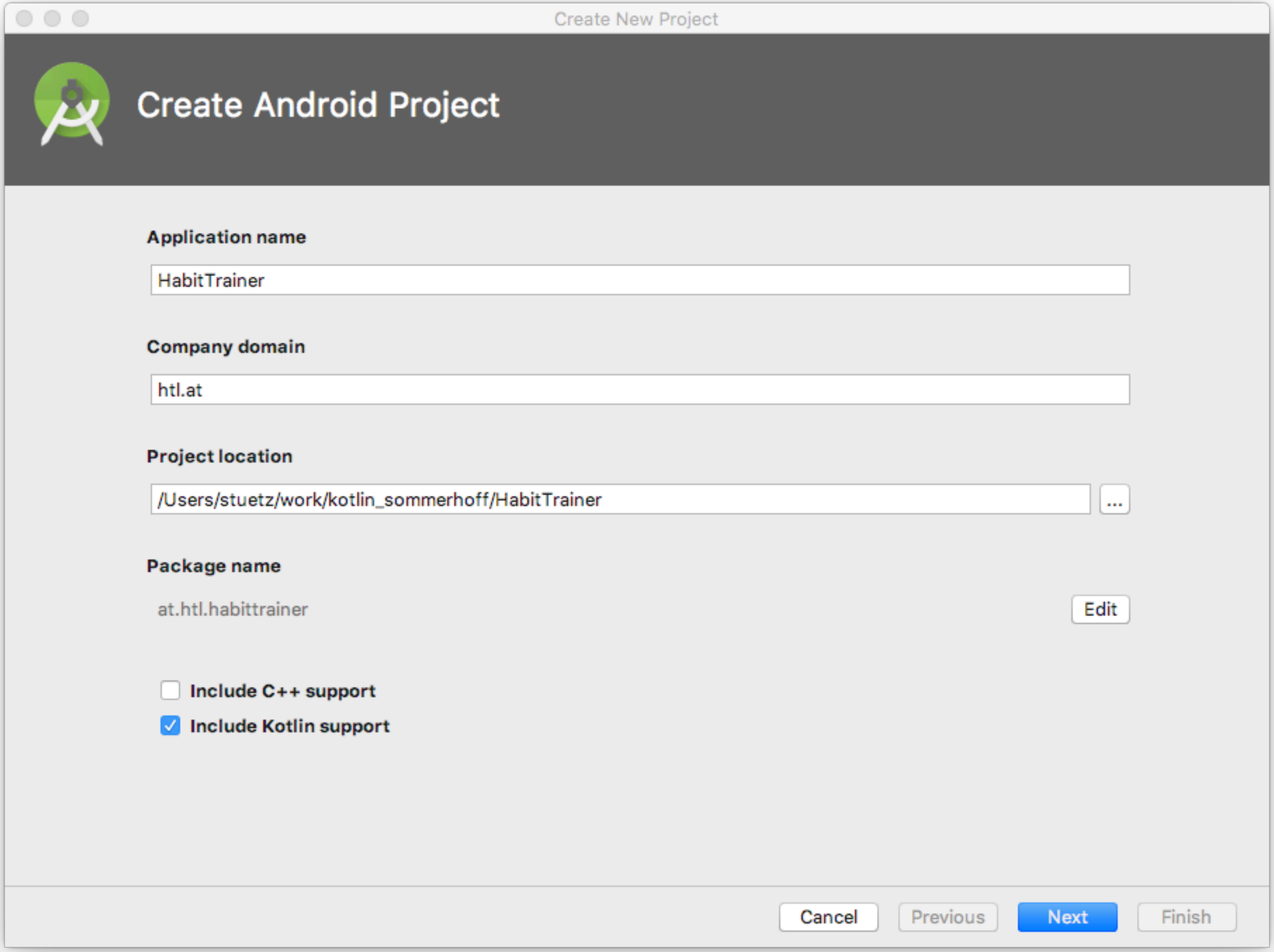
Kotlin

Android

https://kotlinlang.org/



Create Android Project

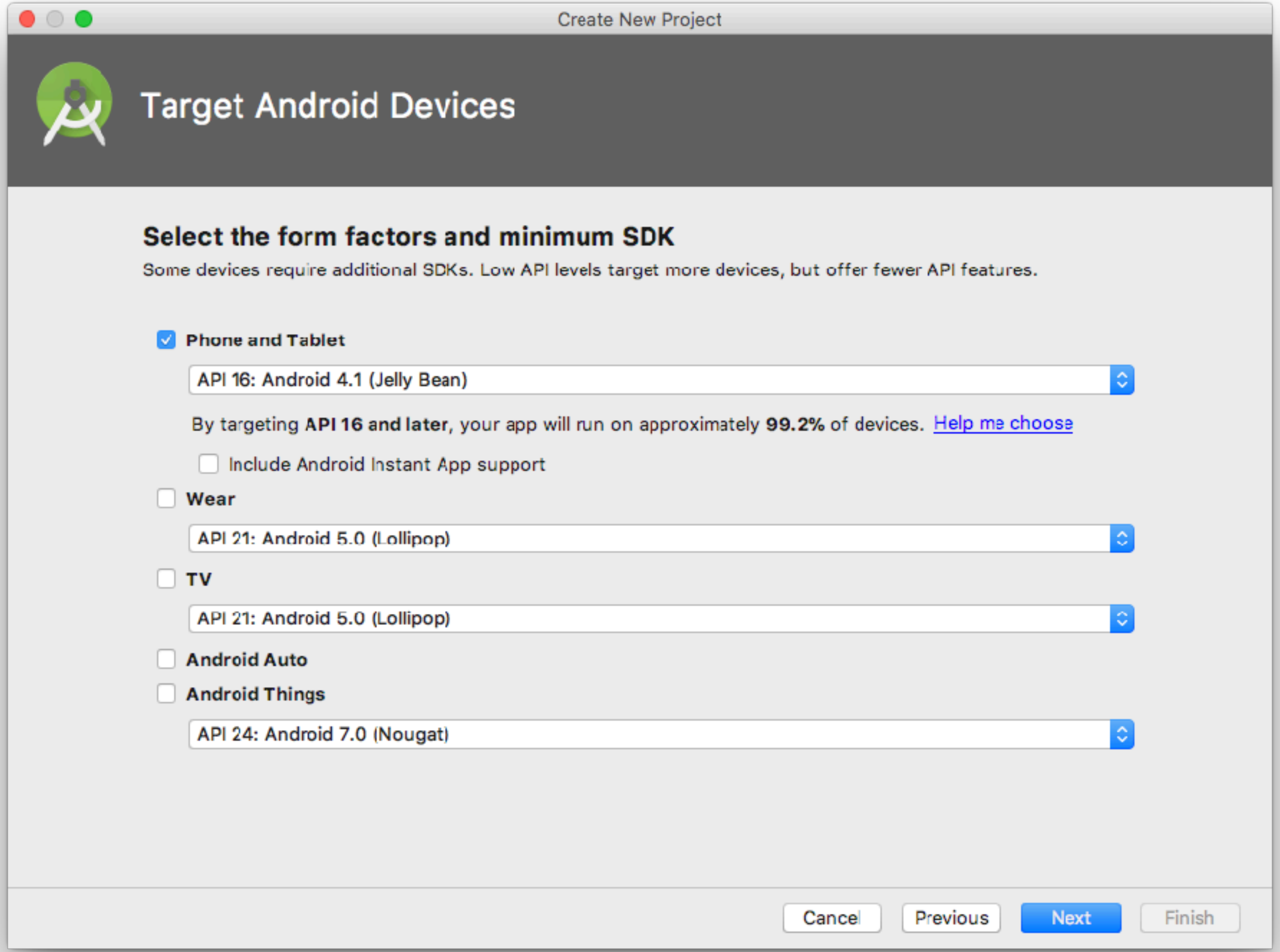
Application name: HabitTrainer

Company domain: htl.at

Project location: /Users/stuetz/work/kotlin_sommerhoff/HabitTrainer

Package name: at.htl.habittrainer

- [ ] Include C++ support
- [x] Include Kotlin support



Create New Project

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 16: Android 4.1 (Jelly Bean)

By targeting **API 16 and later**, your app will run on approximately **99.2%** of devices. Help me choose

☐ Include Android Instant App support

☐ **Wear**

API 21: Android 5.0 (Lollipop)

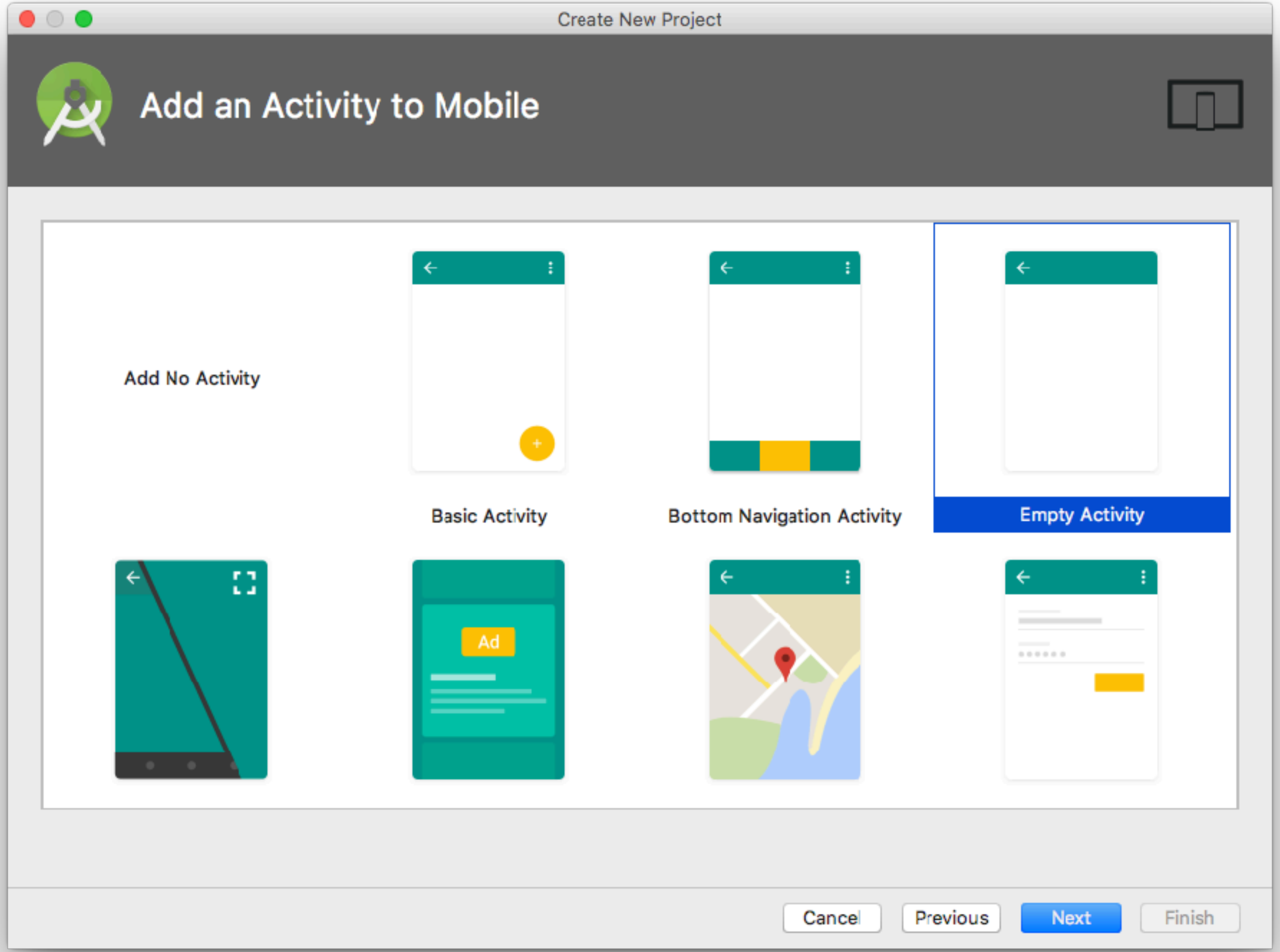
☐ **TV**

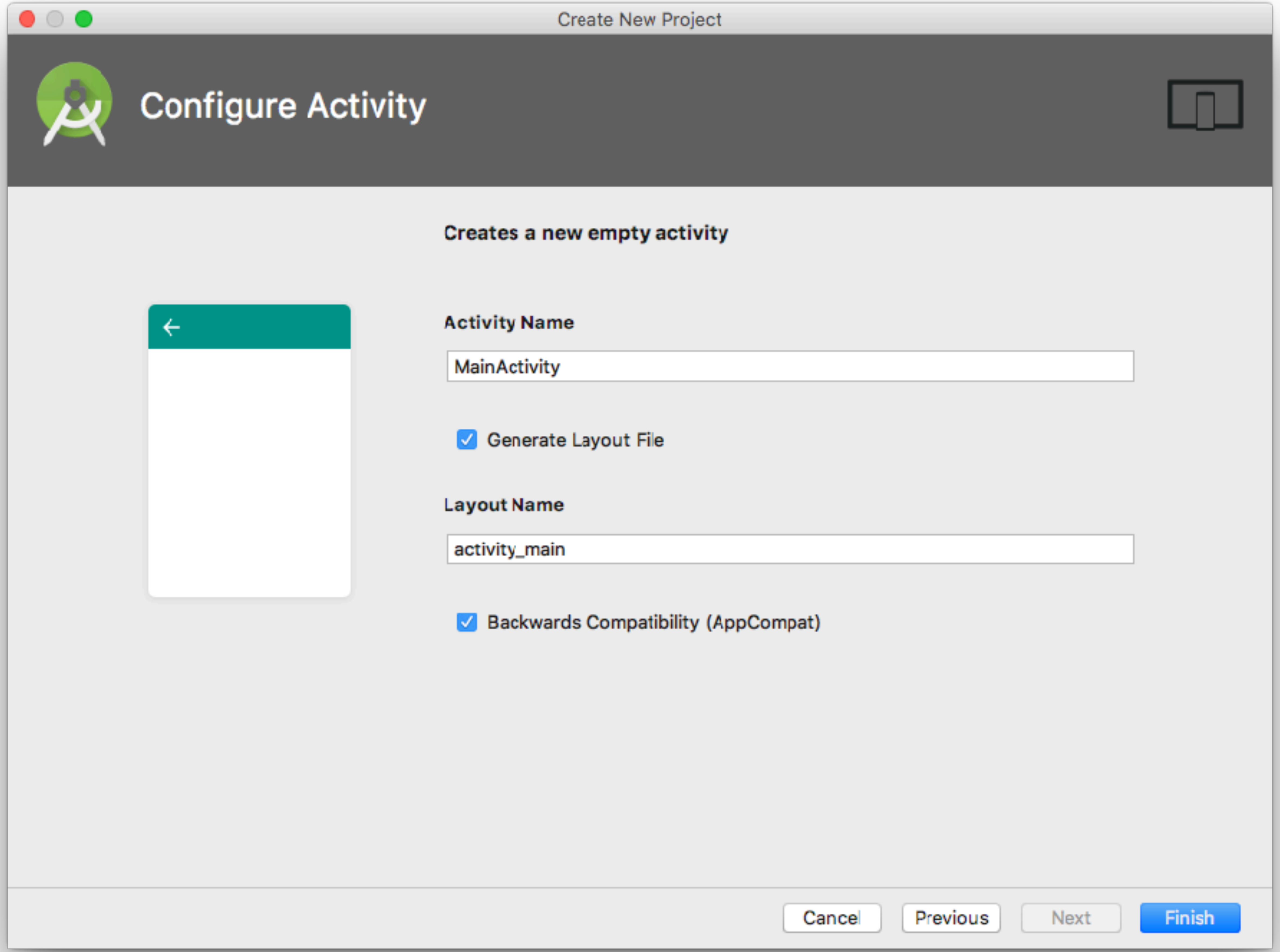
API 21: Android 5.0 (Lollipop)

