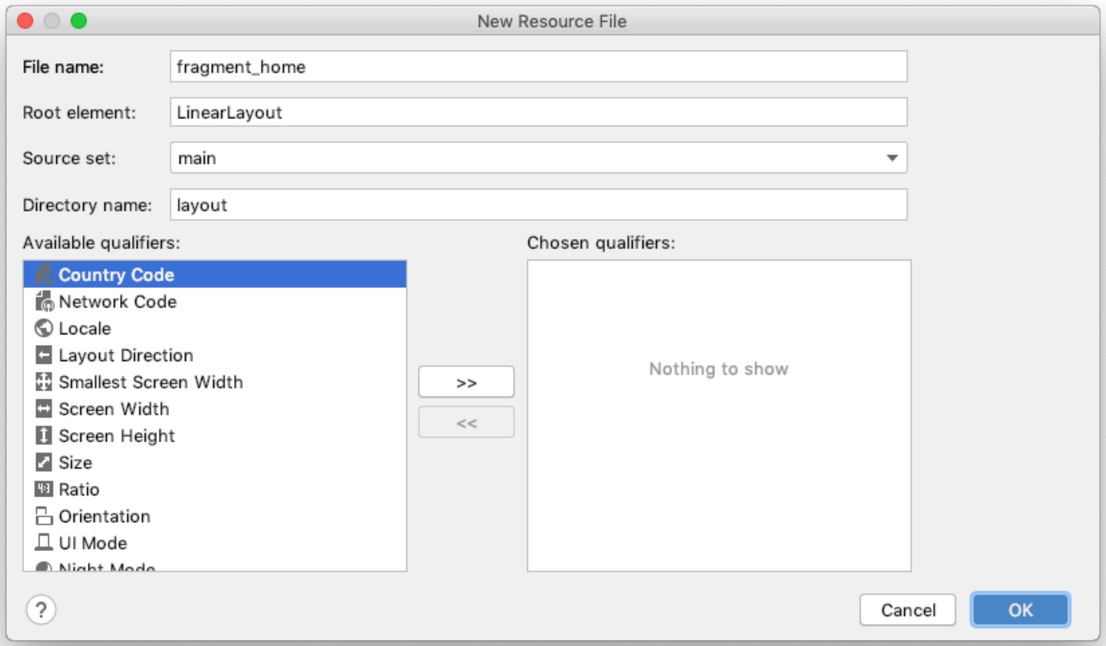

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

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

        <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/colorPrimary"
                android:theme="@style/ThemeOverlay.AppCompat.Dark"/>

        <com.google.android.material.navigation.NavigationView
                android:id="@+id/nav_view"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:layout_gravity="start"
                app:menu="@menu/bottom_nav_menu"/>
    </LinearLayout>

</androidx.drawerlayout.widget.DrawerLayout>
```

**Was sind DrawerLayouts?**

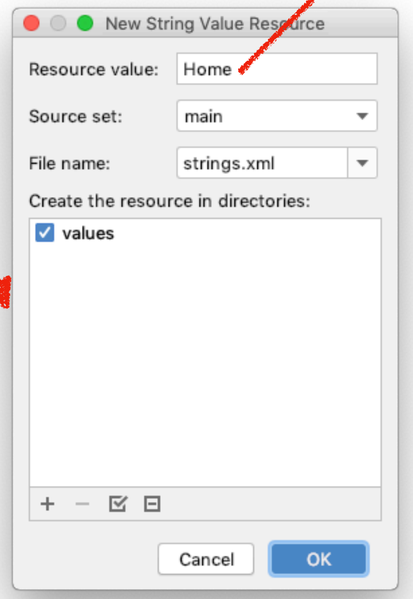
https://youtu.be/DkT0vS14Um0
Implementing_an_Android_Navigation_Drawer_in_Kotlin

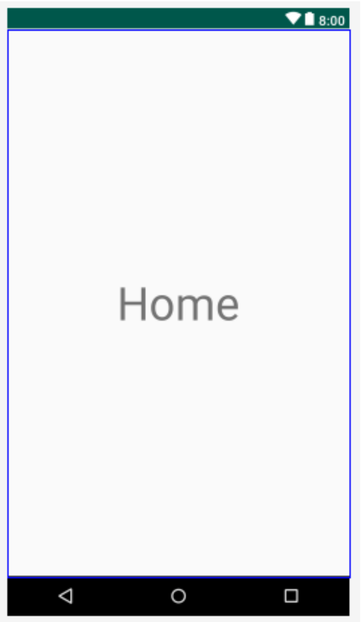
# fragment_home.xml




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

    <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="55sp"
            android:text="@string/home" />
</LinearLayout>
```

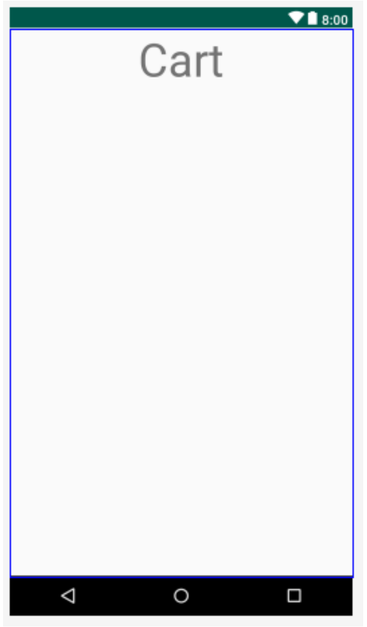



**Show Intention Actions** via ⌥⏎ (Alt+Enter for Win/Linux)

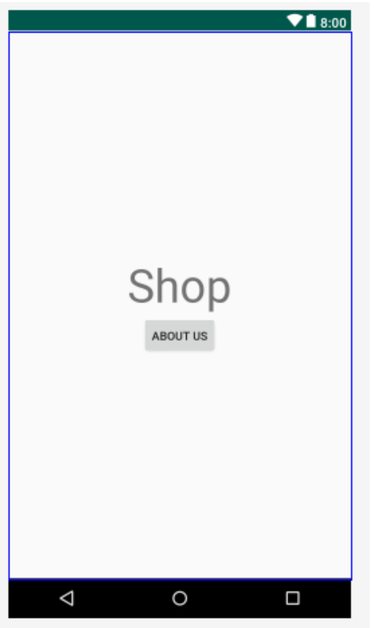
# fragment_cart.xml

```
<?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="match_parent">

    <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/cart"
            android:textSize="55sp"/>
</LinearLayout>
```



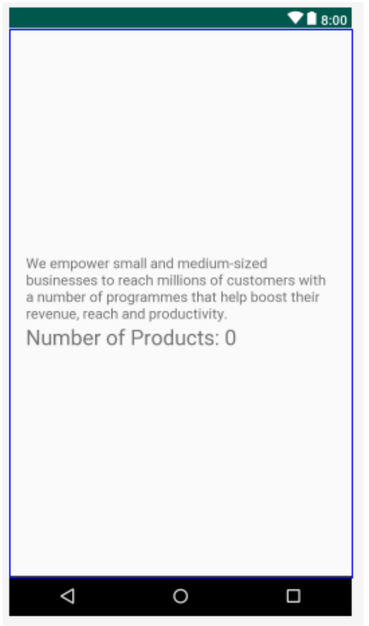
# fragment_shop.xml

```
<?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="match_parent"
        android:gravity="center">

    <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Shop"
            android:textSize="55sp"/>

    <Button
            android:id="@+id/btn_about"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="About Us"/>

</LinearLayout>
```

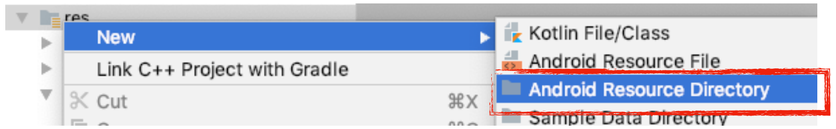


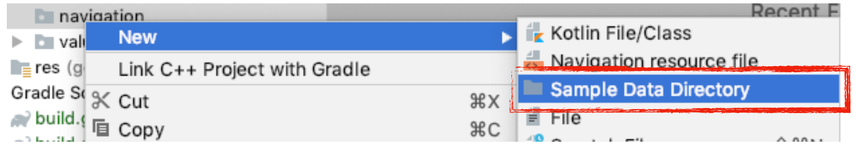
# fragment_about.xml

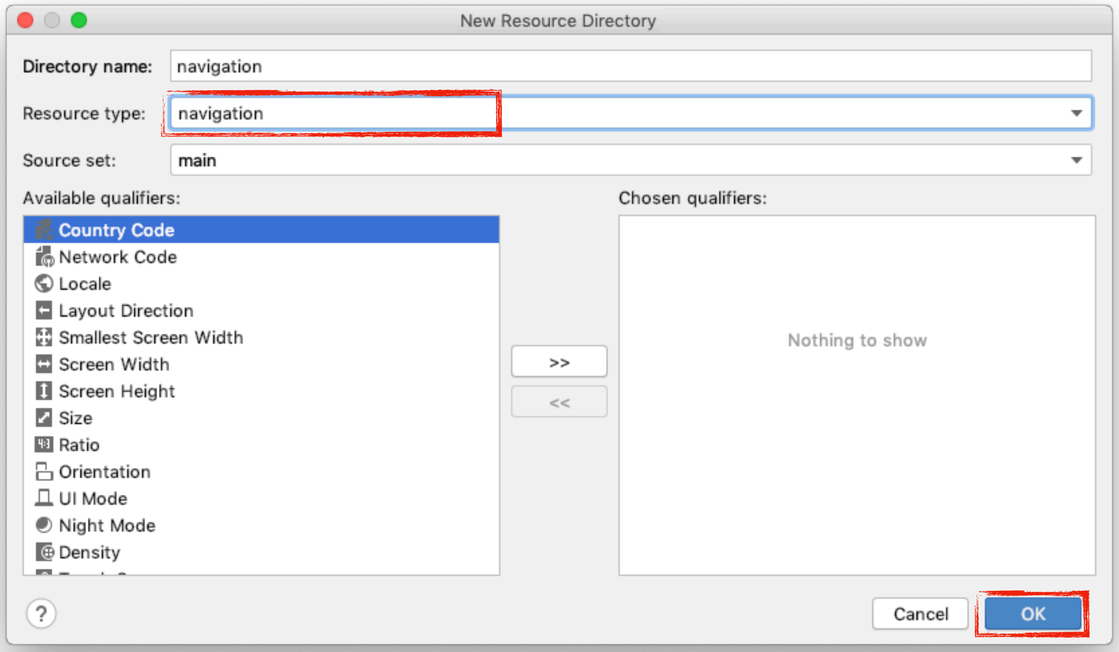
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="20dp">