☐ **Android Auto**

☐ **Android Things**

API 24: Android 7.0 (Nougat)


Create New Project
Add an Activity to Mobile
Add No Activity
Basic Activity
Bottom Navigation Activity
Empty Activity
Ad
Cancel
Previous
Next
Finish



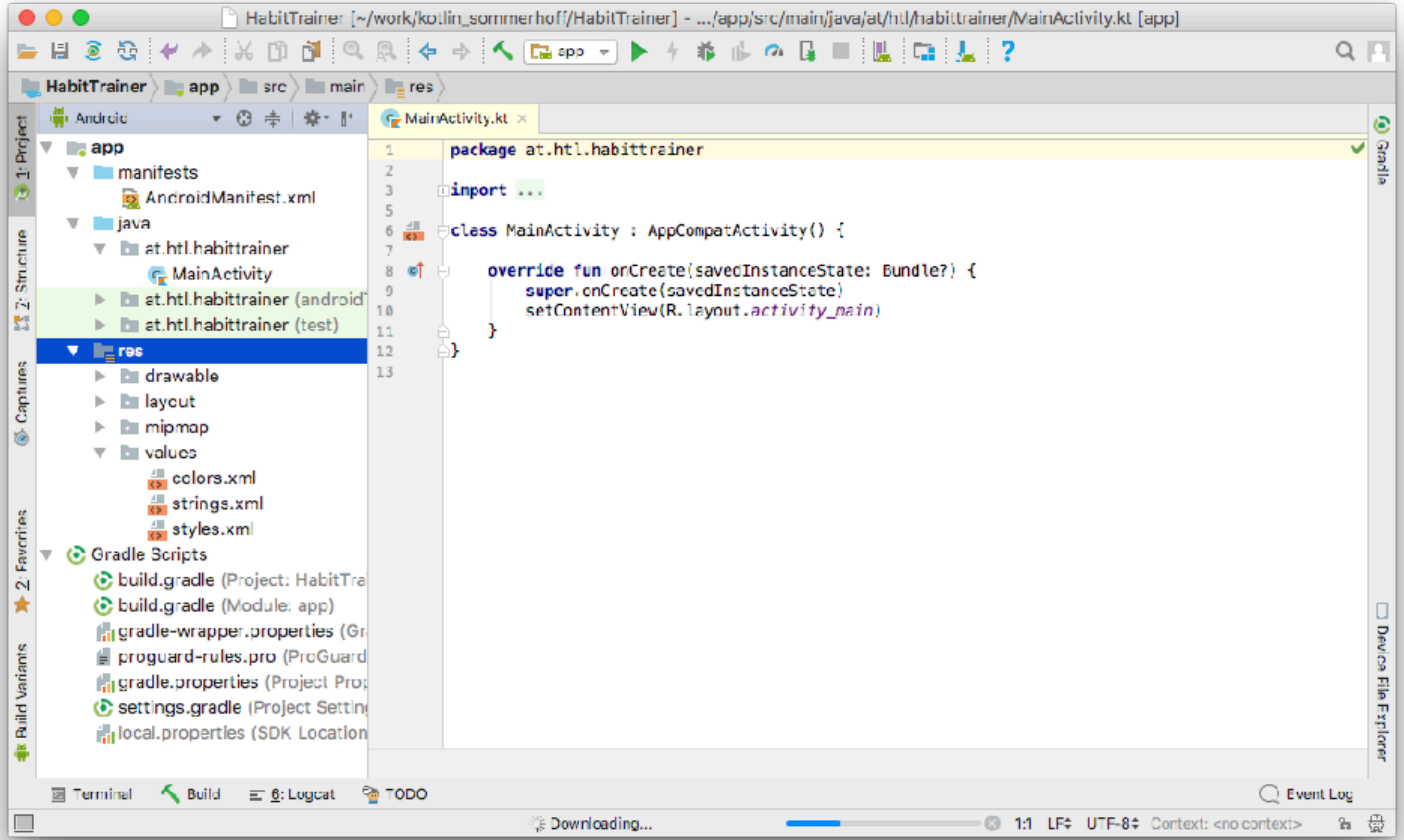
Creates a new empty activity

Activity Name: MainActivity

[✓] Generate Layout File

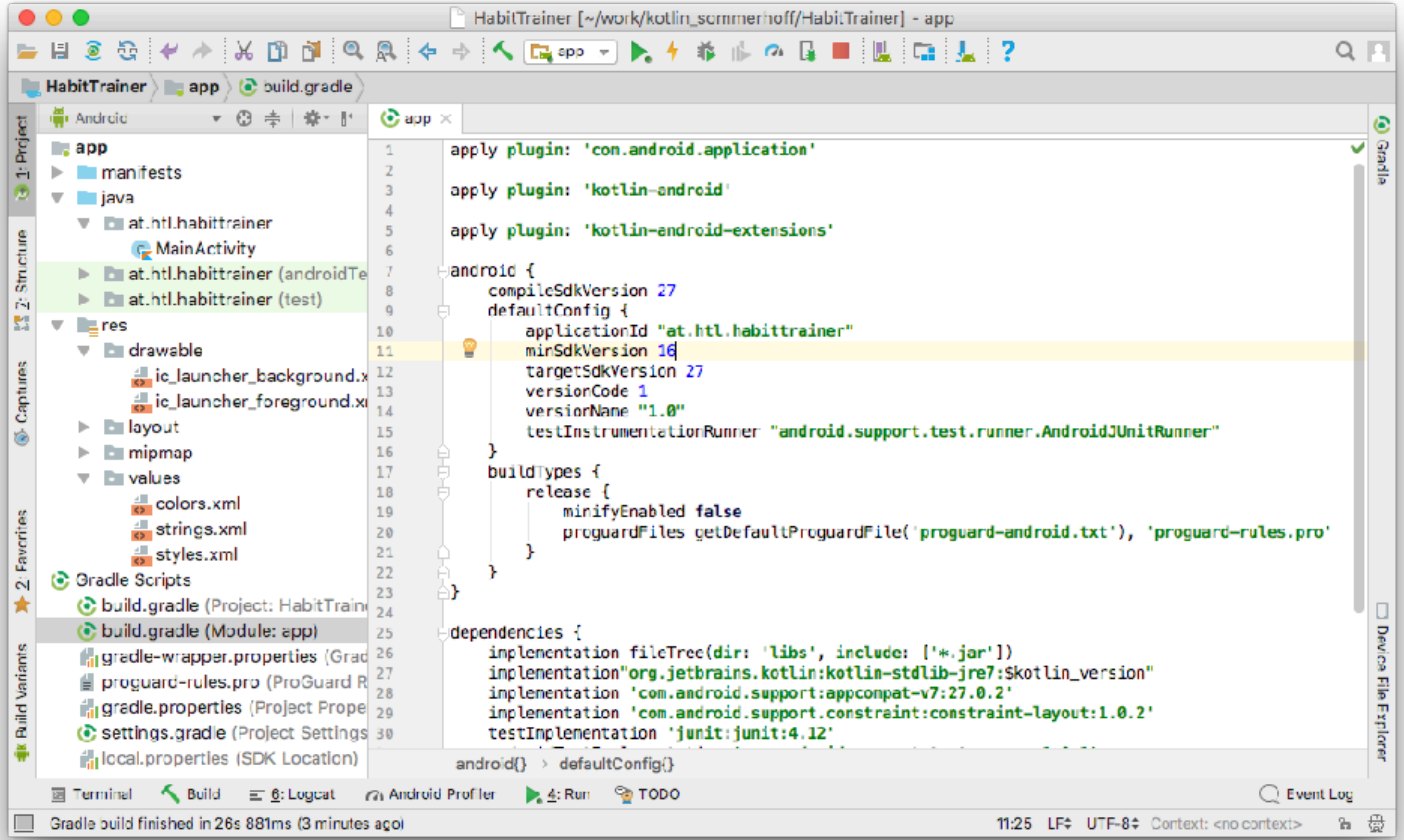
Layout Name: activity_main

[✓] Backwards Compatibility (AppCompat)



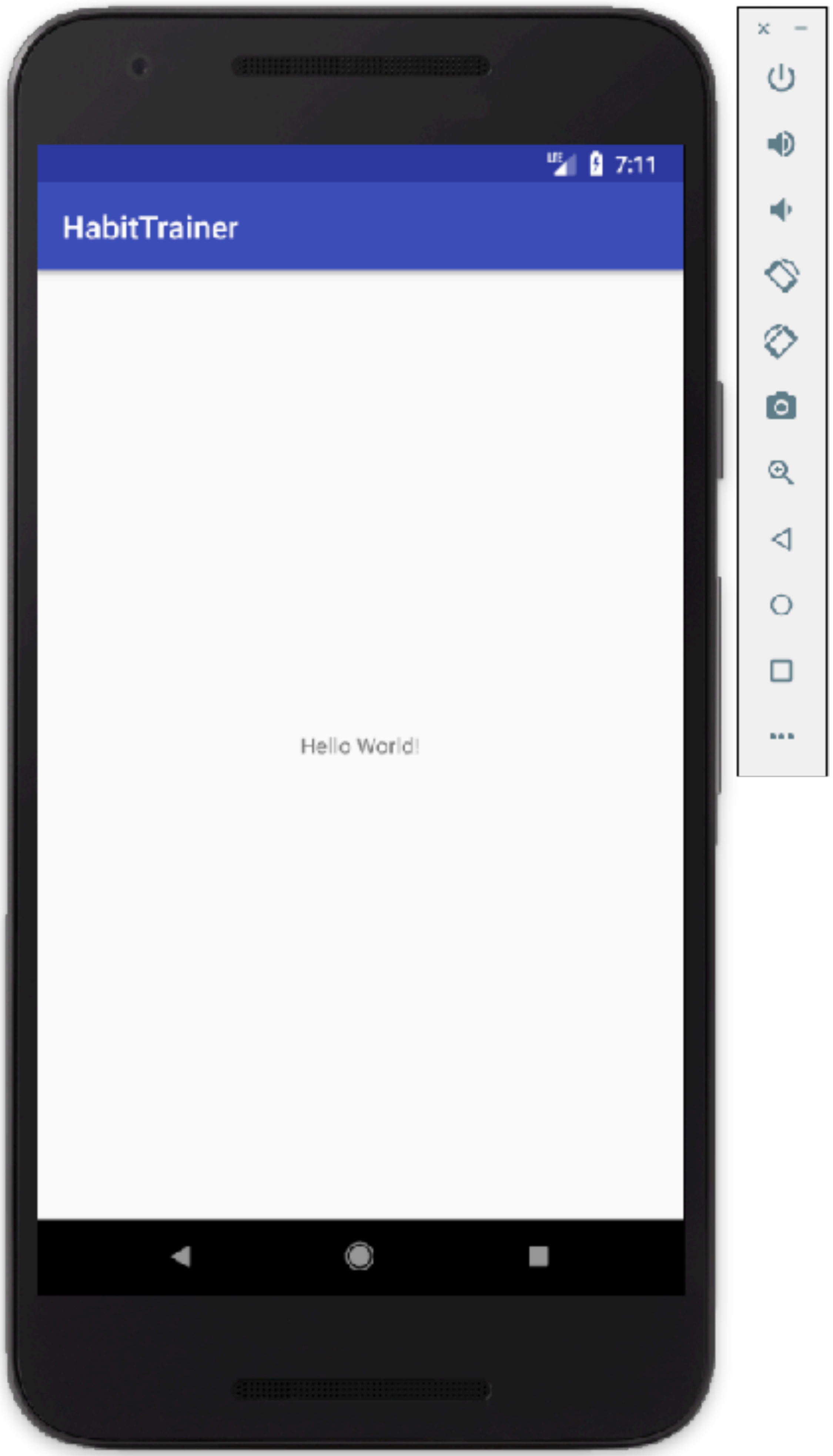
HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - .../app/src/main/java/at/htl/habittrainer/MainActivity.kt [app]
HabitTrainer
app
src
main
res
Android
MainActivity.kt
app
manifests
AndroidManifest.xml
java
at.htl.habittrainer
MainActivity
at.htl.habittrainer (androidTest)
at.htl.habittrainer (test)
res
drawable
layout
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: HabitTrainer)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
2: Structure
Captures
2: Favorites
Build Variants
Gradle
Device File Explorer
Terminal
Build
6: Logcat
TODO
Event Log
Downloading...
1:1 LF UTF-8 Context: <no context>

```kotlin
package at.htl.habittrainer

import ...

class MainActivity : AppCompatActivity() {

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


HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - app
HabitTrainer
app
build.gradle
Android
app
manifests
java
at.htl.habittrainer
MainActivity
at.htl.habittrainer (androidTe
at.htl.habittrainer (test)
res
drawable
ic_launcher_background.x
ic_launcher_foreground.xi
layout
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: HabitTrain
build.gradle (Module: app)
gradle-wrapper.properties (Grad
proguard-rules.pro (ProGuard R
gradle.properties (Project Prope
settings.gradle (Project Settings
local.properties (SDK Location)
1: Project
2: Structure
Captures
2: Favorites
Build Variants
Gradle
Device File Explorer
android{} › defaultConfig{}
Terminal
Build
6: Logcat
Android Profiler
4: Run
TODO
Event Log
Gradle build finished in 26s 881ms (3 minutes ago)
11:25
LF
UTF-8
Context: <no context>

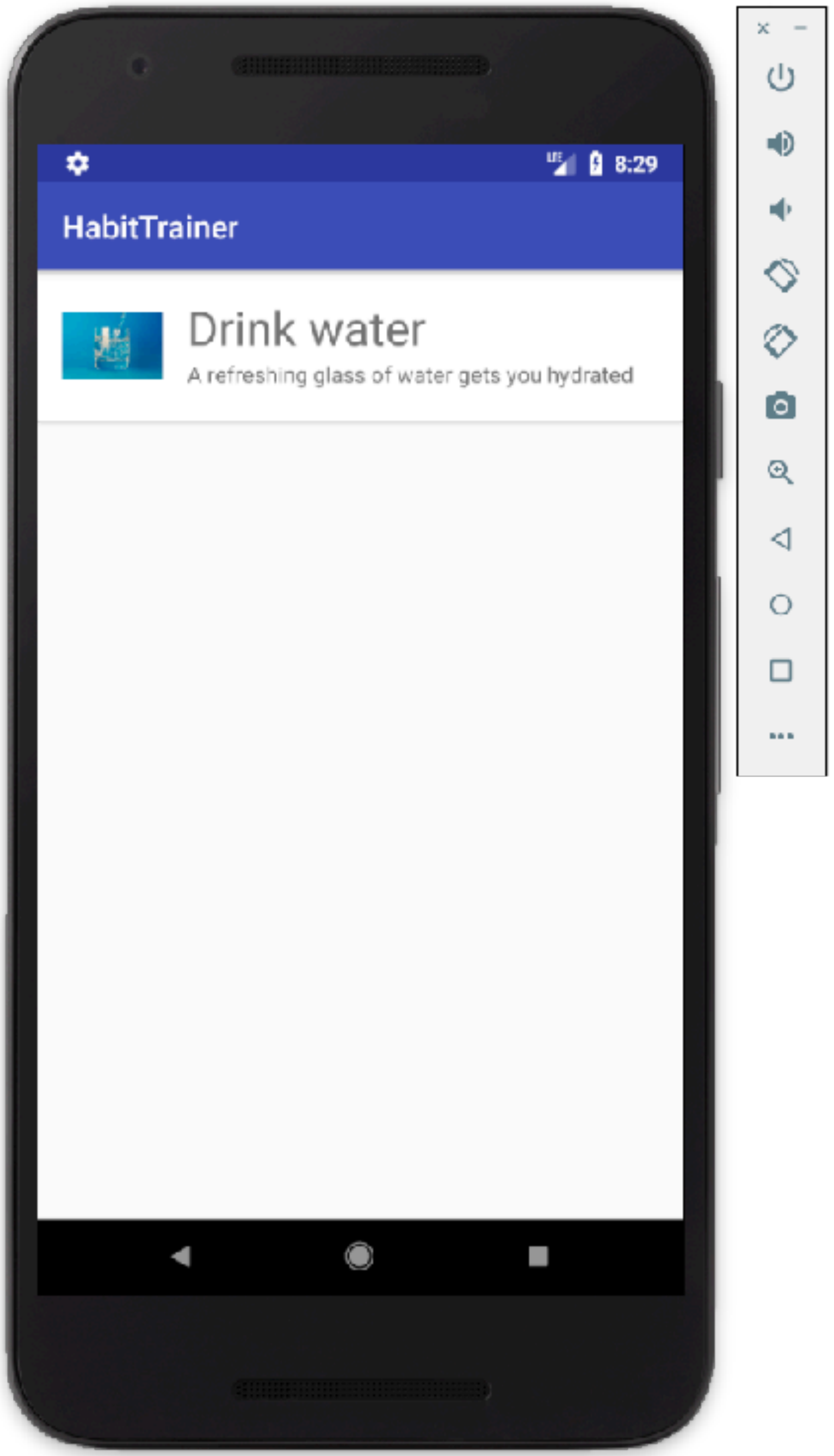
```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'

android {
    compileSdkVersion 27
    defaultConfig {
        applicationId "at.htl.habittrainer"
        minSdkVersion 16
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:27.0.2'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
```

ev.


7:11
HabitTrainer
Hello World!

activity_main.xml

```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=".MainActivity">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <RelativeLayout
            android:padding="16dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">


            <ImageView
                android:id="@+id/iv_icon"
                android:src="@drawable/water"
                android:layout_alignParentLeft="true"
                android:layout_alignParentTop="true"
                android:layout_marginRight="16dp"
                android:layout_width="64dp"
                android:layout_height="64dp" />

            <TextView
                android:id="@+id/tv_title"
                android:layout_toRightOf="@+id/iv_icon"
                android:layout_alignParentTop="true"
                android:text="Drink water"
                android:textSize="30sp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/tv_description"
                android:text="A refreshing glass of water gets you hydrated"
                android:layout_toRightOf="@+id/iv_icon"
                android:layout_below="@+id/tv_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

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



Drink water

A refreshing glass of water gets you hydrated

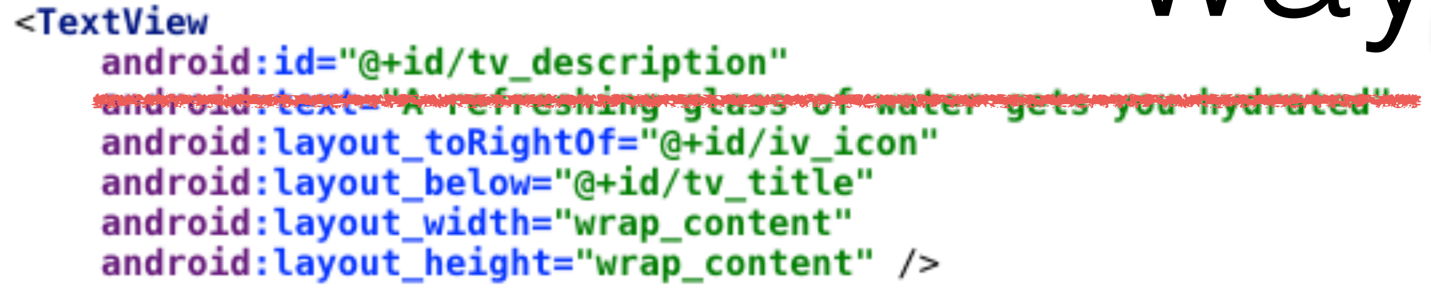
Zugriff auf Views - the Java way

```
package at.htl.habittrainer

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView

class MainActivity : AppCompatActivity() {

1   private lateinit var tvDescription: TextView

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

2       tvDescription = findViewById(R.id.tv_description)
3       tvDescription?.text = "A refreshing glass of water gets you hydrated"

}
```

1. Zuerst muß man eine Java-Variable für jede View anlegen.
2. Man muß der Java-Variable das View-Objekt zuweisen
3. Nun kann man dem View-Objekt Werte zuweisen

Hier wird sichergestellt, dass das Objekt != null ist

Zugriff auf Views - the Kotlin way



```
<TextView
    android:id="@+id/tv_description"
    ~~android:text="A refreshing glass of water gets you hydrated"~~
    android:layout_toRightOf="@+id/iv_icon"
    android:layout_below="@+id/tv_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

1 Zeile löschen

Die „kotlin-android-extensions“ scannen automatisch die Layout-Files und erstellen Variablen für jede View mit deren id

```
package at.htl.habittrainer

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
? a                   ttrainer.R.id.tv_description? (multiple choices...) ⌥⏎
        tv_description.text = "A refreshing glass of water gets you hydrated"

    }
}
```

3

2

Challenge: Set Texts and Image Source Programmatically

Challenge: Use the Kotlin Android Extensions

In this challenge, you'll familiarize yourself with the Kotlin Android Extensions (and with layouts).

1. Set all fixed texts and image sources (drawables) from the layout programmatically in your MainActivity. Use the Kotlin Android Extensions to access all layout elements.
2. Fix all warnings in the layout file.

Hints

For #1: Make sure you have a unique ID set for each element.

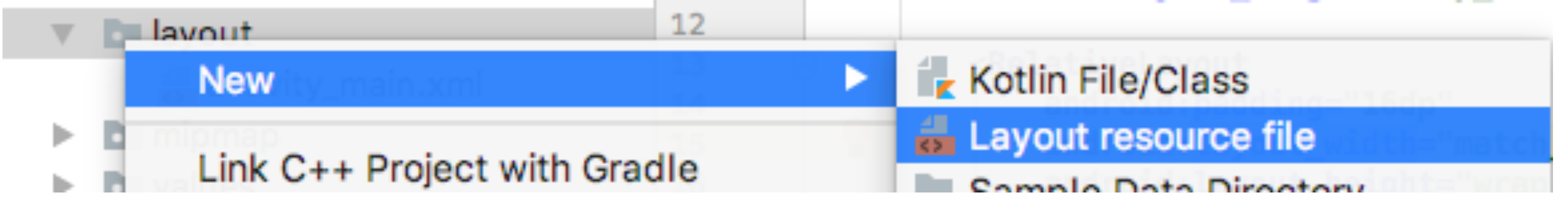
For #2: Use Android Studio's suggestions by moving the cursor into the yellow highlighting and pressing Alt+Enter.

MainActivity.kt

```
package at.htl.habittrainer

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    private lateinit var tvDescription: TextView

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

        iv_icon.setImageResource(R.drawable.water)
        tv_title.text = getString(R.string.drink_water)
        tv_description.text = getString(R.string.drink_water_desc)

    }
}
```

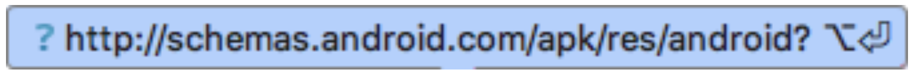
activity_main.xml

```xml
<RelativeLayout
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">


    <ImageView
        android:id="@+id/iv_icon"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="16dp"
        android:layout_width="64dp"
        android:layout_height="64dp"
        android:layout_marginEnd="16dp"
        android:contentDescription="@string/habit_icon" />

    <TextView
        android:id="@+id/tv_title"
        android:layout_toRightOf="@+id/iv_icon"
        android:layout_alignParentTop="true"
        android:textSize="30sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toEndOf="@+id/iv_icon" />

    <TextView
        android:id="@+id/tv_description"
        android:layout_toRightOf="@+id/iv_icon"
        android:layout_toEndOf="@+id/iv_icon"
        android:layout_below="@+id/tv_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



single_card.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="16dp">


        <ImageView
            android:id="@+id/iv_icon"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_marginEnd="16dp"
            android:layout_marginRight="16dp"
            android:contentDescription="@string/habit_icon" />

        <TextView
            android:id="@+id/tv_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_toEndOf="@+id/iv_icon"
            android:layout_toRightOf="@+id/iv_icon"
            android:textSize="30sp" />

        <TextView
            android:id="@+id/tv_description"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tv_title"
            android:layout_toEndOf="@+id/iv_icon"
            android:layout_toRightOf="@+id/iv_icon" />

    </RelativeLayout>

</android.support.v7.widget.CardView>
```

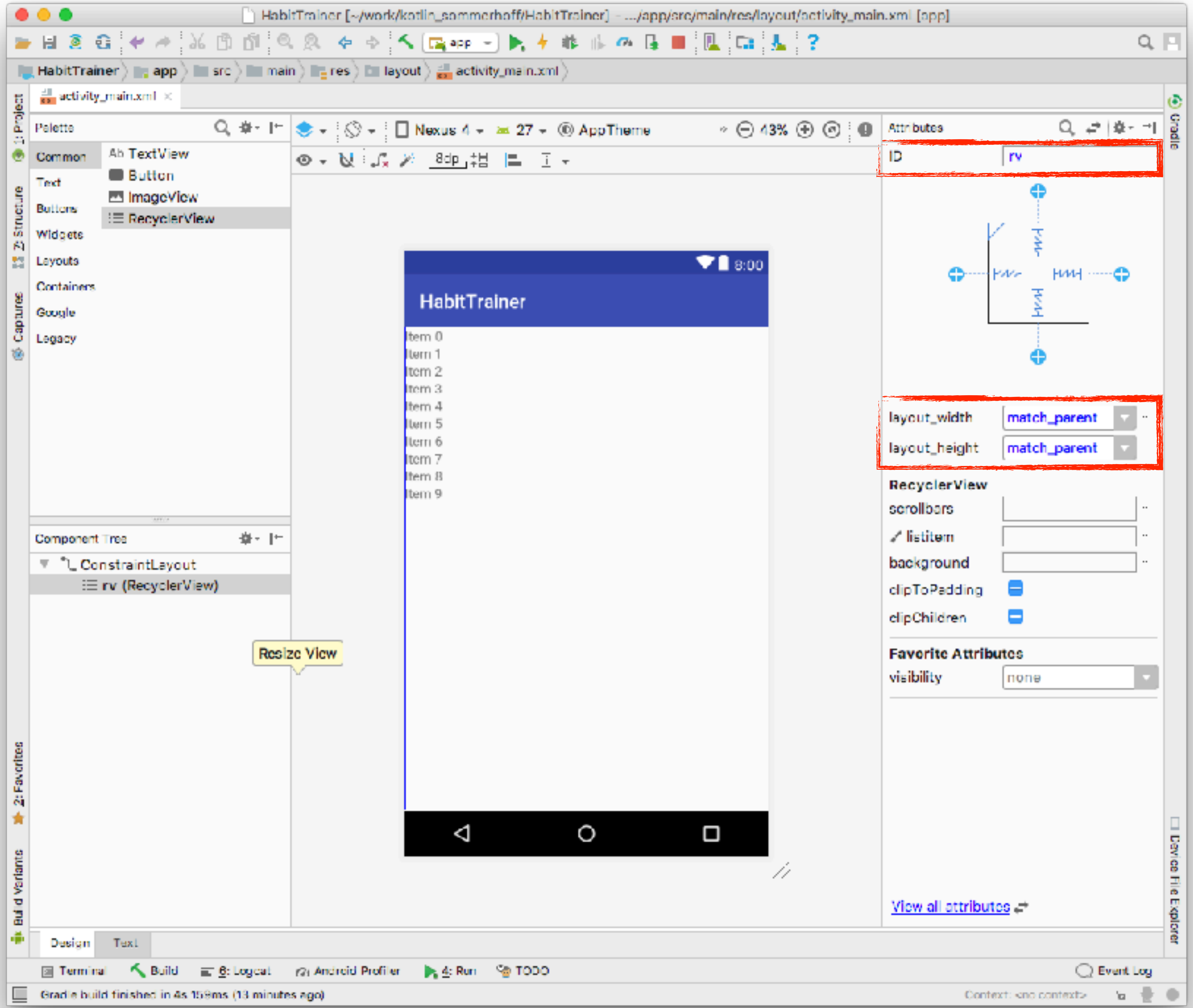


1. Neues Layout „single_card.xml“ anlegen
2. Das CardView-Element aus activity_main.xml verschieben
3. Den Namespace generieren lassen
4. Ebenso fehlende layout-einträge

build.gradle (Module: app)

Dependency für RecyclerView eintragen

```
dependencies {
    compile 'com.android.support:cardview-v7:27.0.2'
    compile 'com.android.support:recyclerview-v7:27.0.2'
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:27.0.2'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```


HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - .../app/src/main/res/layout/activity_main.xml [app]

ID: rv

layout_width: match_parent

layout_height: match_parent

RecyclerView

scrollbars

listitem

background

clipToPadding

clipChildren

Favorite Attributes

visibility: none

Component Tree

ConstraintLayout

rv (RecyclerView)

Gradle build finished in 4s 159ms (13 minutes ago)

activity_main.xml

```xml
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rv"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

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

MainActivity.kt

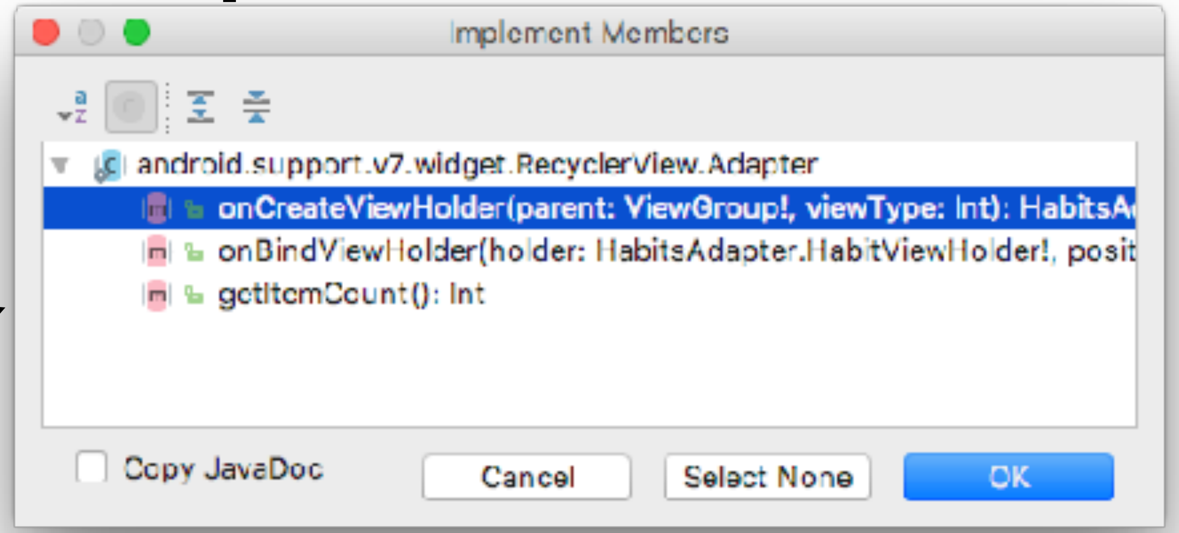
```kotlin
class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }
}
```

Data Class Habit.kt