    <TextView
            android:id="@+id/tv_details"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="start"
            android:text="@string/detail"
            android:textSize="17sp"/>

    <TextView
            android:id="@+id/tv_product_count"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="start"
            android:text="Number of Products: 0"
            android:textSize="25sp"/>

</LinearLayout>
```

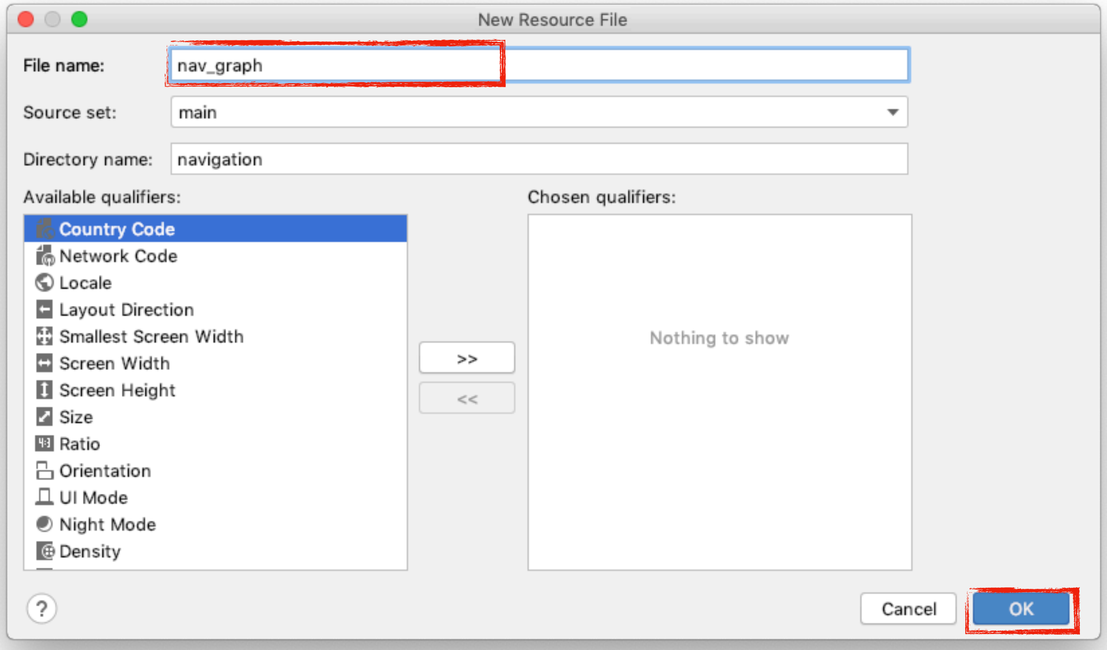


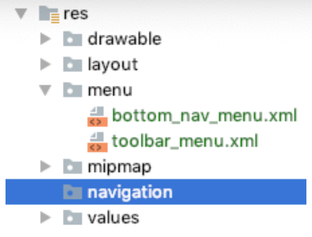
# Erstellen des Navigations-Files

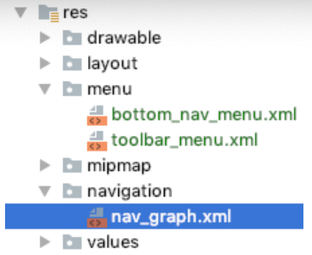


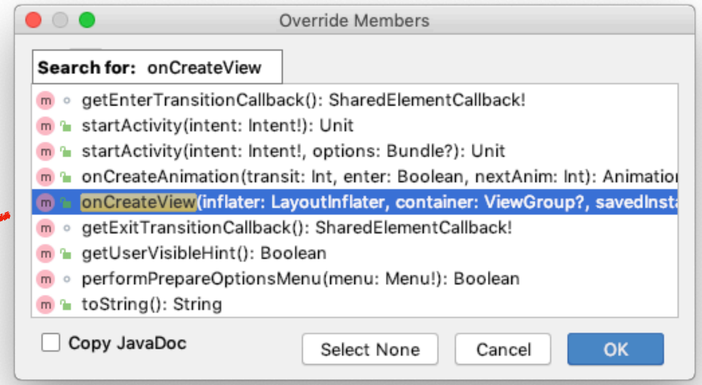



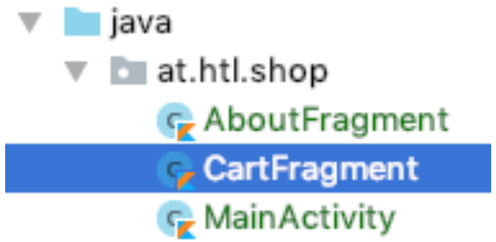



# AboutFragment.kt


```
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class AboutFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_about, container, false)
    }

}
```

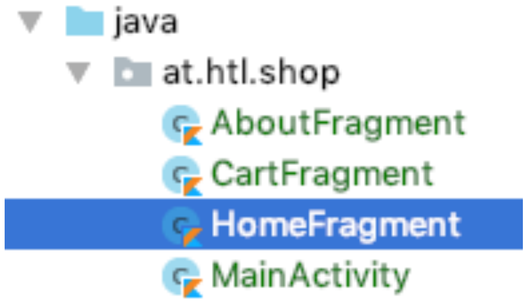


# CartFragment.kt




```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class CartFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_cart, container, false)
    }
}
```

# HomeFragment.kt




```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class HomeFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false)
    }

}
```

# ShopFragment.kt


```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment

class ShopFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_shop, container, false)
    }

}
```

# Navigation Components

- Navigation graph – contains all navigated information in centralized location
- NavHostFragment – special widget to add in layout
- NavController – Kotlin or Java object that keeps track of current position

android-jetpack-navigation-components-shop.v01.zip

# NavHostFragment hinzufügen

1

2

3

4

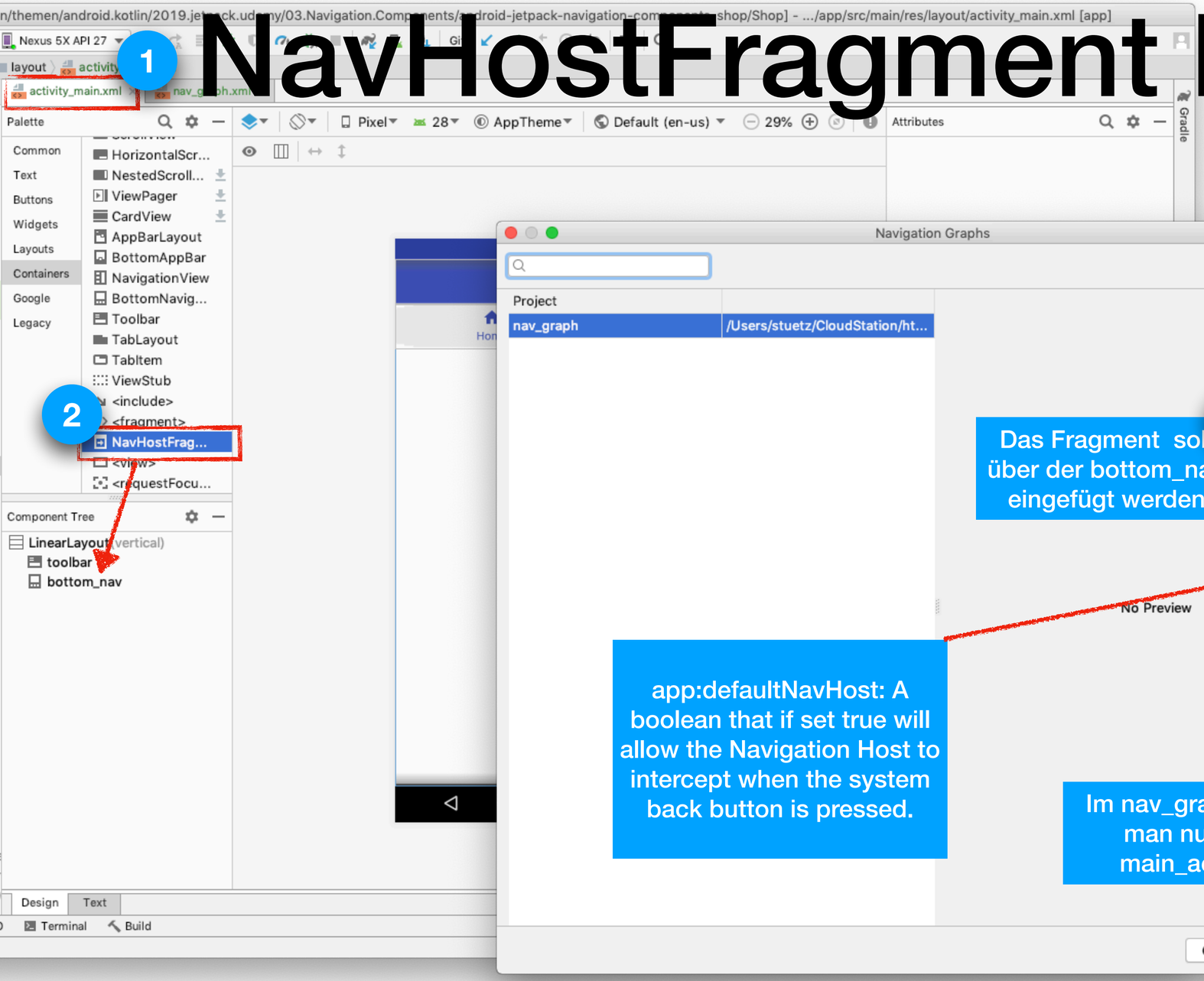
5

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

    <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary"
            android:theme="@style/ThemeOverlay.AppCompat.Dark"/>

    <fragment
            android:id="@+id/nav_host_fragment"
            android:name="androidx.navigation.fragment.NavHostFragment"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            app:defaultNavHost="true"
            app:navGraph="@navigation/nav_graph"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
            android:id="@+id/bottom_nav"
            app:menu="@menu/bottom_nav_menu"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

</LinearLayout>
```

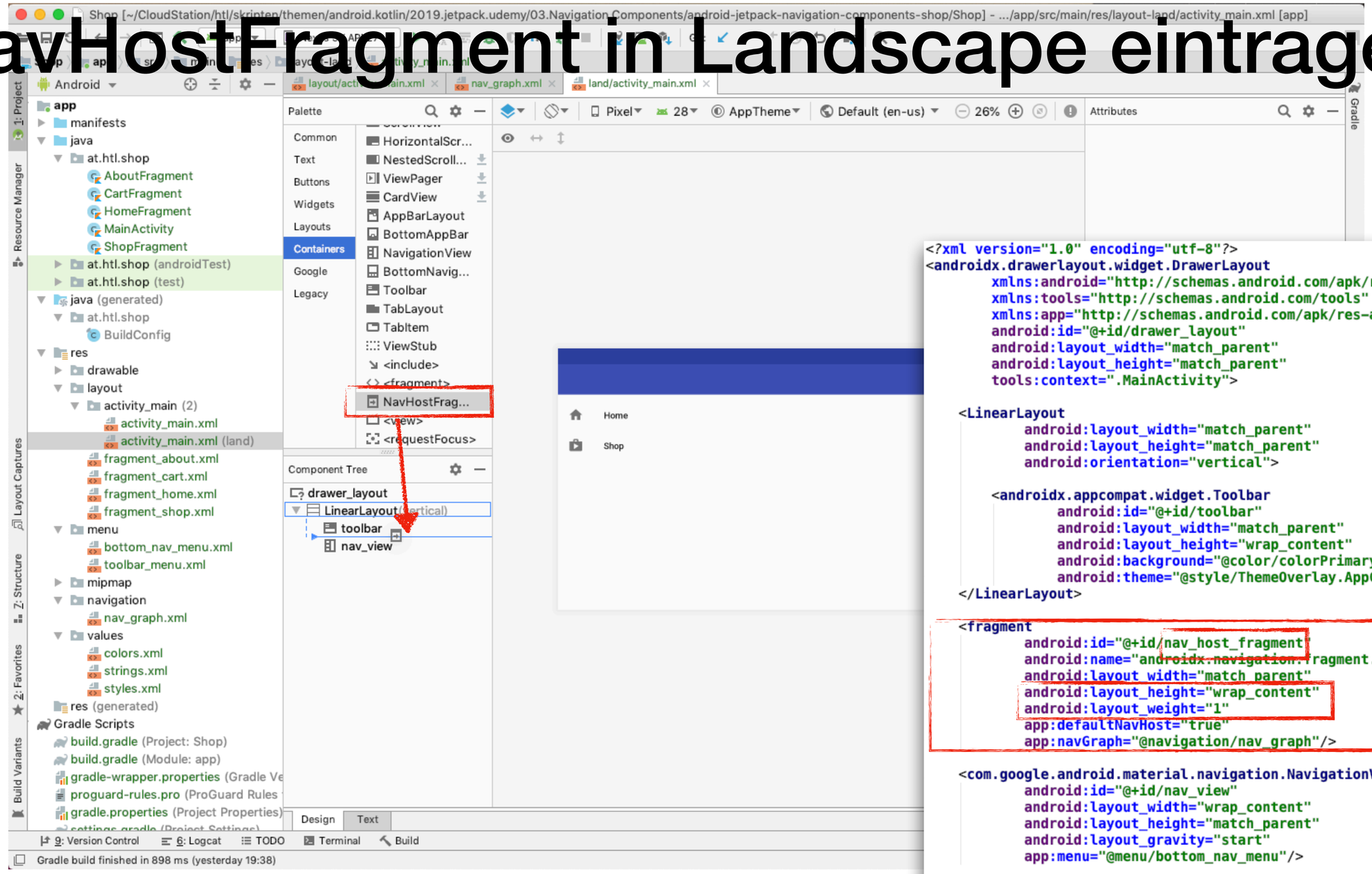
Das Fragment soll über der bottom_nav eingefügt werden

app:defaultNavHost: A boolean that if set true will allow the Navigation Host to intercept when the system back button is pressed.