```kotlin
package at.htl.habittrainer

data class Habit(val title: String, val description: String, val image: Int)

fun getSampleHabits(): List<Habit> {
    return listOf(
            Habit("Go for walk",
                    "A nice walk in the sun gets you ready for the day ahead",
                    R.drawable.walk),

            Habit("Drink a glass of water",
                    "A refreshing glass of water gets you hydrated",
                    R.drawable.water)
    )
}
```

HabitsAdapter.kt

Ctrl-I


Implement Members

android.support.v7.widget.RecyclerView.Adapter
onCreateViewHolder(parent: ViewGroup!, viewType: Int): HabitsA
onBindViewHolder(holder: HabitsAdapter.HabitViewHolder!, posit
getItemCount(): Int

```kotlin
package at.htl.habittrainer

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

class HabitsAdapter(val habits: List<Habit>) : RecyclerView.Adapter<HabitsAdapter.HabitViewHolder>() {

    class HabitViewHolder(val iv: View) : RecyclerView.ViewHolder(iv)

    override fun onBindViewHolder(holder: HabitViewHolder?, position: Int) {
        TODO("not implemented")
    }

    override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): HabitViewHolder {
        TODO("not implemented")     }

    override fun getItemCount(): Int {
        return habits.size
    }
}
```

kann verkürzt werden auf

```kotlin
override fun getItemCount() = habits.size
```

Create a new ViewHolder

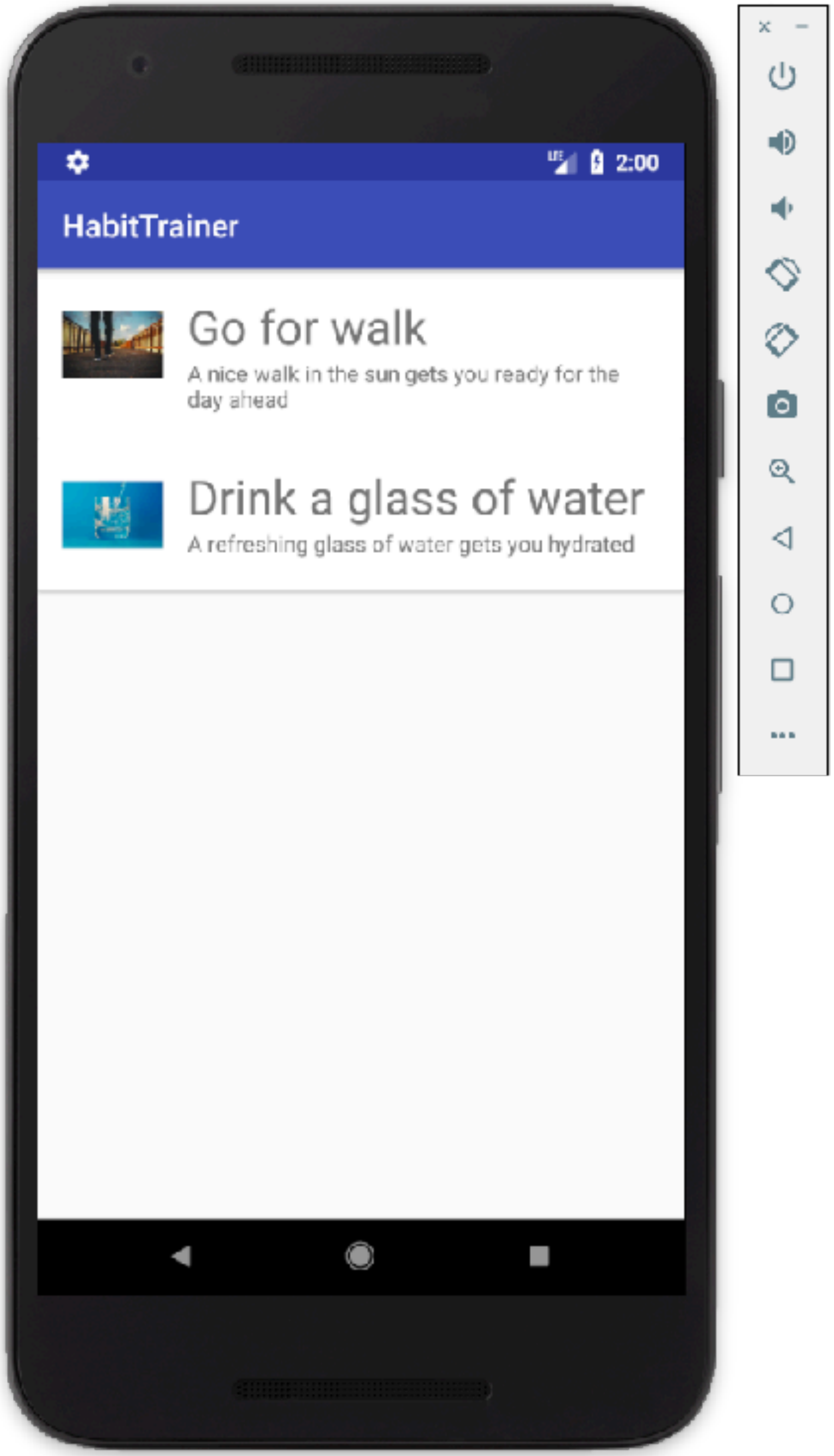
```
package at.htl.habittrainer

import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

class HabitsAdapter(val habits: List<Habit>) : RecyclerView.Adapter<HabitsAdapter.HabitViewHolder>() {

    class HabitViewHolder(val iv: View) : RecyclerView.ViewHolder(iv)

    override fun onBindViewHolder(holder: HabitViewHolder, position: Int) {

    }

    // Create a new ViewHolder
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): HabitViewHolder {
        val view = LayoutInflater
                .from(parent.context)
                .inflate(R.layout.single_card, parent, false)
        return HabitViewHolder(view)
    }

    override fun getItemCount() = habits.size
}
```

parent kann hier nicht null werden

sonst „java.lang.IllegalStateException: The specified child already has a parent. You must call removeView() on the child's parent first“

Specify contents

```
package at.htl.habittrainer

import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import kotlinx.android.synthetic.main.single_card.view.*

class HabitsAdapter(val habits: List<Habit>) : RecyclerView.Adapter<HabitsAdapter.HabitViewHolder>() {

    class HabitViewHolder(val card: View) : RecyclerView.ViewHolder(card)

    // Specifies the contents for the shown habit
    override fun onBindViewHolder(holder: HabitViewHolder?, index: Int) {
        if (holder != null) {  // if wegen SmartCast
            val habit = habits[index]
            holder.card.tv_title.text = habit.title
            holder.card.tv_description.text = habit.description
            holder.card.iv_icon.setImageResource(habit.image)
        }
    }

    // Create a new ViewHolder
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): HabitViewHolder {
        val view = LayoutInflater
                .from(parent.context)
                .inflate(R.layout.single_card, parent, false)
        return HabitViewHolder(view)
    }

    override fun getItemCount() = habits.size
}
```


2:00

HabitTrainer

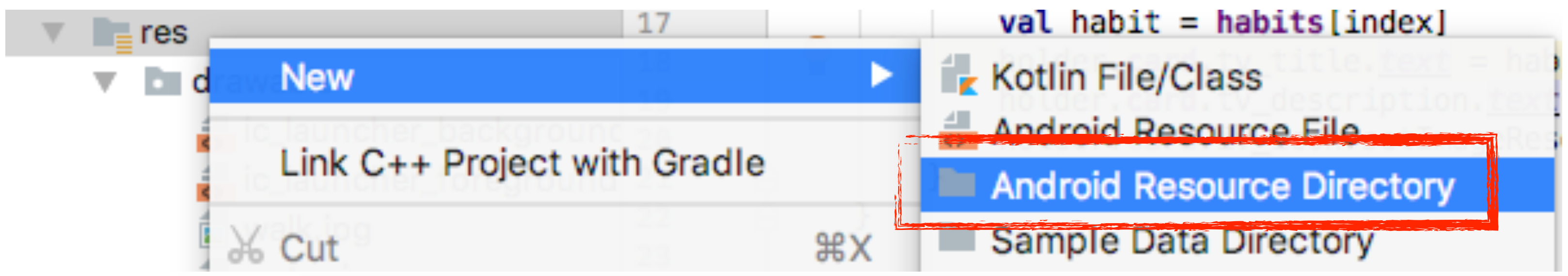
Go for walk

A nice walk in the sun gets you ready for the day ahead

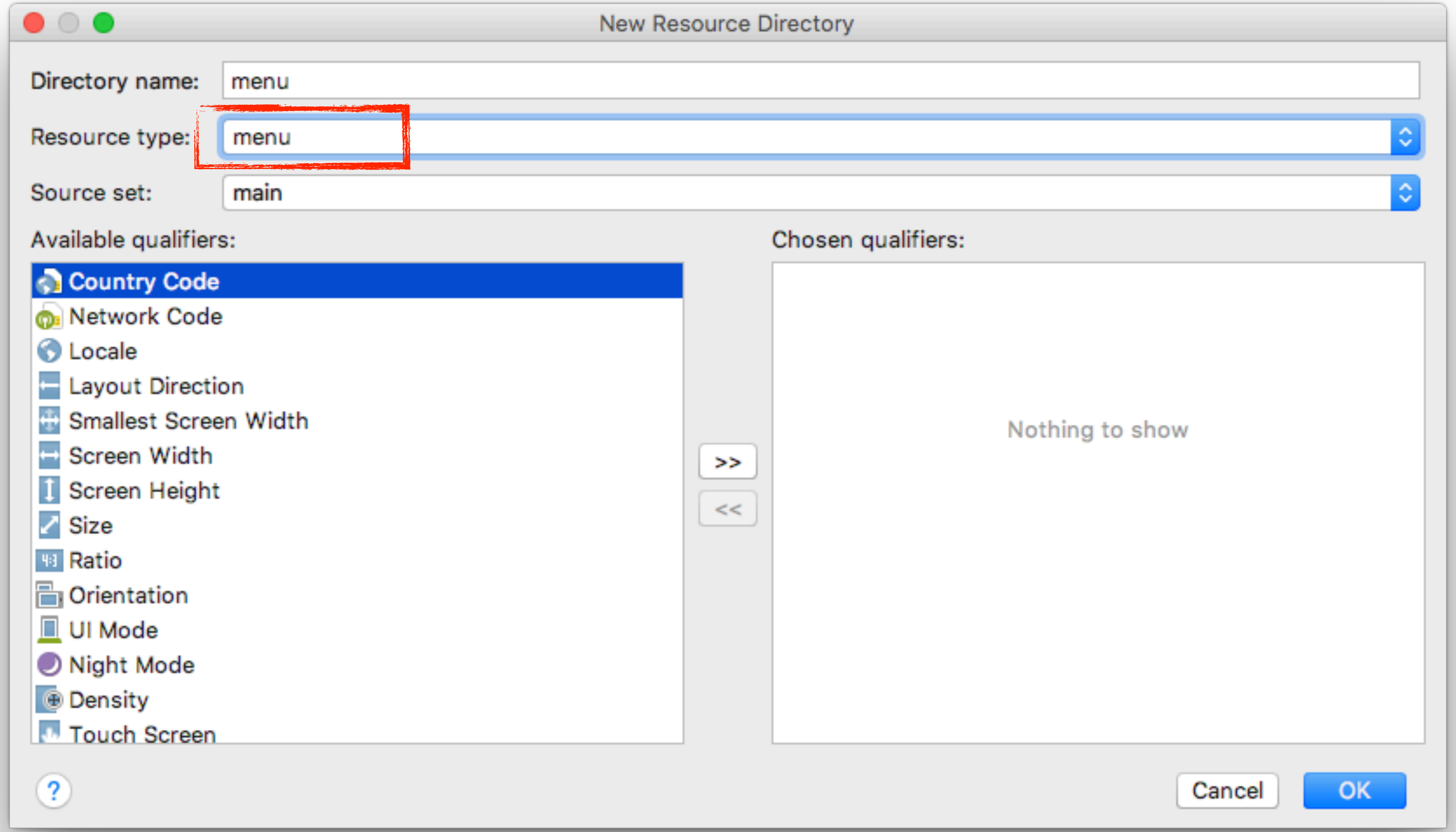
Drink a glass of water

A refreshing glass of water gets you hydrated

Creating a Menu



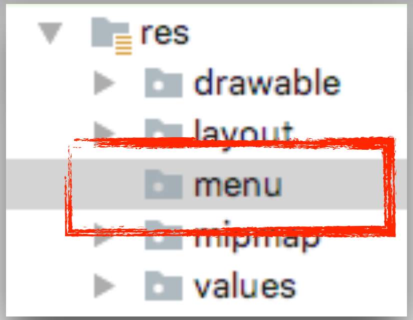
```
val habit = habits[index]
```

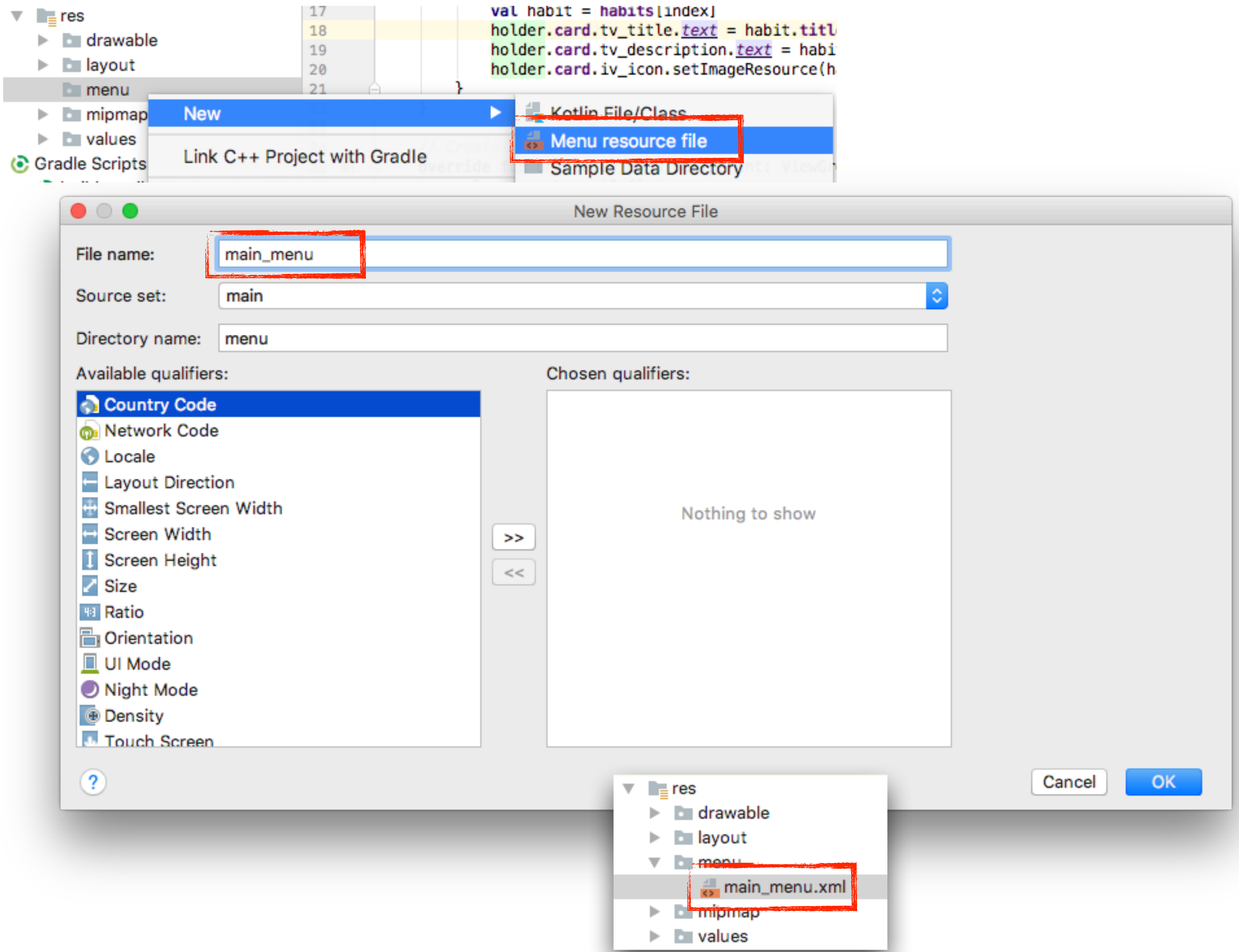

New Resource Directory

Directory name: menu

Resource type: menu

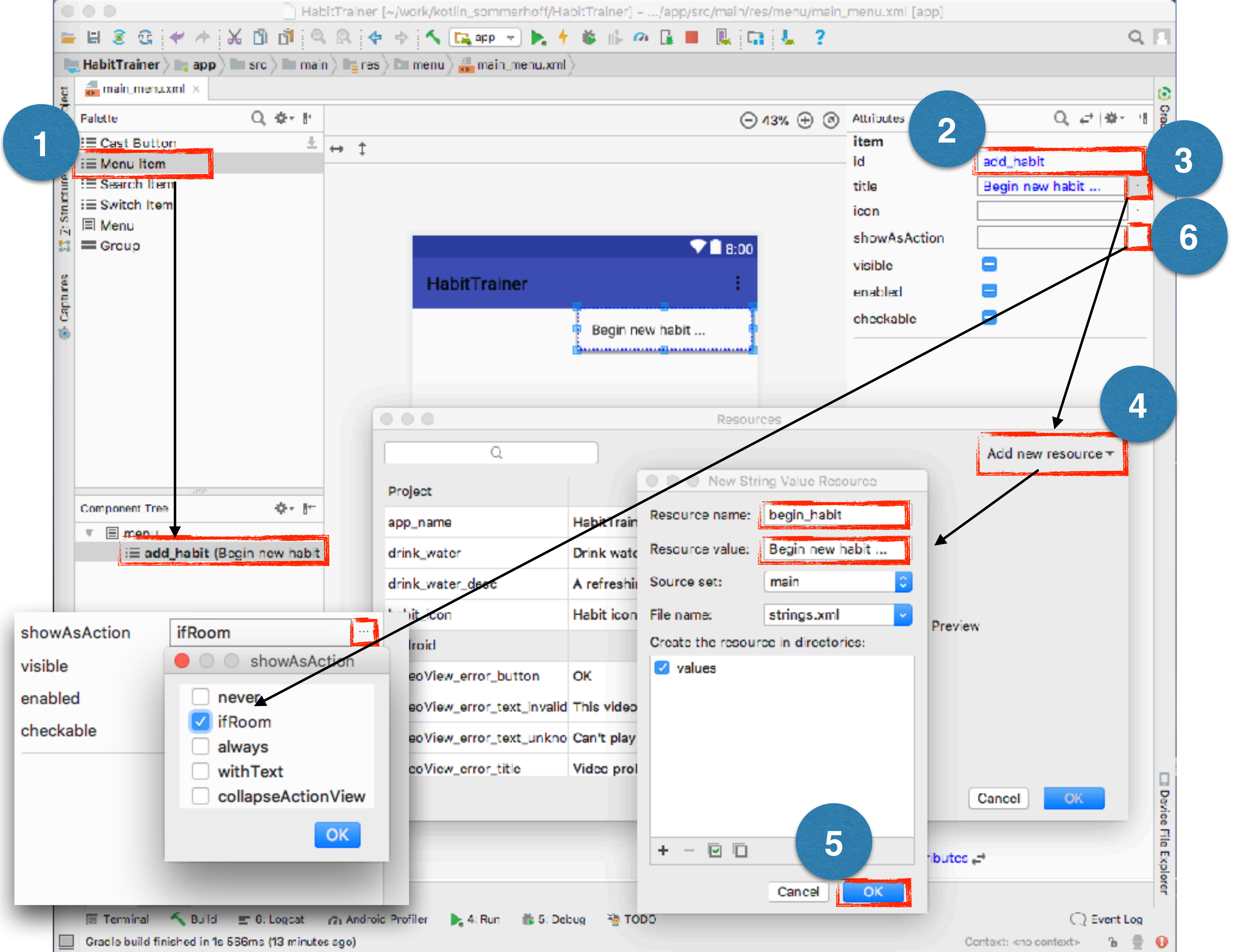
Source set: main