Im nav_graph sieht man nun die main_activity

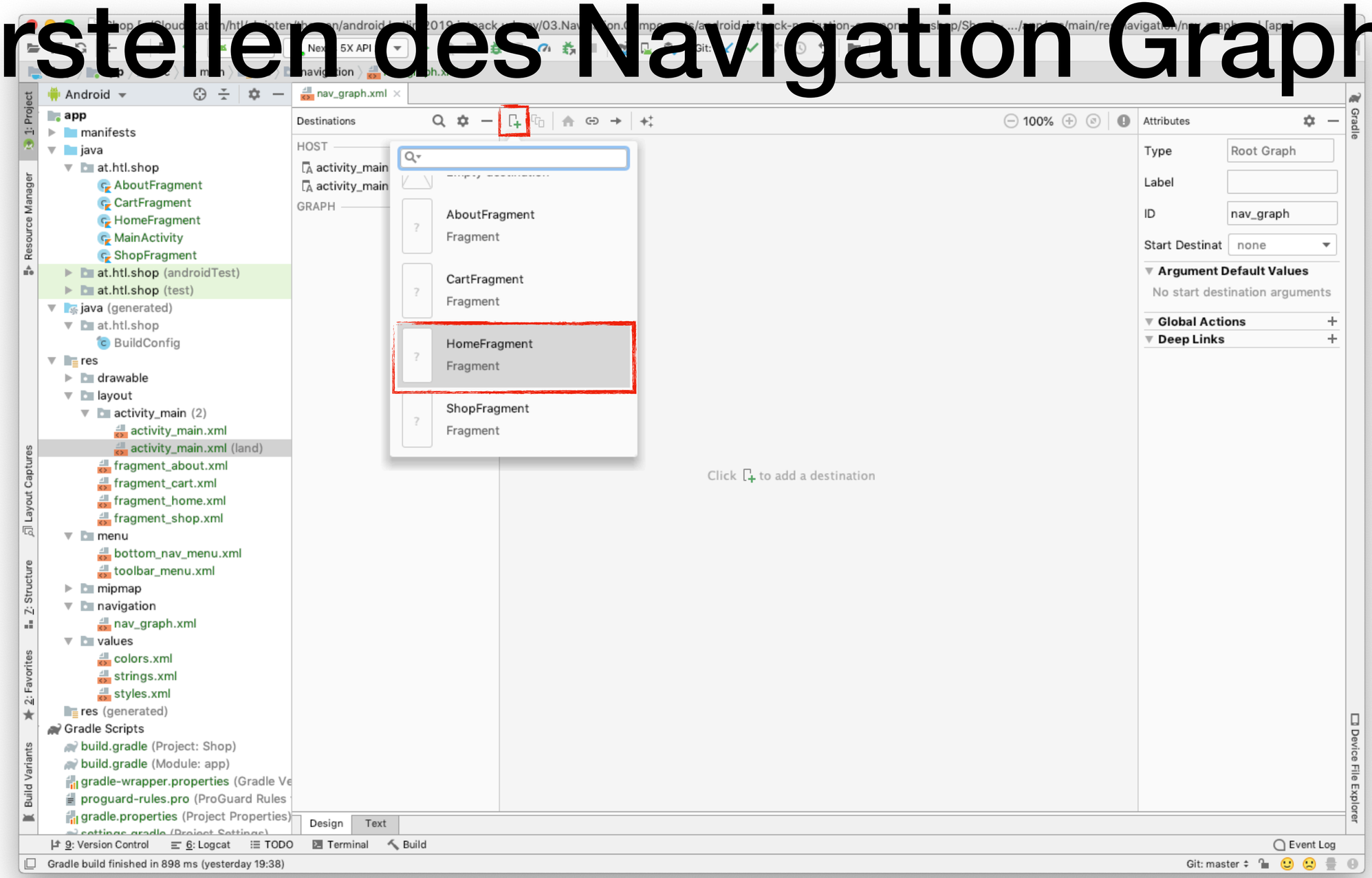


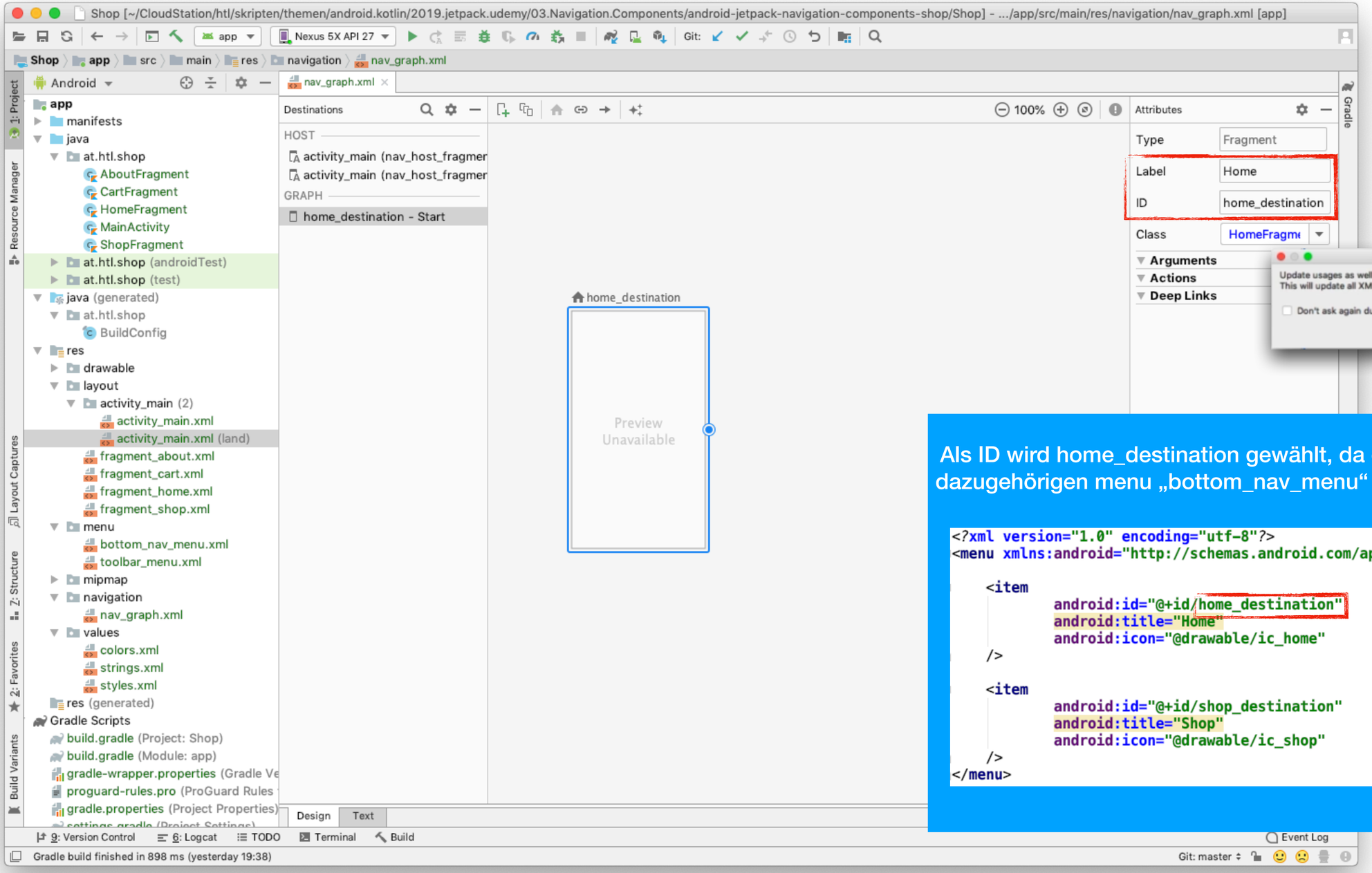

# NavHostFragment in Landscape eintragen




```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
        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:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

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

        <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/colorPrimary"
                android:theme="@style/ThemeOverlay.AppCompat.Dark"/>
    </LinearLayout>

    <fragment
            android:id="@+id/nav_host_fragment"
            android:name="androidx.navigation.fragment.NavHostFragment"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            app:defaultNavHost="true"
            app:navGraph="@navigation/nav_graph"/>

    <com.google.android.material.navigation.NavigationView
            android:id="@+id/nav_view"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            app:menu="@menu/bottom_nav_menu"/>

</androidx.drawerlayout.widget.DrawerLayout>
```

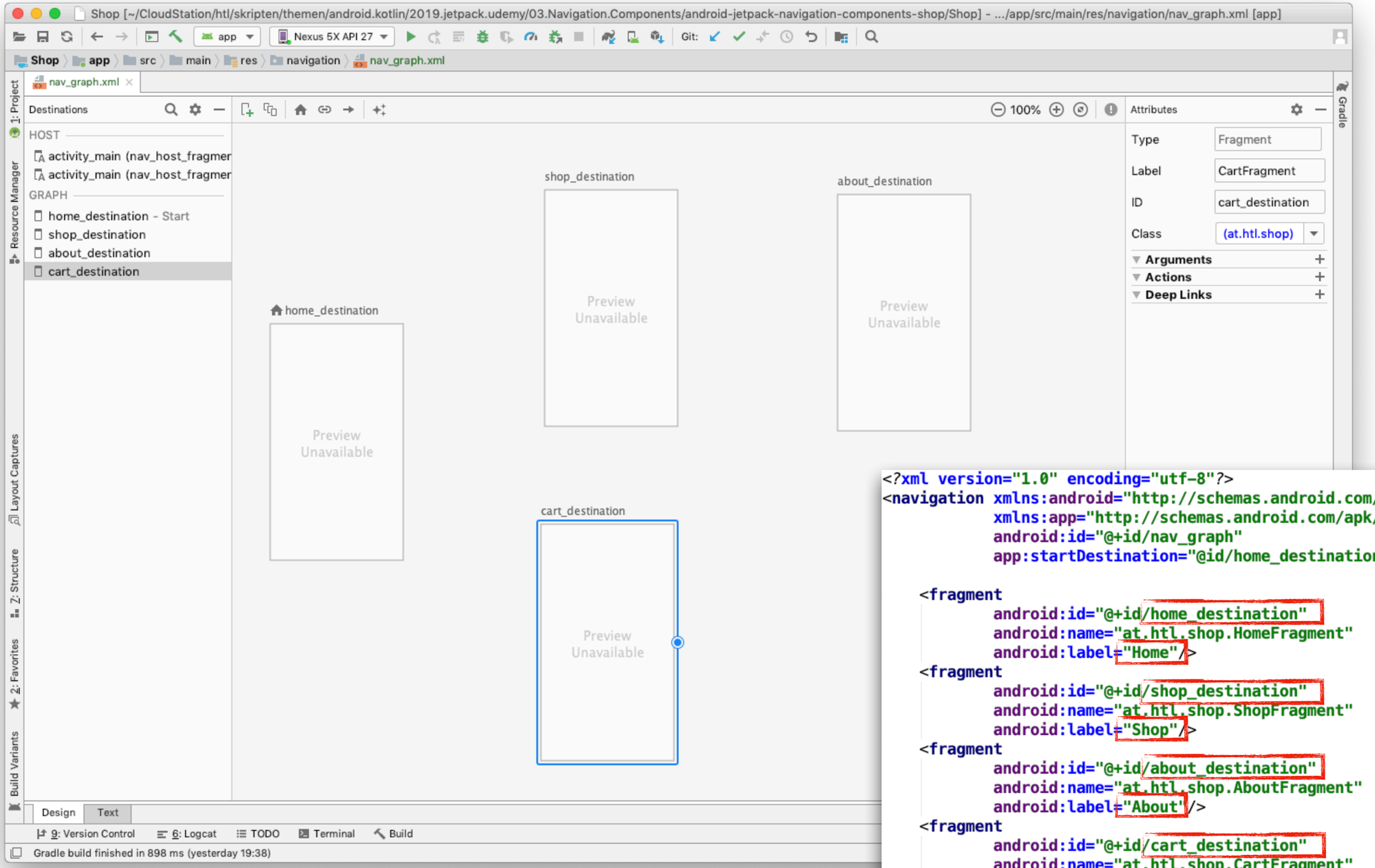
# Erstellen des Navigation Graphs






Als ID wird home_destination gewählt, da dies bereits im dazugehörigen menu „bottom_nav_menu“ eingetragen ist

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
            android:id="@+id/home_destination"
            android:title="Home"
            android:icon="@drawable/ic_home"
    />

    <item
            android:id="@+id/shop_destination"
            android:title="Shop"
            android:icon="@drawable/ic_shop"
    />
</menu>
```




```xml
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/nav_graph"
            app:startDestination="@id/home_destination">

    <fragment
            android:id="@+id/home_destination"
            android:name="at.htl.shop.HomeFragment"
            android:label="Home" />
    <fragment
            android:id="@+id/shop_destination"
            android:name="at.htl.shop.ShopFragment"
            android:label="Shop" />
    <fragment
            android:id="@+id/about_destination"
            android:name="at.htl.shop.AboutFragment"
            android:label="About" />
    <fragment
            android:id="@+id/cart_destination"
            android:name="at.htl.shop.CartFragment"
            android:label="Cart" />
</navigation>
```

# MainActivity.kt 1

In der MainActivity werden folgende Komponenten konfiguriert:
BottomNavigationBar
Toolbar
NavigationDrawer

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val navController =
            Navigation.findNavController(this, R.id.nav_host_fragment)

        setUpBottomNav(navController)
        setUpSideNav(navController)
        setUpActionBar(navController)
    }

    private fun setUpBottomNav(navController: NavController) {

    }

    private fun setUpSideNav(navController: NavController) {

    }

    private fun setUpActionBar(navController: NavController) {

    }
}
```

Lassen Sie sich die Methoden generieren

# MainActivity.kt 2

activity_main.xml

```
<com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_nav"
        app:menu="@menu/bottom_nav_menu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```

Der safe-call-Operator '?.' wird verwendet, da das Landscape-Layout kein bottom_nav enthält

https://kotlinlang.org/docs/reference/null-safety.html#safe-calls

**NavigationUI**

Class which hooks up elements of your application such as global navigation patterns like a navigation drawer or bottom nav bar with your NavController.

```
private fun setUpBottomNav(navController: NavController) {
    bottom_nav?.let {
        NavigationUI.setupWithNavController(it, navController)
    }
}

private fun setUpSideNav(navController: NavController) {
    nav_view?.let {
        NavigationUI.setupWithNavController(it, navController)
    }
}

private fun setUpActionBar(navController: NavController) {
    NavigationUI.setupActionBarWithNavController(
        this,
        navController,
        drawer_layout
    )
}
```

# let { ... }

### 2. This vs. it argument

If we look at `T.run` and `T.let`, both functions are similar except for one thing, the way they accept the argument. The below shows the same logic for both functions.

```
stringVariable?.run {
      println("The length of this String is $length")
}

// Similarly.

stringVariable?.let {
      println("The length of this String is ${it.length}")
}
```

If you check the `T.run` function signature, you'll notice the `T.run` is just made as extension function calling `block: T.()`. Hence all within the scope, the `T` could be referred as `this`.In programming, `this` could be omitted most of the time. Therefore in our example above, we could use `$length` in the `println` statement, instead of `${this.length}`. I call this as sending in ***this as argument***.

However for `T.let` function signature, you'll notice that `T.let` is sending itself into the function i.e. `block: (T)`. Hence this is like a lambda argument sent it. It could be referred within the scope function as `it`. So I call this as sending in ***it as argument***.

https://medium.com/@elye.project/mastering-kotlin-standard-functions-run-with-let-also-and-apply-9cd334b0ef84

# MainActivity.kt 3

Erstellen des Option-Menus

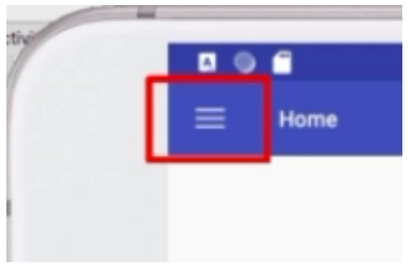
```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.toolbar_menu, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem?): Boolean {
    val navController = Navigation.findNavController(this, R.id.nav_host_fragment)
    val navigated = NavigationUI.onNavDestinationSelected(item!!, navController)
    return navigated || super.onOptionsItemSelected(item)
}
```

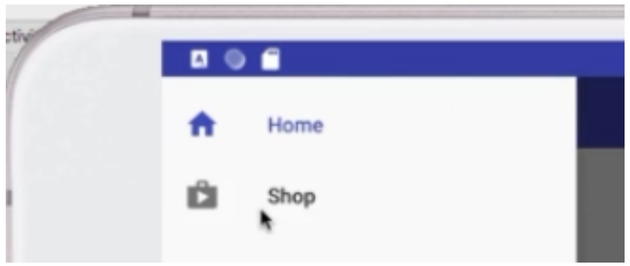
„item is not null“
Es wird ein NPE geworfen, falls item doch null ist

https://kotlinlang.org/docs/reference/null-safety.html#the--operator

navigateUp sorgt dafür, dass der Hamburger durch einen Pfeil (icon ‘up’) ersetzt wird, wenn zu einem anderen Fragment gewechselt wird

```
override fun onSupportNavigateUp(): Boolean {
    return NavigationUI.navigateUp(
        Navigation.findNavController(this,R.id.nav_host_fragment), drawer_layout
    )
}
```




# MainActivity.kt 4

```kotlin
package at.htl.shop

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import androidx.navigation.NavController
import androidx.navigation.Navigation
import androidx.navigation.ui.NavigationUI
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(toolbar)

        val navController =
            Navigation.findNavController(this, R.id.nav_host_fragment)

        setUpBottomNav(navController)
        setUpSideNav(navController)
        setUpActionBar(navController)
    }

    private fun setUpBottomNav(navController: NavController) {
        bottom_nav?.let {
            NavigationUI.setupWithNavController(it, navController)
        }
    }

    private fun setUpSideNav(navController: NavController) {
        nav_view?.let {
            NavigationUI.setupWithNavController(it, navController)
        }
    }

    private fun setUpActionBar(navController: NavController) {
        NavigationUI.setupActionBarWithNavController(
            this,
            navController,
            drawer_layout
        )
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.toolbar_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        val navController = Navigation.findNavController(this, R.id.nav_host_fragment)
        val navigated = NavigationUI.onNavDestinationSelected(item!!, navController)
        return navigated || super.onOptionsItemSelected(item)
    }

    override fun onSupportNavigateUp(): Boolean {
        return NavigationUI.navigateUp(
            Navigation.findNavController(this,R.id.nav_host_fragment), drawer_layout
        )
    }
}
```

# Überblick

MainActivity.kt

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    setSupportActionBar(toolbar)

    val navController =
        Navigation.findNavController(this, R.id.nav_host_fragment)

    setUpBottomNav(navController)
    setUpSideNav(navController)
    setUpActionBar(navController)
}

private fun setUpBottomNav(navController: NavController) {
    bottom_nav?.let {
        NavigationUI.setupWithNavController(it, navController)
    }
}

private fun setUpSideNav(navController: NavController) {
    nav_view?.let {
        NavigationUI.setupWithNavController(it, navController)
    }
}

private fun setUpActionBar(navController: NavController) {
    NavigationUI.setupActionBarWithNavController(
        this,
        navController,
        drawer_layout
    )
}
```

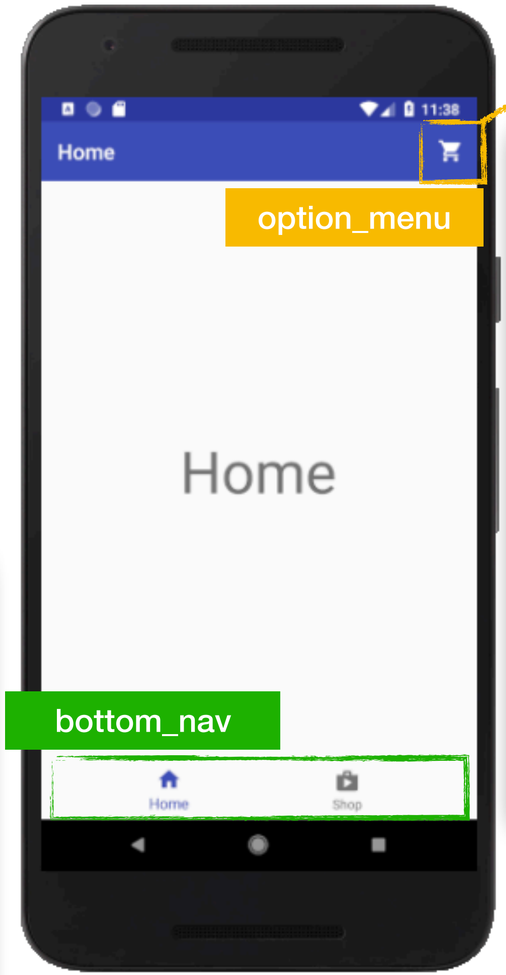
In der MainActivity werden die Komponenten konfiguriert

bezieht sich auf die Landscape activity_main.xml

`android:id="@+id/drawer_layout"`

bottom_nav_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/home_destination"
        android:title="Home"
        android:icon="@drawable/ic_home"
    />

    <item
        android:id="@+id/shop_destination"
        android:title="Shop"
        android:icon="@drawable/ic_shop"
    />
</menu>
```




option_menu

bottom_nav

toolbar_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/cart_destination"
        android:icon="@drawable/ic_shopping_cart"
        android:title="Cart"
        app:showAsAction="ifRoom"
    />
</menu>
```

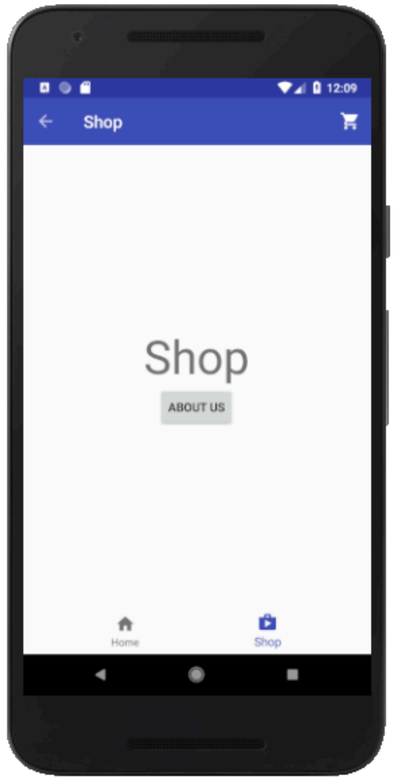
nav_graph.xml

```
<?xml version="1.0" encoding="utf-8"?>
<navigation
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/nav_graph"
        app:startDestination="@id/home_destination">

    <fragment
        android:id="@+id/home_destination"
        android:name="at.htl.shop.HomeFragment"
        android:label="Home"/>
    <fragment
        android:id="@+id/shop_destination"
        android:name="at.htl.shop.ShopFragment"
        android:label="Shop"/>
    <fragment
        android:id="@+id/about_destination"
        android:name="at.htl.shop.AboutFragment"
        android:label="About"/>
    <fragment
        android:id="@+id/cart_destination"
        android:name="at.htl.shop.CartFragment"
        android:label="Cart"/>
</navigation>
```

```
class CartFragment : Fragment() {
    override fun onCreateView(…): View? {
        return inflater.inflate(R.layout.fragment_cart, container, false)
    }
}
```

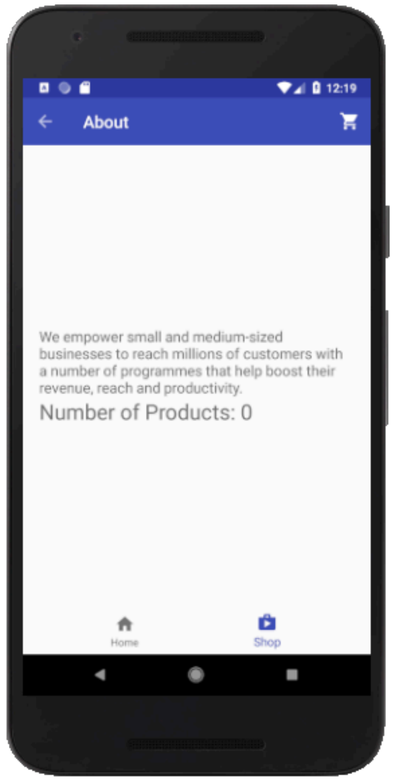
Im CartFragment werden die Fragmente am Screen dargestellt (inflated)