```
        val habit = habits[index]
        holder.card.tv_title.text = habit.titl
        holder.card.tv_description.text = habi
        holder.card.iv_icon.setImageResource(h
    }
```

File name: main_menu
Source set: main
Directory name: menu


HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - .../app/src/main/res/menu/main_menu.xml [app]

HabitTrainer > app > src > main > res > menu > main_menu.xml

item
id: add_habit
title: Begin new habit ...
icon:
showAsAction:
visible
enabled
checkable

Component Tree
menu
add_habit (Begin new habit

Resources

Project	
app_name	HabitTrain
drink_water	Drink wate
drink_water_desc	A refreshi
habit_icon	Habit icon

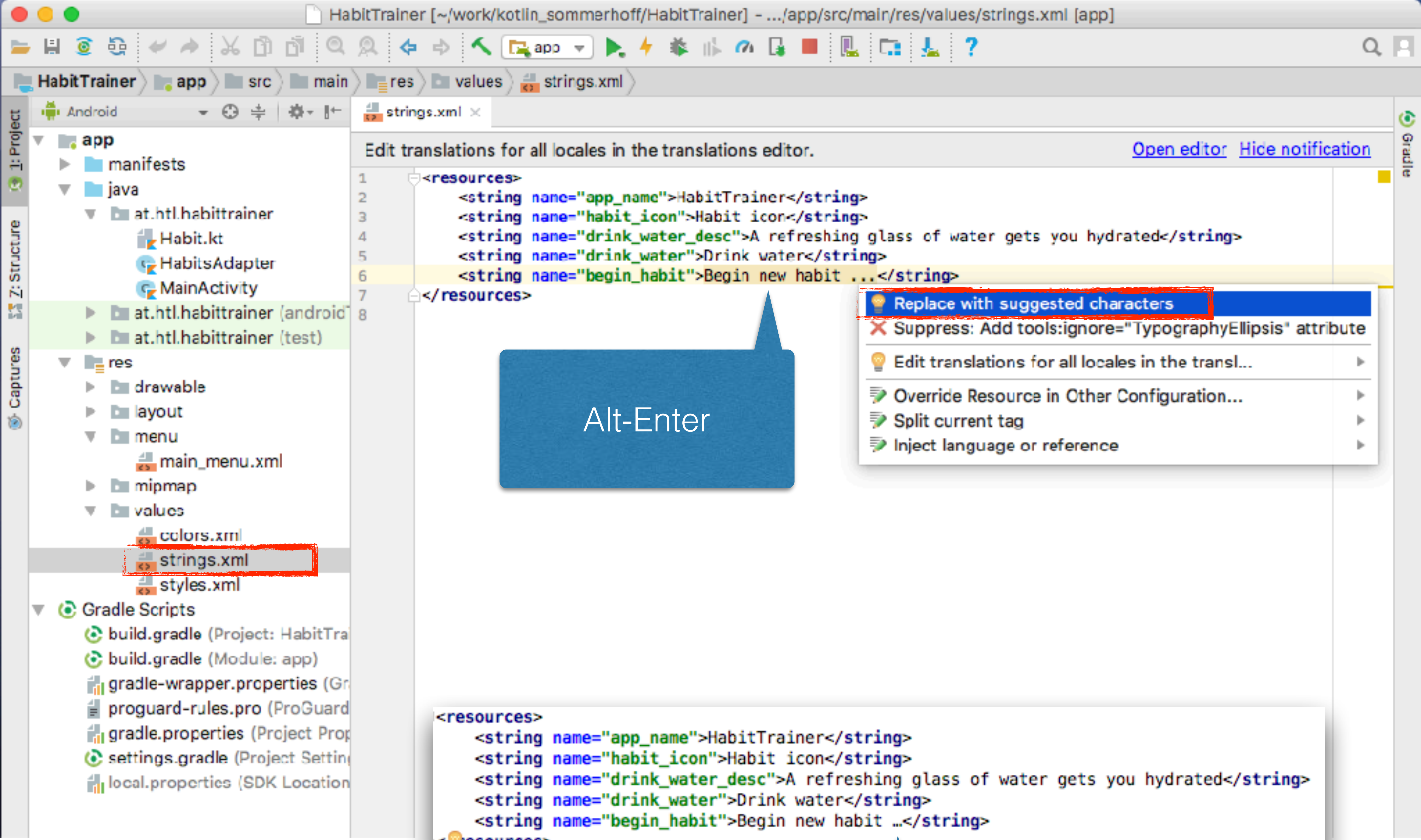
droid	
eoView_error_button	OK
eoView_error_text_invalid	This video
eoView_error_text_unkno	Can't play
eoView_error_title	Video prol

New String Value Resource
Resource name: begin_habit
Resource value: Begin new habit ...
Source set: main
File name: strings.xml
Create the resource in directories:
✓ values

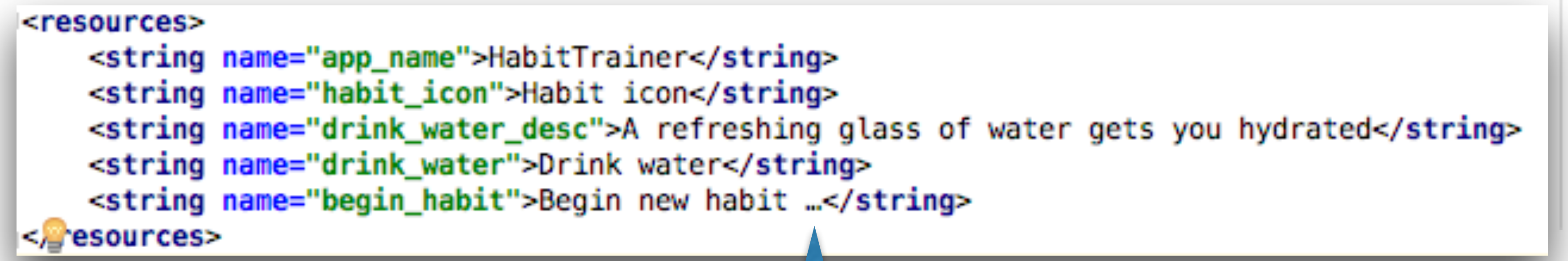
showAsAction: ifRoom
visible
enabled
checkable

showAsAction
☐ never
☑ ifRoom
☐ always
☐ withText
☐ collapseActionView

Gradle build finished in 1s 536ms (13 minutes ago)

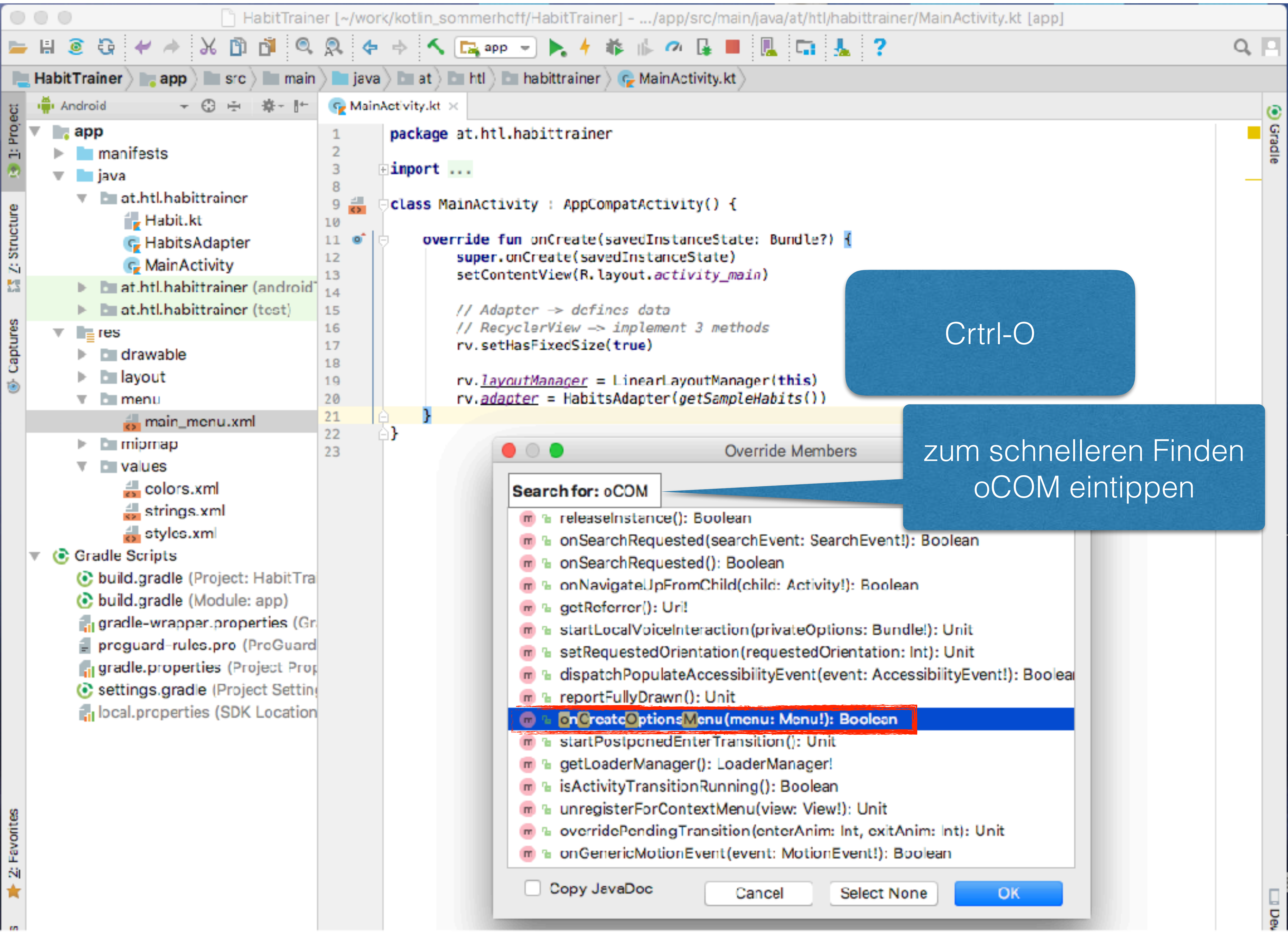

HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - .../app/src/main/res/values/strings.xml [app]
HabitTrainer › app › src › main › res › values › strings.xml
app
manifests
java
at.htl.habittrainer
Habit.kt
HabitsAdapter
MainActivity
at.htl.habittrainer (androidTest)
at.htl.habittrainer (test)
res
drawable
layout
menu
main_menu.xml
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: HabitTrainer)
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro
gradle.properties
settings.gradle
local.properties
Edit translations for all locales in the translations editor.
Open editor
Hide notification
Replace with suggested characters
Suppress: Add tools:ignore="TypographyEllipsis" attribute
Edit translations for all locales in the transl...
Override Resource in Other Configuration...
Split current tag
Inject language or reference
Alt-Enter