# onClickListener implementieren



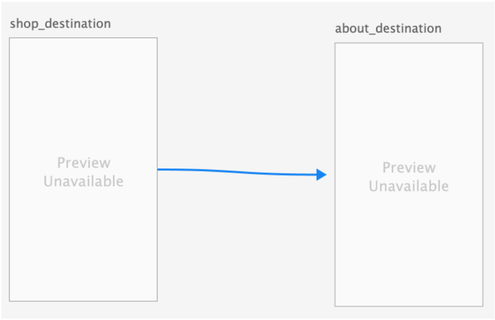

Um zum About Fragment zu gelangen, müssen wir den Button konfigurieren

```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.navigation.Navigation
import kotlinx.android.synthetic.main.fragment_shop.*

class ShopFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_shop, container, false)
    }

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

        btn_about.setOnClickListener {
            Navigation.findNavController(it).navigate(R.id.about_destination)
        }
    }
}
```

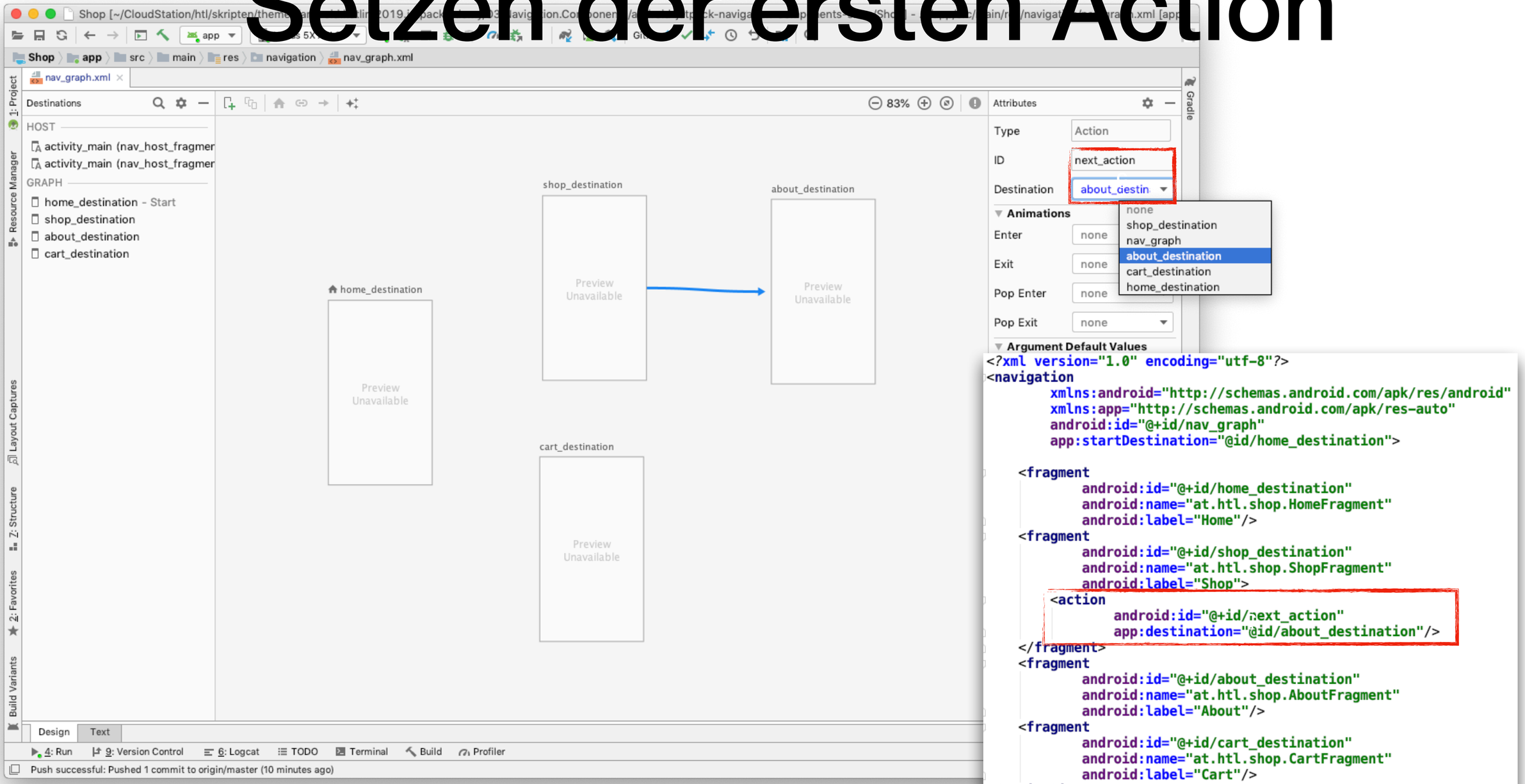



Ergebnis

# Navigation Actions



# Setzen der ersten Action

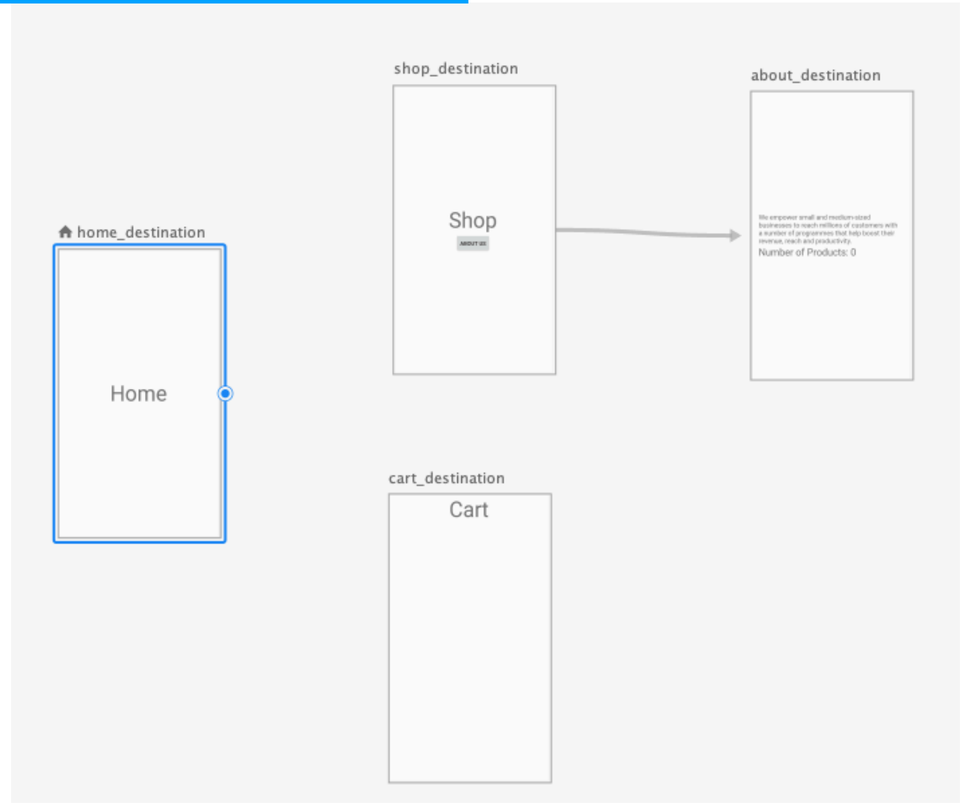



```xml
<?xml version="1.0" encoding="utf-8"?>
<navigation
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/nav_graph"
        app:startDestination="@id/home_destination">

    <fragment
            android:id="@+id/home_destination"
            android:name="at.htl.shop.HomeFragment"
            android:label="Home"/>
    <fragment
            android:id="@+id/shop_destination"
            android:name="at.htl.shop.ShopFragment"
            android:label="Shop">
        <action
                android:id="@+id/next_action"
                app:destination="@id/about_destination"/>
    </fragment>
    <fragment
            android:id="@+id/about_destination"
            android:name="at.htl.shop.AboutFragment"
            android:label="About"/>
    <fragment
            android:id="@+id/cart_destination"
            android:name="at.htl.shop.CartFragment"
            android:label="Cart"/>
</navigation>
```

# nav_graph.xml: Layout eintragen

```xml
<?xml version="1.0" encoding="utf-8"?>
<navigation
        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/nav_graph"
        app:startDestination="@id/home_destination">

    <fragment
            android:id="@+id/home_destination"
            android:name="at.htl.shop.HomeFragment"
            android:label="Home"
            tools:layout="@layout/fragment_home" />
    <fragment
            android:id="@+id/shop_destination"
            android:name="at.htl.shop.ShopFragment"
            android:label="Shop"
            tools:layout="@layout/fragment_shop">
        <action
                android:id="@+id/next_action"
                app:destination="@id/about_destination" />
    </fragment>
    <fragment
            android:id="@+id/about_destination"
            android:name="at.htl.shop.AboutFragment"
            android:label="About"
            tools:layout="@layout/fragment_about" />
    <fragment
            android:id="@+id/cart_destination"
            android:name="at.htl.shop.CartFragment"
            android:label="Cart"
            tools:layout="@layout/fragment_cart" />
</navigation>
```

Nun werden die fragment-Elemente mit ihren Layouts ergänzt. Dadurch sieht man im Navigations-Editor das jeweilige Layout

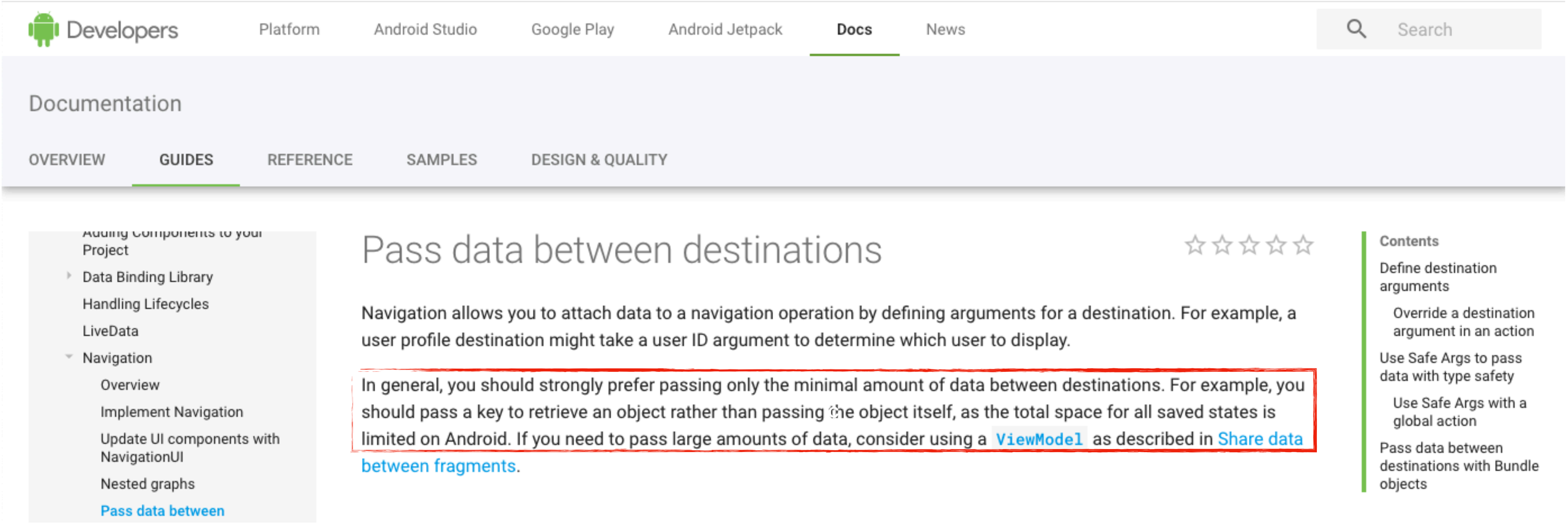


# Verwenden der Action beim onClickListener

```
package at.htl.shop

import ...

class ShopFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_shop, container, false)
    }

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

        btn_about.setOnClickListener {
            // Navigation.findNavController(it).navigate(R.id.about_destination)

            Navigation.findNavController(it).navigate(R.id.next_action)
        }
    }
}
```


funktioniert!

# Pass Data between Destinations Using SafeArgs




https://developer.android.com/topic/libraries/architecture/navigation/navigation-pass-data

# Konfigurieren von SafeArgs

Shop

Configure project in Project Structure dialog. Open Project Str

```
// Top-level build file where you can add configuration options common to all sub-projects/m

buildscript {
    ext.kotlin_version = '1.3.21'
    repositories {
        google()
        jcenter()

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.5.0-alpha04'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath "android.arch.navigation:navigation-safe-args-gradle-plugin:1.0.0-beta02"

        // 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
}
```

app

Configure project in Project Structure dialog. Open Project Structure

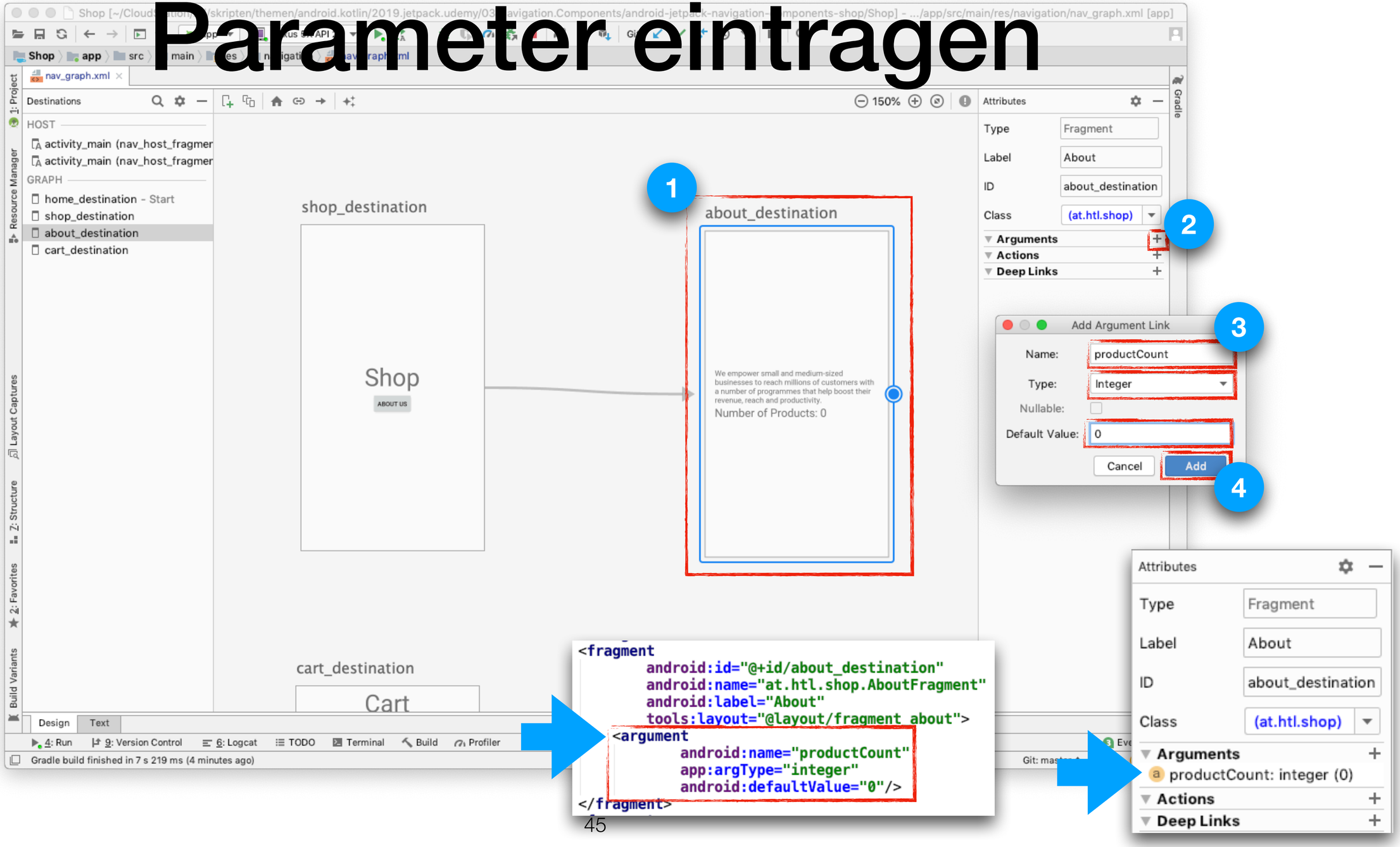
```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'

apply plugin: "androidx.navigation.safeargs.kotlin"

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "at.htl.shop"
        minSdkVersion 23
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.tx
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
```

https://developer.android.com/jetpack/androidx/releases/navigation#declaring_dependencies

# Aufgabenstellung


Eine Zufallszahl wird vom shop- zum about-fragment übergeben

# Parameter eintragen



# ShopFragment.kt

```kotlin
package at.htl.shop

import ...

class ShopFragmentDirections private constructor() {
    private data class NextAction(val productCount: Int = 0) : NavDirections {
        override fun getActionId(): Int = at.htl.shop.R.id.next_action

        override fun getArguments(): Bundle {
            val result = Bundle()
            result.putInt("productCount", this.productCount)
            return result
        }
    }

    companion object {
        fun nextAction(productCount: Int = 0): NavDirections = NextAction(productCount)
    }
}
```

```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.navigation.Navigation
import kotlinx.android.synthetic.main.fragment_shop.*
import kotlin.random.Random

class ShopFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_shop, container, false)
    }

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

        btn_about.setOnClickListener {
            // Navigation.findNavController(it).navigate(R.id.about_destination)

            //Navigation.findNavController(it).navigate(R.id.next_action)

            val nextAction = ShopFragmentDirections.nextAction(Random.nextInt(200))
            Navigation.findNavController(it).navigate(nextAction)
        }
    }
}
```

# AboutFragment.kt

```kotlin
package at.htl.shop

import ...

data class AboutFragmentArgs(val productCount: Int = 0) : NavArgs {
    fun toBundle(): Bundle {
        val result = Bundle()
        result.putInt("productCount", this.productCount)
        return result
    }

    companion object {
        @JvmStatic
        fun fromBundle(bundle: Bundle): AboutFragmentArgs {
            bundle.setClassLoader(AboutFragmentArgs::class.java.classLoader)
            val __productCount : Int
            if (bundle.containsKey("productCount")) {
                __productCount = bundle.getInt("productCount")
            } else {
                __productCount = 0
            }
            return AboutFragmentArgs(__productCount)
        }
    }
}
```

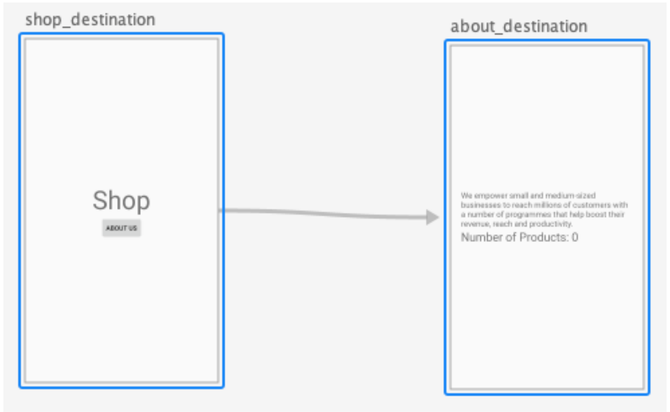
```kotlin
package at.htl.shop

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.main.fragment_about.*

class AboutFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_about, container, false)
    }

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

        arguments?.let {
            val safeArgs = AboutFragmentArgs.fromBundle(it)
            tv_product_count.text = "Total Products Available: ${safeArgs.productCount}"
        }
    }
}
```