```
<resources>
    <string name="app_name">HabitTrainer</string>
    <string name="habit_icon">Habit icon</string>
    <string name="drink_water_desc">A refreshing glass of water gets you hydrated</string>
    <string name="drink_water">Drink water</string>
    <string name="begin_habit">Begin new habit ...</string>
</resources>
```



```
<resources>
    <string name="app_name">HabitTrainer</string>
    <string name="habit_icon">Habit icon</string>
    <string name="drink_water_desc">A refreshing glass of water gets you hydrated</string>
    <string name="drink_water">Drink water</string>
    <string name="begin_habit">Begin new habit …</string>
</resources>
```

Die drei Punkte nehmen nun weniger Platz weg



Crtrl-O

zum schnelleren Finden oCOM eintippen

```
package at.htl.habittrainer

import ...

class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }
}
```

Override Members

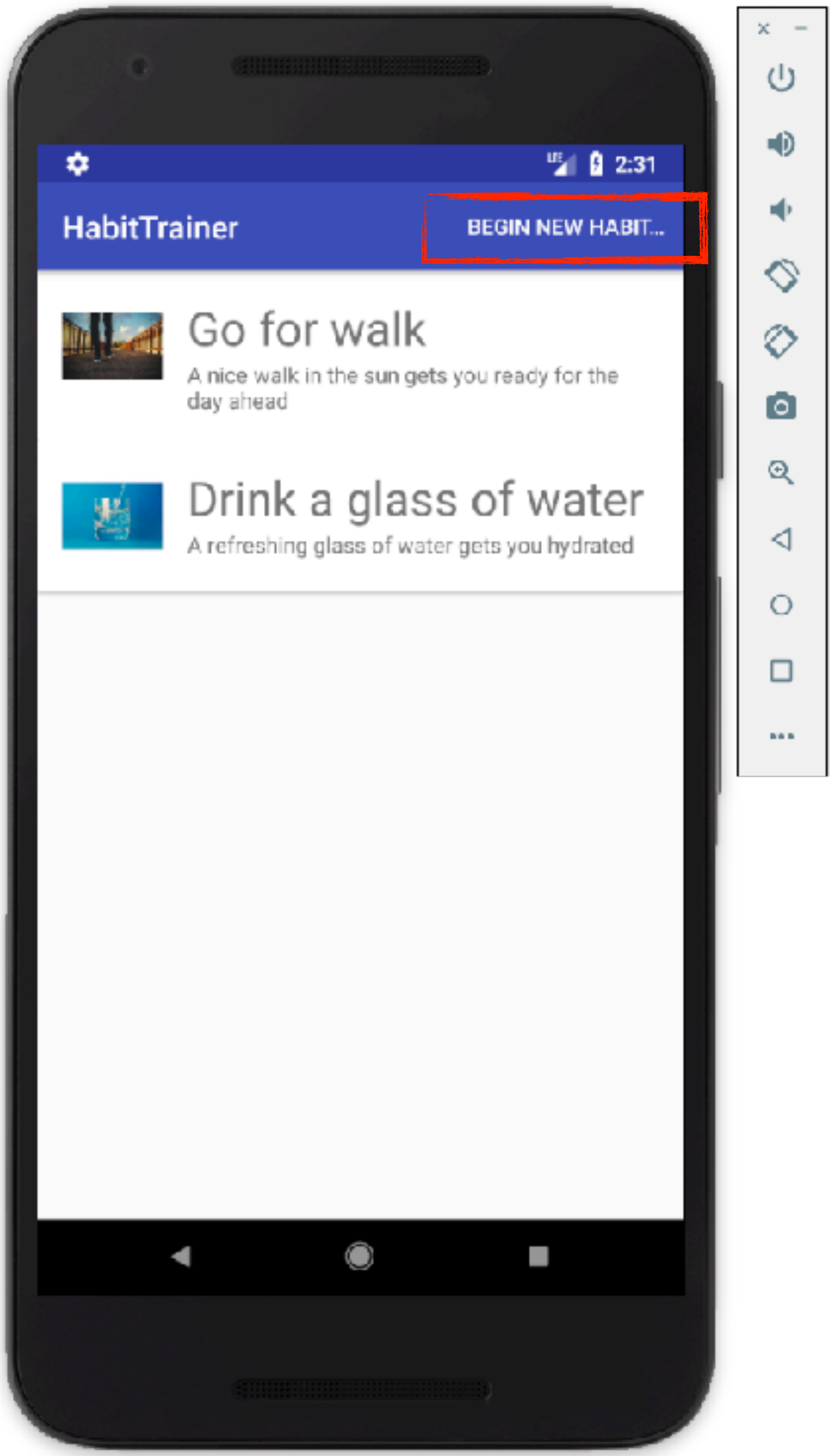
Search for: oCOM

```
releaseInstance(): Boolean
onSearchRequested(searchEvent: SearchEvent!): Boolean
onSearchRequested(): Boolean
onNavigateUpFromChild(child: Activity!): Boolean
getReferrer(): Uri!
startLocalVoiceInteraction(privateOptions: Bundle!): Unit
setRequestedOrientation(requestedOrientation: Int): Unit
dispatchPopulateAccessibilityEvent(event: AccessibilityEvent!): Boolean
reportFullyDrawn(): Unit
onCreateOptionsMenu(menu: Menu!): Boolean
startPostponedEnterTransition(): Unit
getLoaderManager(): LoaderManager!
isActivityTransitionRunning(): Boolean
unregisterForContextMenu(view: View!): Unit
overridePendingTransition(enterAnim: Int, exitAnim: Int): Unit
onGenericMotionEvent(event: MotionEvent!): Boolean
```

HabitTrainer [~/work/kotlin_sommerhcft/HabitTrainer] - .../app/src/main/java/at/htl/habittrainer/MainActivity.kt [app]
HabitTrainer > app > src > main > java > at > htl > habittrainer > MainActivity.kt
Android
MainActivity.kt
app
manifests
java
at.htl.habittrainer
Habit.kt
HabitsAdapter
MainActivity
at.htl.habittrainer (androidTest)
at.htl.habittrainer (test)
res
drawable
layout
menu
main_menu.xml
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: HabitTrainer)
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro (ProGuard
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Copy JavaDoc
Cancel
Select None
OK

MainActivity.kt

```
package at.htl.habittrainer

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v7.widget.LinearLayoutManager
import android.view.Menu
import android.widget.TextView
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }

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



Go for walk

A nice walk in the sun gets you ready for the day ahead

Drink a glass of water

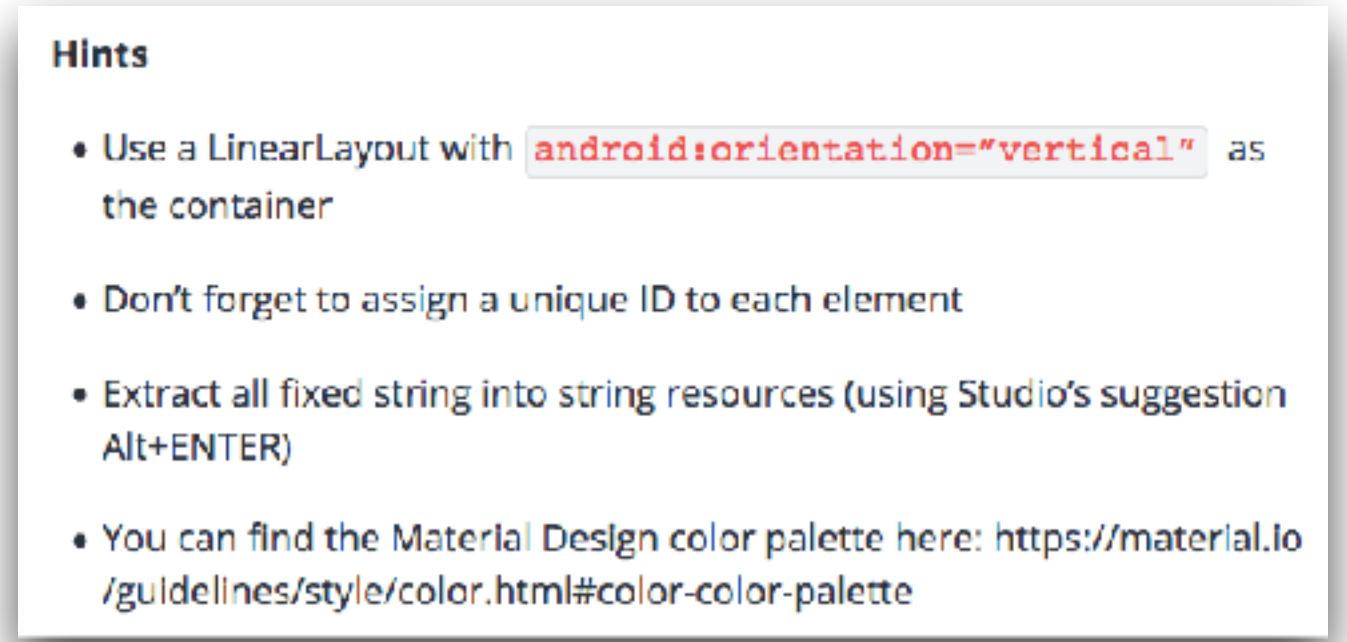
A refreshing glass of water gets you hydrated

Detail input form

Challenge: Build the Activity Layout

Challenge: Build a Layout that Lets Users Create Habits

- Try to build a layout (no functionality, just XML) with
  - An `EditText` for the title
  - An `EditText` for the description
  - A `Button` to choose an image
  - An `ImageView` to preview the selected image
  - A `Button` to save the new habit
  - A `TextView` to show possible error messages to the user
  - --
  - Try to make the `EditText` for the description span two lines "using inputType"
  - Try to make the text color of the error `TextView` red (look up a color code from Google's Material Design Guidelines)

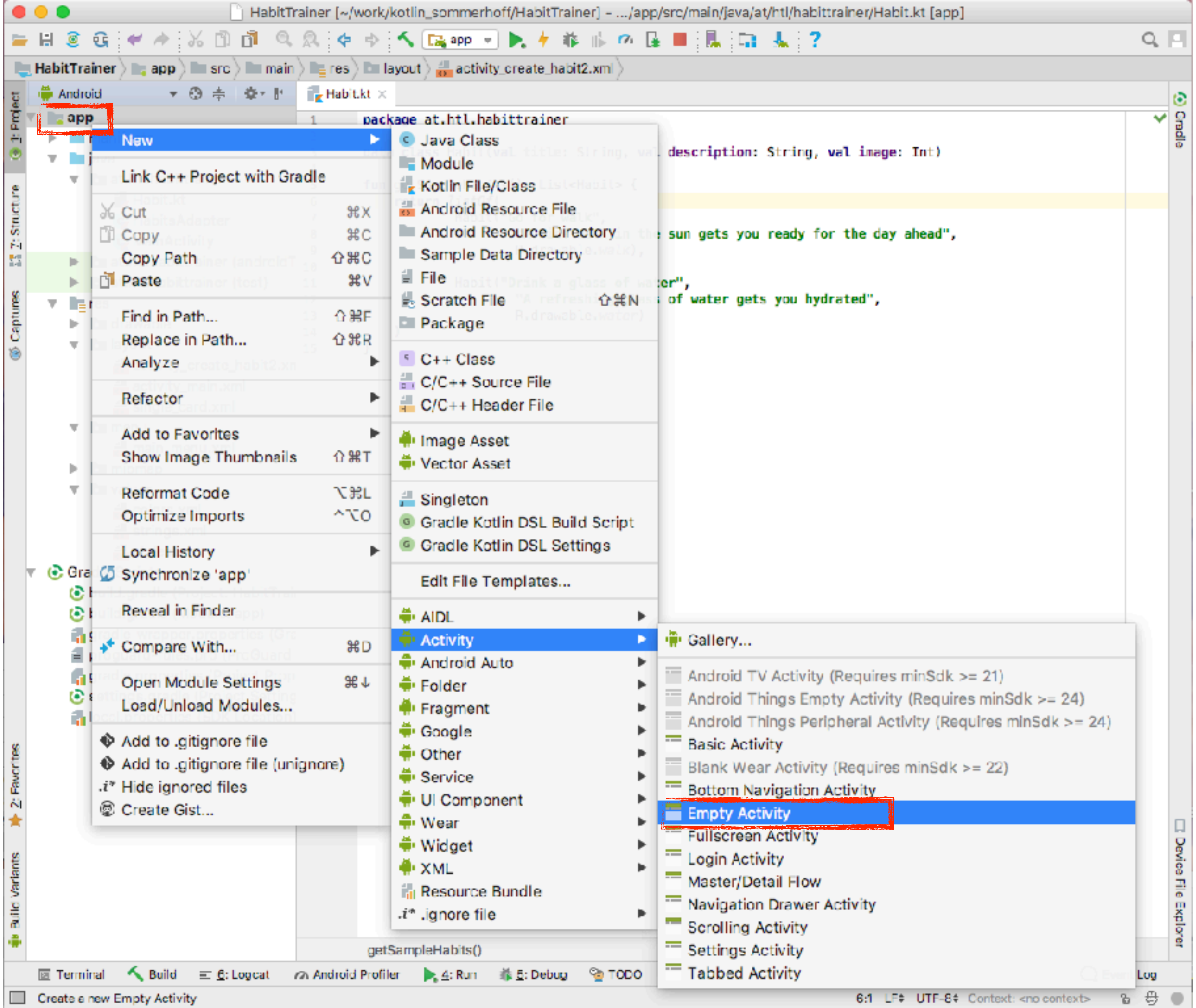


Hints

- Use a LinearLayout with `android:orientation="vertical"` as the container
- Don't forget to assign a unique ID to each element
- Extract all fixed string into string resources (using Studio's suggestion Alt+ENTER)
- You can find the Material Design color palette here: https://material.io/guidelines/style/color.html#color-color-palette