# SafeArgs

SENDER

EMPFÄNGER

```
class ShopFragment : Fragment() {

    override fun onCreateView(…): View? { … }

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

        btn_about.setOnClickListener {
            val nextAction = ShopFragmentDirections
                                .nextAction(Random.nextInt(200))
            Navigation.findNavController(it).navigate(nextAction)
        }
    }
}
```




```
class AboutFragment : Fragment() {

    override fun onCreateView(…): View? { … }

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

        arguments?.let {
            val safeArgs = AboutFragmentArgs.fromBundle(it)
            tv_product_count.text =
                „Total Products Available: ${safeArgs.productCount}"
        }
    }
}
```

ShopFragment  next_action  AboutFragment

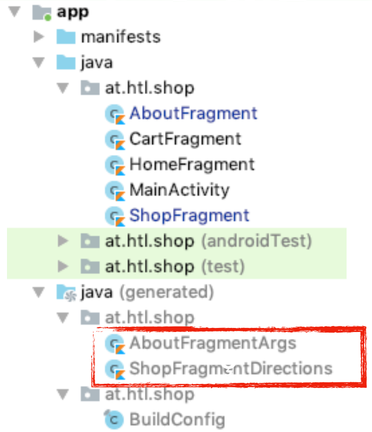
automatisch generierte Klassen

ShopFragmentDirection

NextAction

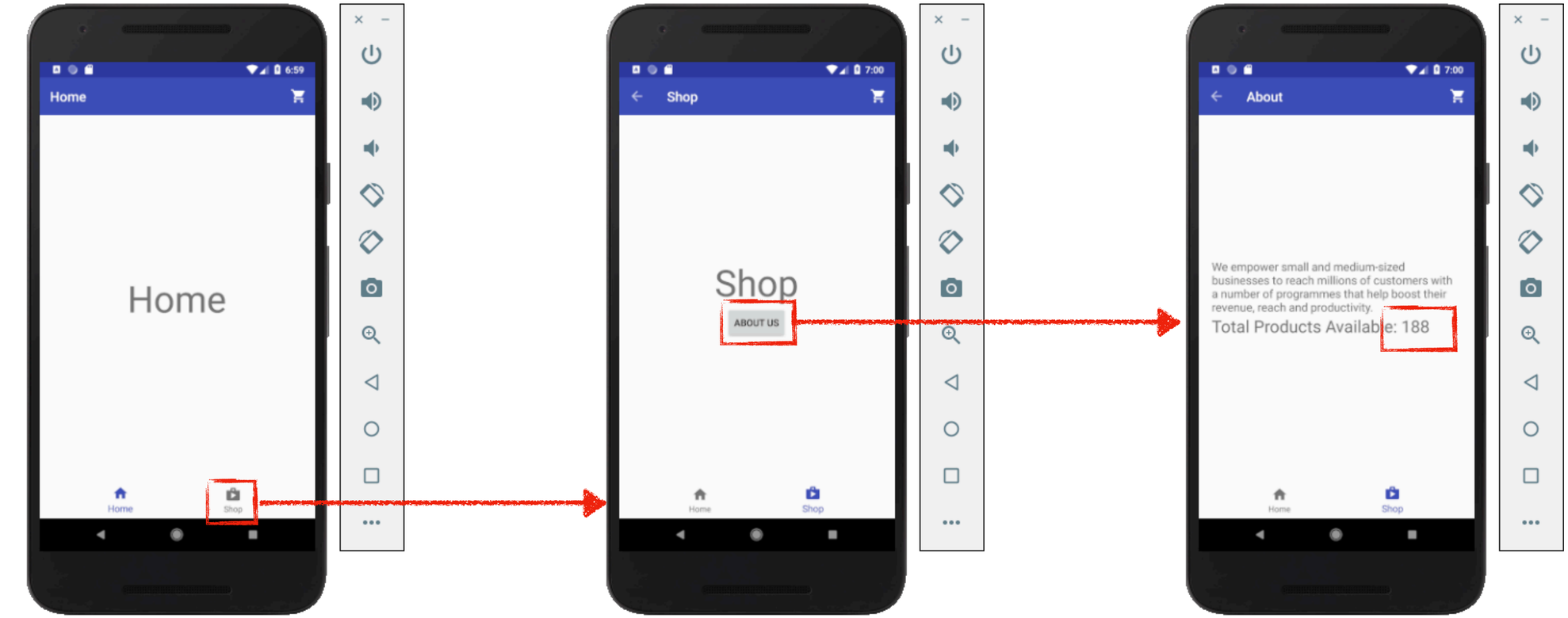
AboutFragmentArgs

```
class ShopFragmentDirections private constructor() {
    private data class NextAction(val productCount: Int = 0) : NavDirections {
        override fun getActionId(): Int = at.htl.shop.R.id.next_action

        override fun getArguments(): Bundle {
            val result = Bundle()
            result.putInt("productCount", this.productCount)
            return result
        }
    }

    companion object {
        fun nextAction(productCount: Int = 0): NavDirections = NextAction(productCount)
    }
}
```




```
data class AboutFragmentArgs(val productCount: Int = 0) : NavArgs {
    fun toBundle(): Bundle {
        val result = Bundle()
        result.putInt("productCount", this.productCount)
        return result
    }

    companion object {
        @JvmStatic
        fun fromBundle(bundle: Bundle): AboutFragmentArgs {
            bundle.setClassLoader(AboutFragmentArgs::class.java.classLoader)
            val __productCount : Int
            if (bundle.containsKey("productCount")) {
                __productCount = bundle.getInt("productCount")
            } else {
                __productCount = 0
            }
            return AboutFragmentArgs(__productCount)
        }
    }
}
```

# Testlauf


Home
Home
Shop
Shop
ABOUT US
About
We empower small and medium-sized businesses to reach millions of customers with a number of programmes that help boost their revenue, reach and productivity.
Total Products Available: 188

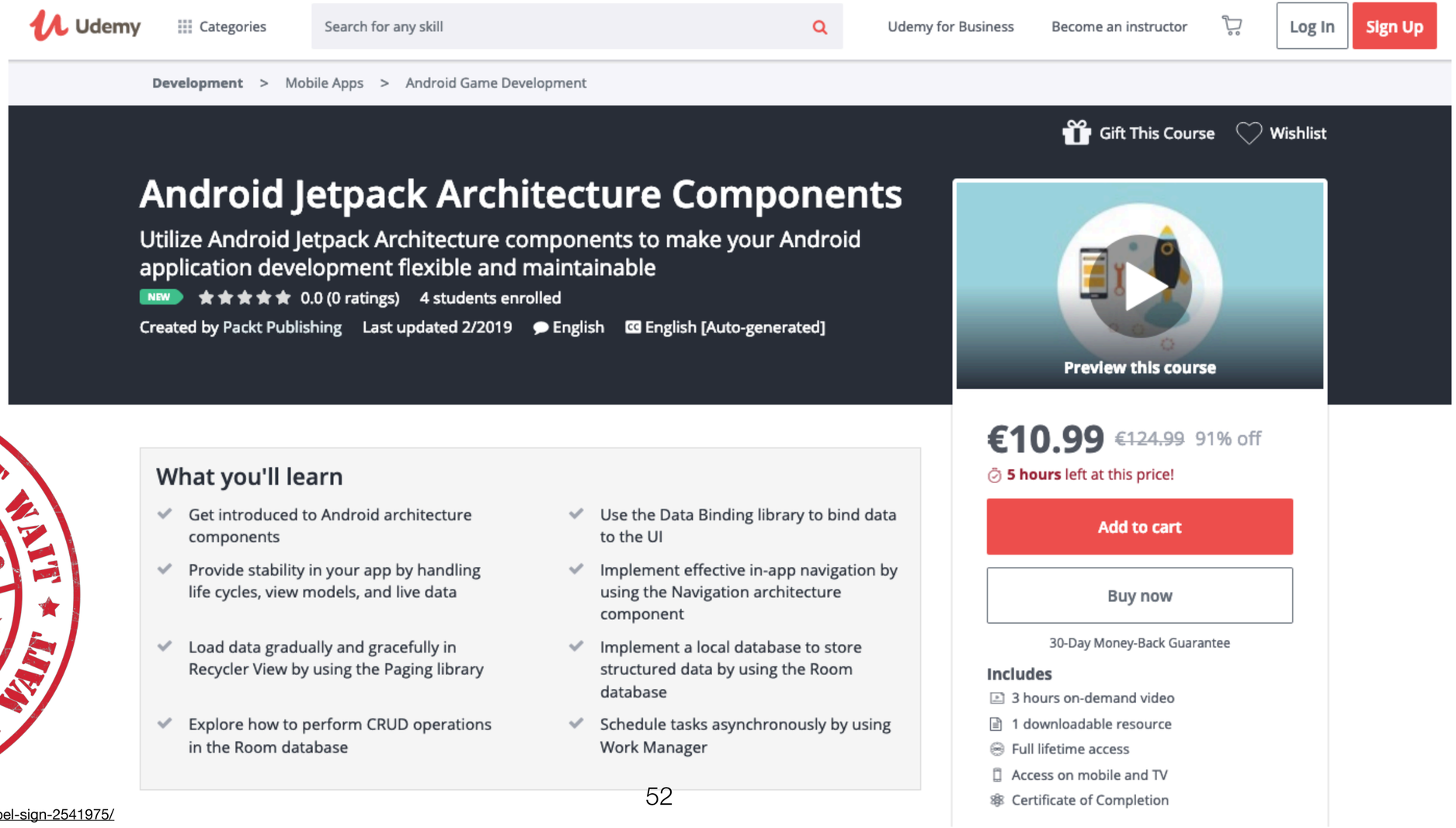
# Wie gehts weiter?

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


# Noch Fragen?

# Source

https://www.udemy.com/android-jetpack-architecture-components/





https://pixabay.com/en/buy-now-seal-buy-now-label-sign-2541975/