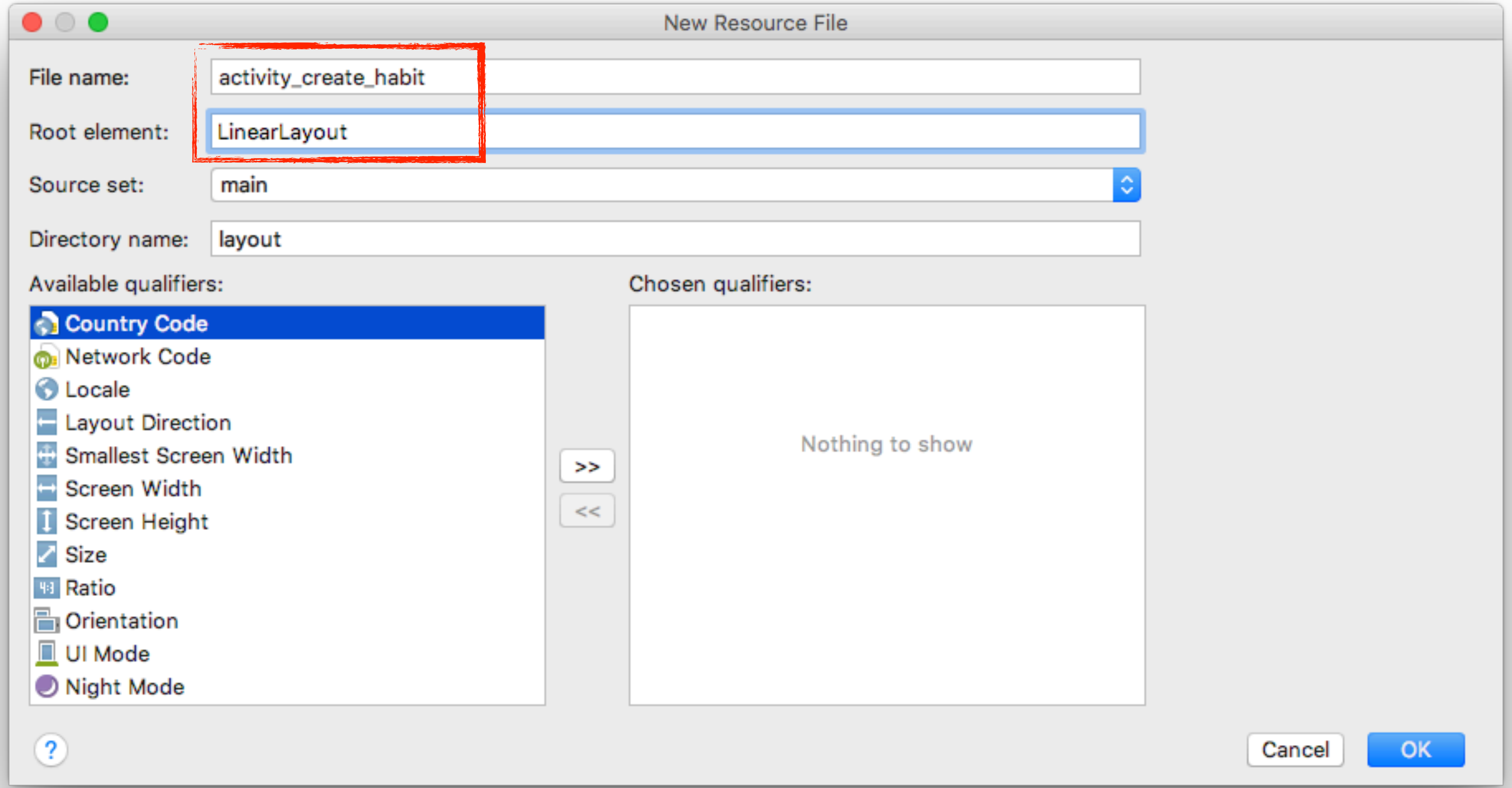
Linear Layout

Variante 1



```
package at.htl.habittrainer
```

Layout File anlegen


New Resource File

File name: activity_create_habit

Root element: LinearLayout

Source set: main

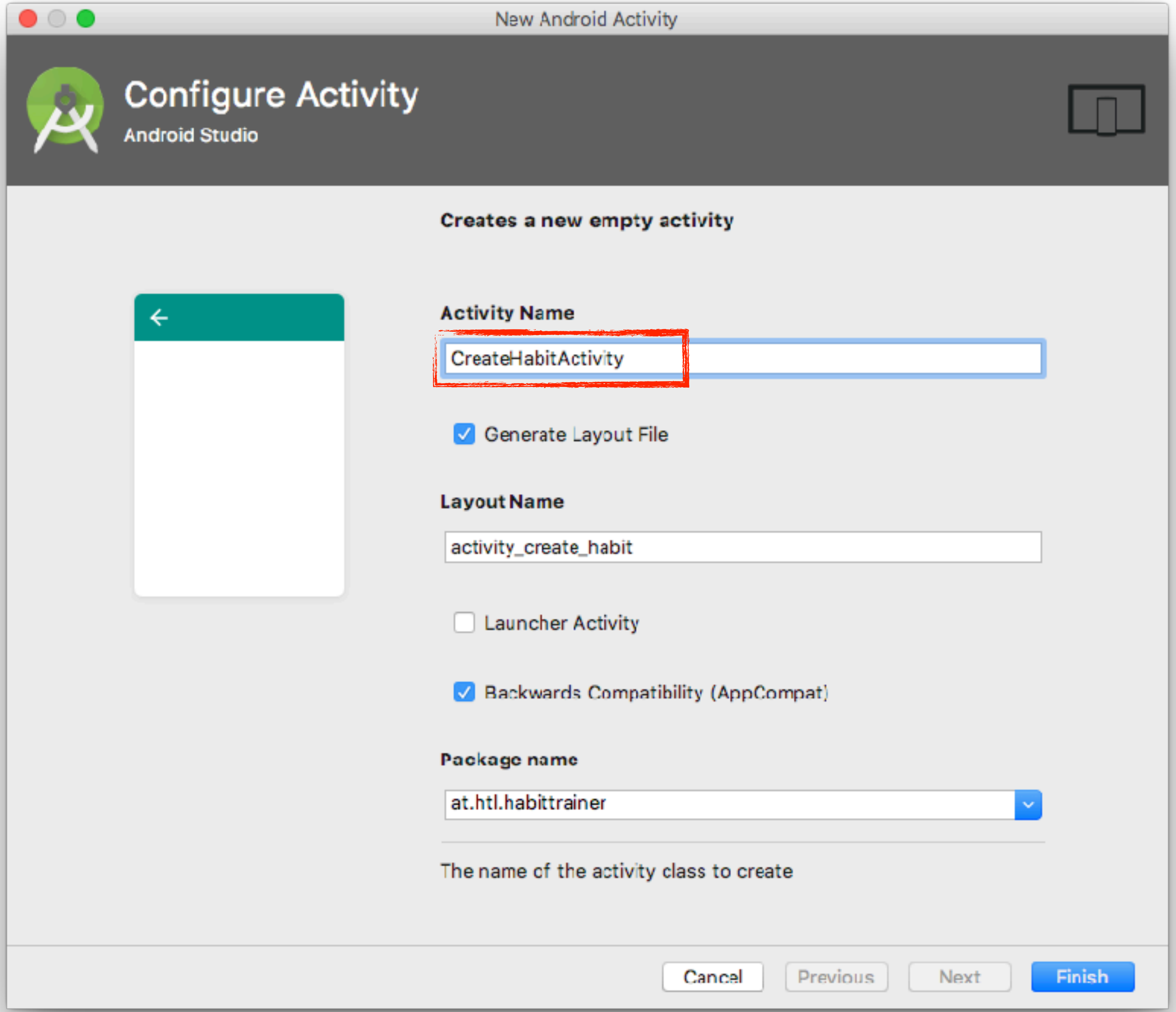
Directory name: layout

Available qualifiers:

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

Chosen qualifiers:

Nothing to show



Creates a new empty activity

Activity Name: CreateHabitActivity

✓ Generate Layout File

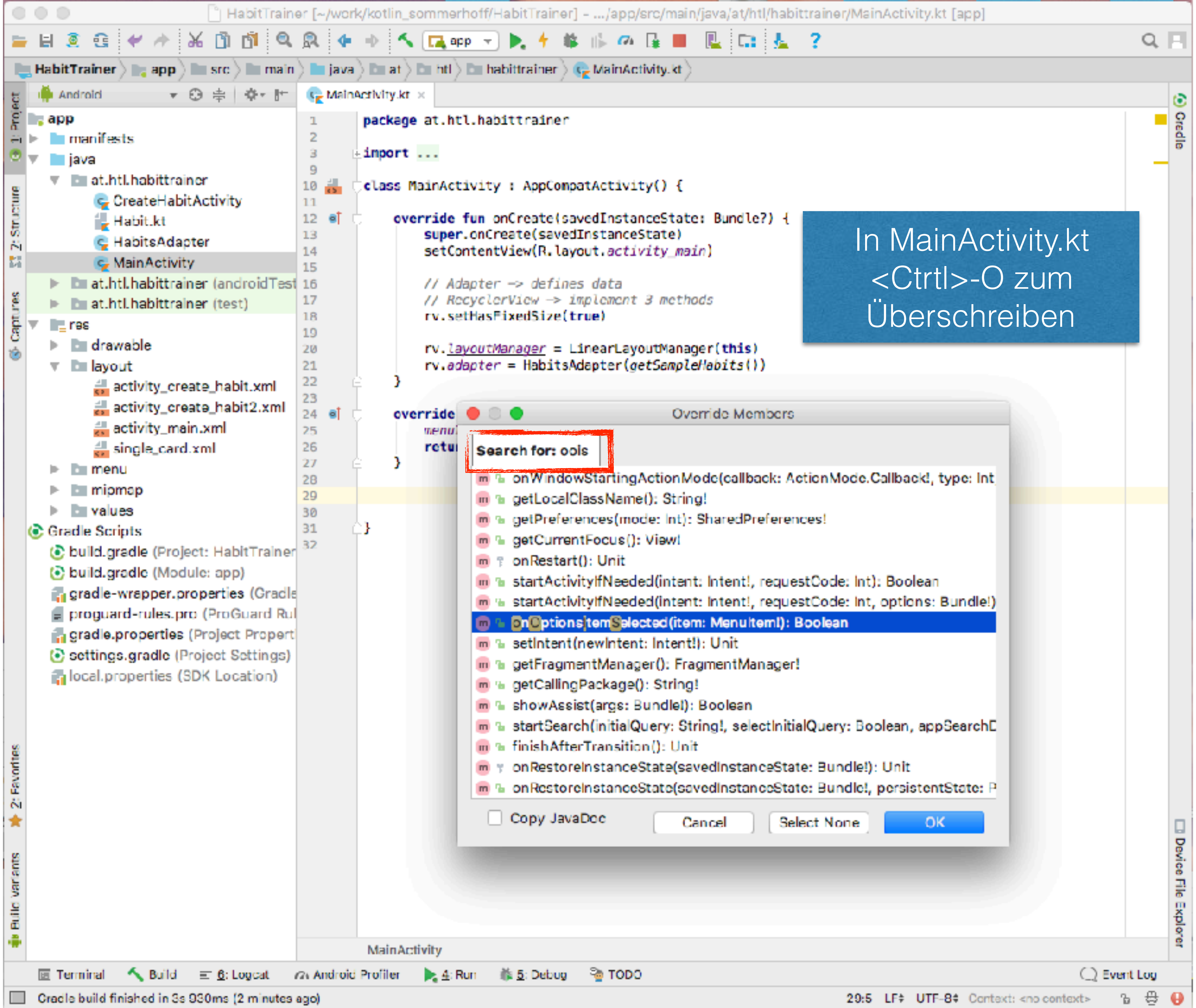
Layout Name: activity_create_habit

☐ Launcher Activity

✓ Backwards Compatibility (AppCompat)

Package name: at.htl.habittrainer

The name of the activity class to create



In MainActivity.kt <Ctrtl>-O zum Überschreiben

```
package at.htl.habittrainer

import ...

class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }

    override
        menu
        retu
    }

}
```

Override Members

Search for: oois

```
onWindowStartingActionMode(callback: ActionMode.Callback!, type: Int
getLocalClassName(): String!
getPreferences(mode: Int): SharedPreferences!
getCurrentFocus(): View!
onRestart(): Unit
startActivityIfNeeded(intent: Intent!, requestCode: Int): Boolean
startActivityIfNeeded(intent: Intent!, requestCode: Int, options: Bundle!)
onOptionsItemSelected(item: MenuItem!): Boolean
setIntent(newIntent: Intent!): Unit
getFragmentManager(): FragmentManager!
getCallingPackage(): String!
showAssist(args: Bundle!): Boolean
startSearch(initialQuery: String!, selectInitialQuery: Boolean, appSearchD
finishAfterTransition(): Unit
onRestoreInstanceState(savedInstanceState: Bundle!): Unit
onRestoreInstanceState(savedInstanceState: Bundle!, persistentState: P
```

Copy JavaDoc | Cancel | Select None | OK

Gradle build finished in 3s 030ms (2 minutes ago)

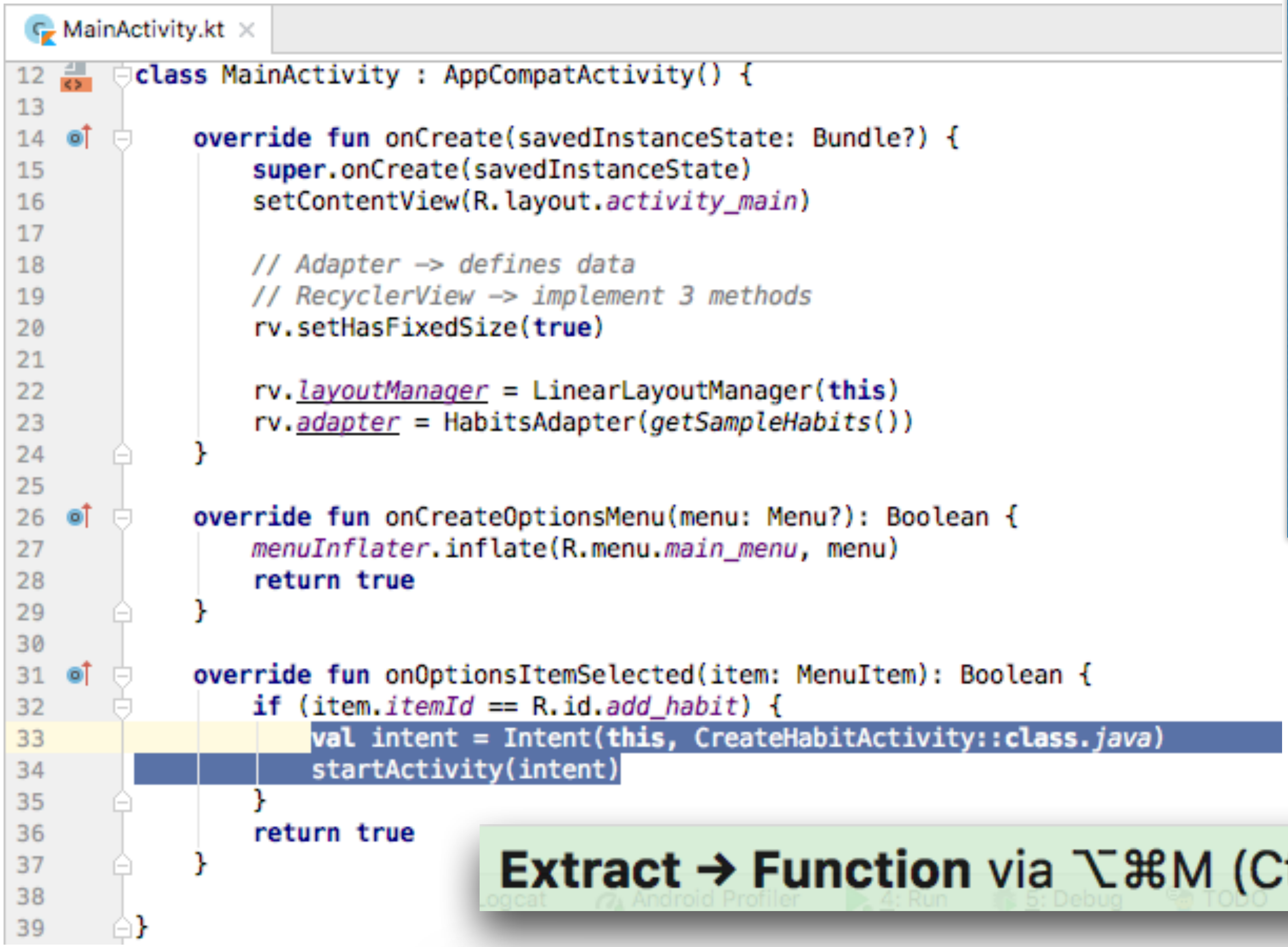
Intents

- implizite Intents
- explizite Intents

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }

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

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.add_habit) {
            val intent = Intent(this, CreateHabitActivity::class.java)
            startActivity(intent)
        }
        return true
    }
}
```

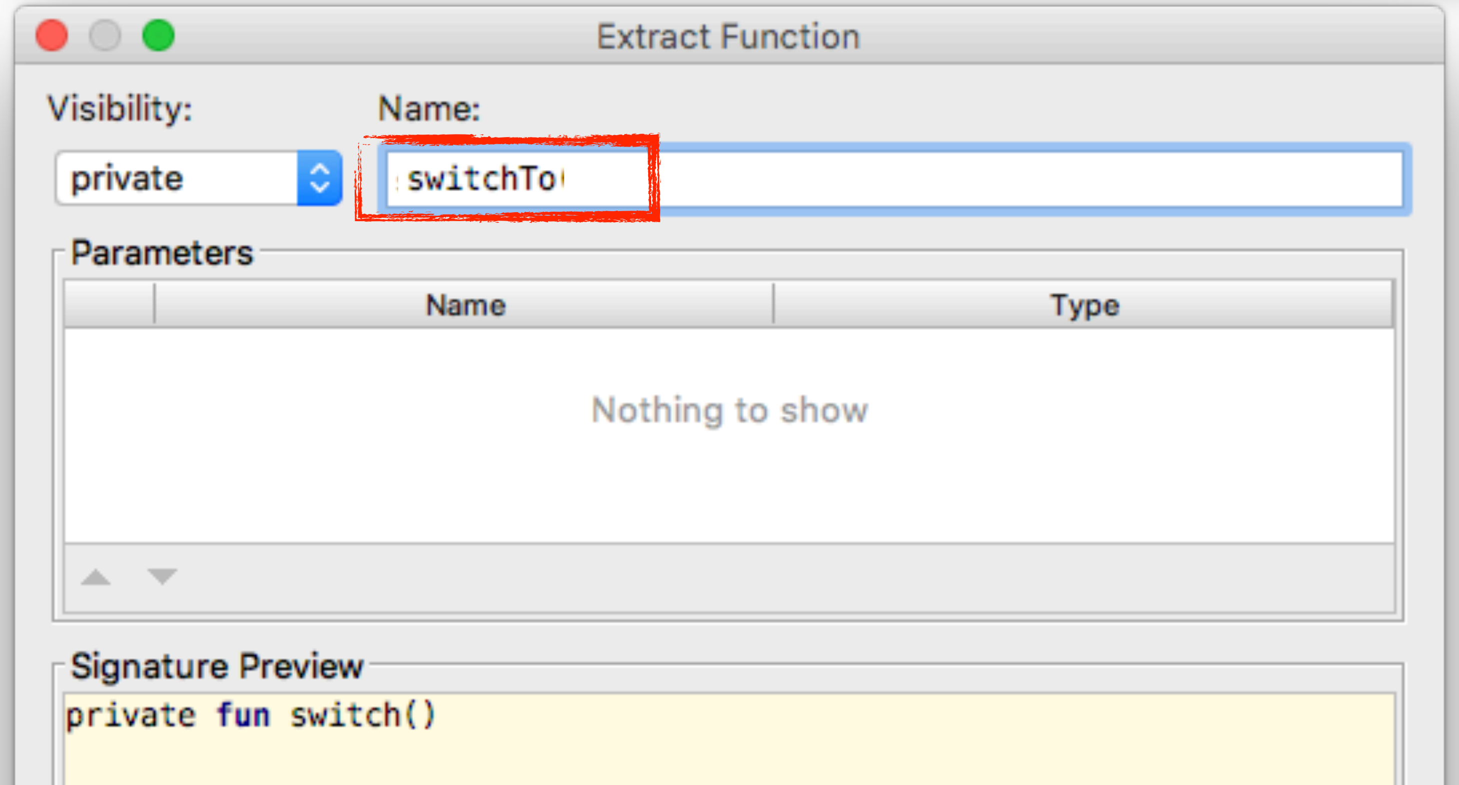
„::class.java“ um Zugriff auf eine Java-Klasse im Bytecode zu erhalten



```kotlin
class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
    }

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

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.add_habit) {
            val intent = Intent(this, CreateHabitActivity::class.java)
            startActivity(intent)
        }
        return true
    }
}
```

1. if-Körper markieren
2. <Alt><Cmd>-M drücken
3. Namen der Funktion „switchTo“ eingeben
4. Ok
5. Parameter ergänzen (nächste Folie)

Extract → Function via ⌥⌘M (Ctrl+Alt+M for Win/Linux)



Visibility: private

Name: switchTo

```kotlin
private fun switch()
```

MainActivity.kt - switchTo()

```kotlin
class MainActivity : AppCompatActivity() {

    ...

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.add_habit) {
            switchTo(CreateHabitActivity::class.java)
        }
        return true
    }

    private fun switchTo(c: Class<*>) {
        val intent = Intent(this, c)
        startActivity(intent)
    }
}
```

Es wird eine beliebige Klasse als Parameter übergeben

activity_create_habit.xml

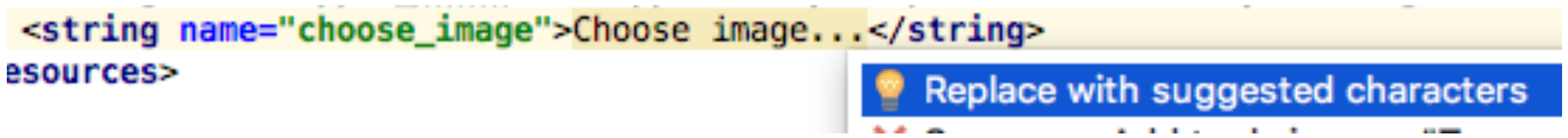
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <EditText
        android:id="@+id/et_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/eat_an_apple" />

    <EditText
        android:id="@+id/et_descr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/apple_descr"
        android:inputType="textMultiLine"
        android:lines="2" />

    <Button
        android:id="@+id/btn_choose_image"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/choose_image" />

    <ImageView
        android:id="@+id/iv_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:contentDescription="@string/selected_image"
        android:padding="10dp" />

    <Button
        android:id="@+id/btn_save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/save"
        />

    <TextView
        android:id="@+id/tv_error"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:visibility="invisible"
        android:textColor="#e53935" />
</LinearLayout>
```

Erstellen Sie dieses Layout. Sie können hierfür natürlich den Designer verwenden.

Untenstehend ersehen Sie die benötigten Textressourcen in values/strings.xml

strings.xml

```
<string name="eat_an_apple">Eat an apple</string>
<string name="apple_descr">An apple a day keeps the doctor away</string
<string name="choose_image">Choose image…</string>
<string name="selected_image">Selected image</string>
<string name="save">Save</string>
```

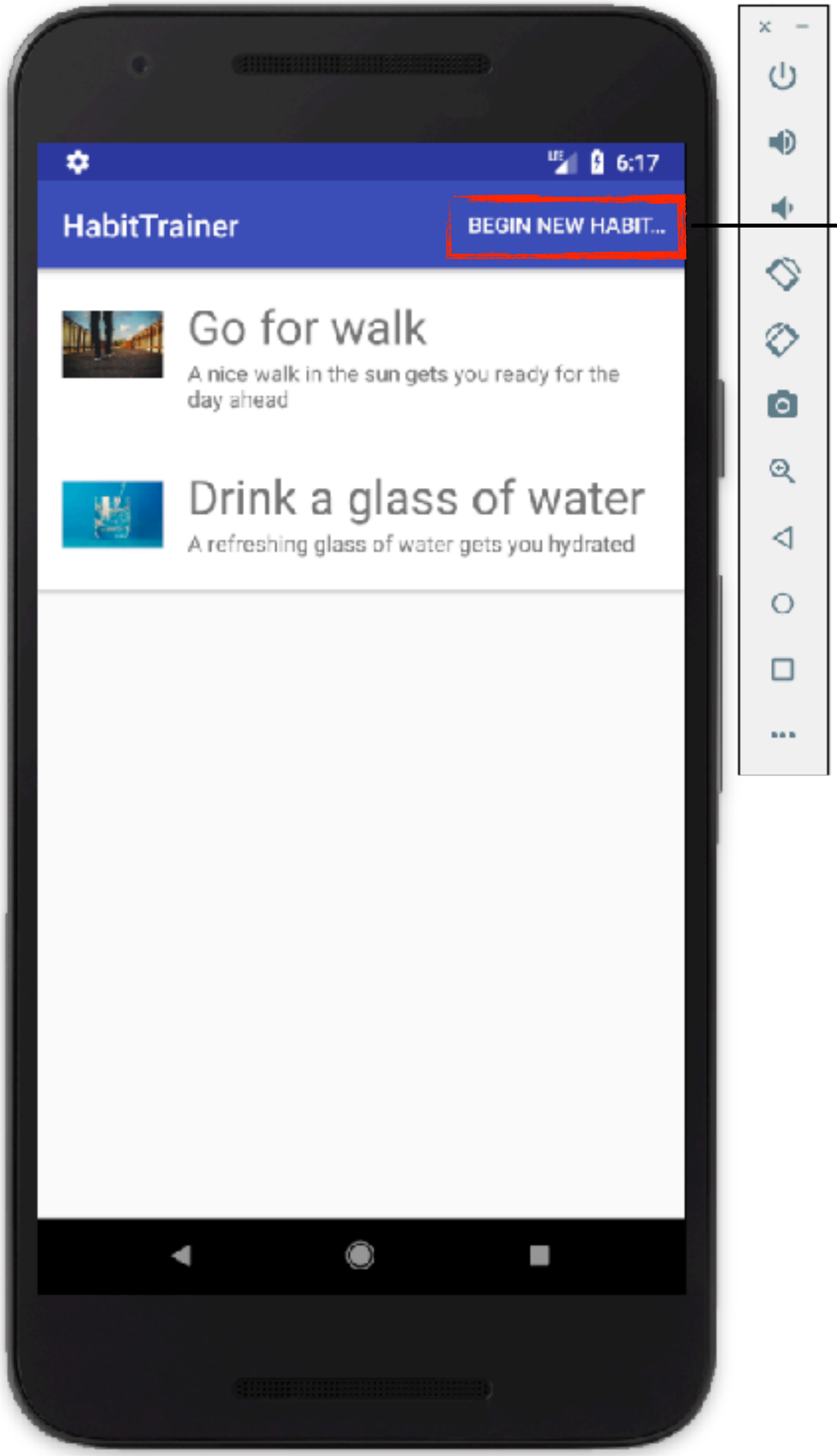


```
<string name="choose_image">Choose image...</string>
esources>
```

Replace with suggested characters

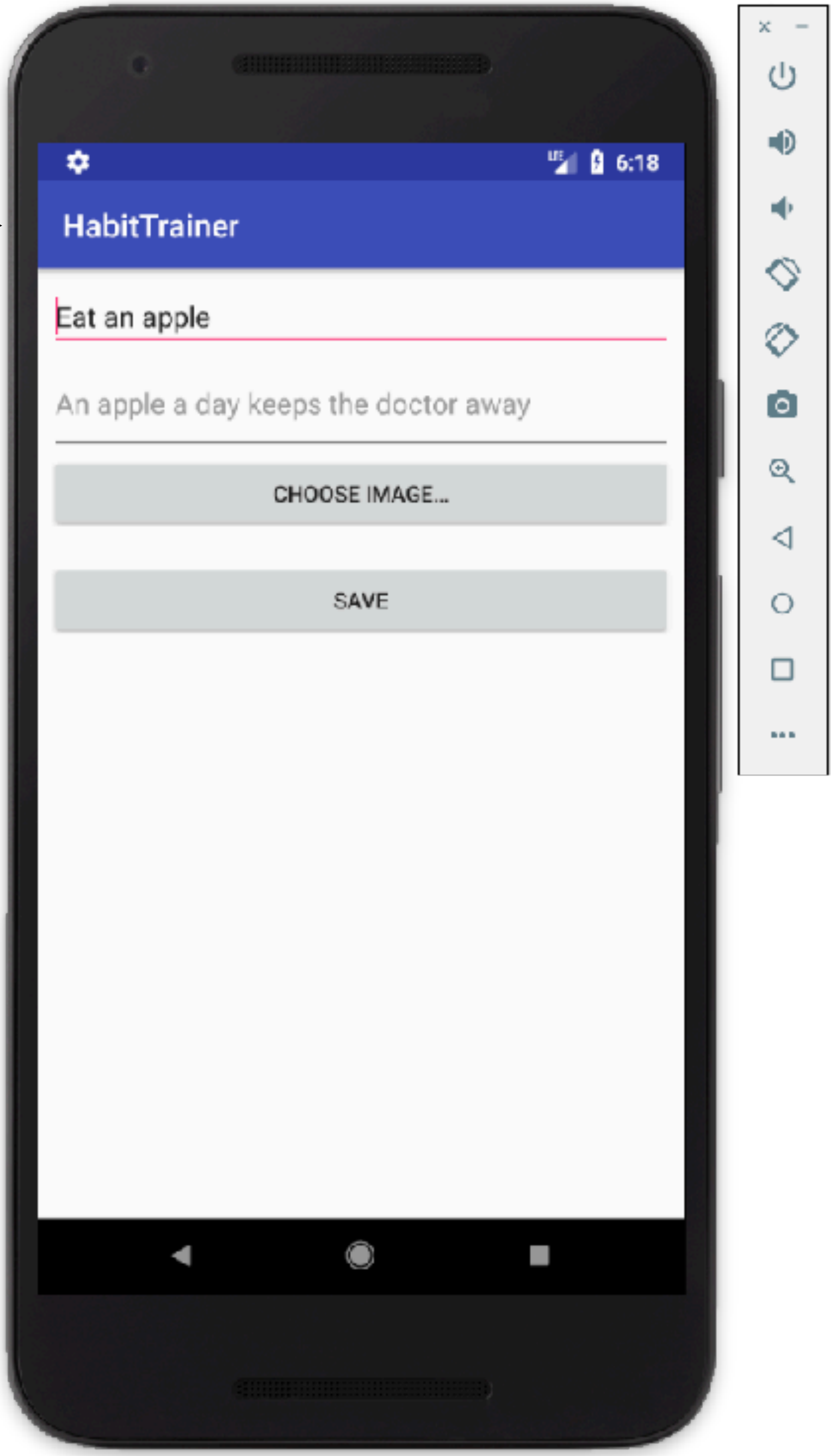
Guidelines für Layout

- https://material.io/guidelines/style/color.html



Go for walk
A nice walk in the sun gets you ready for the day ahead

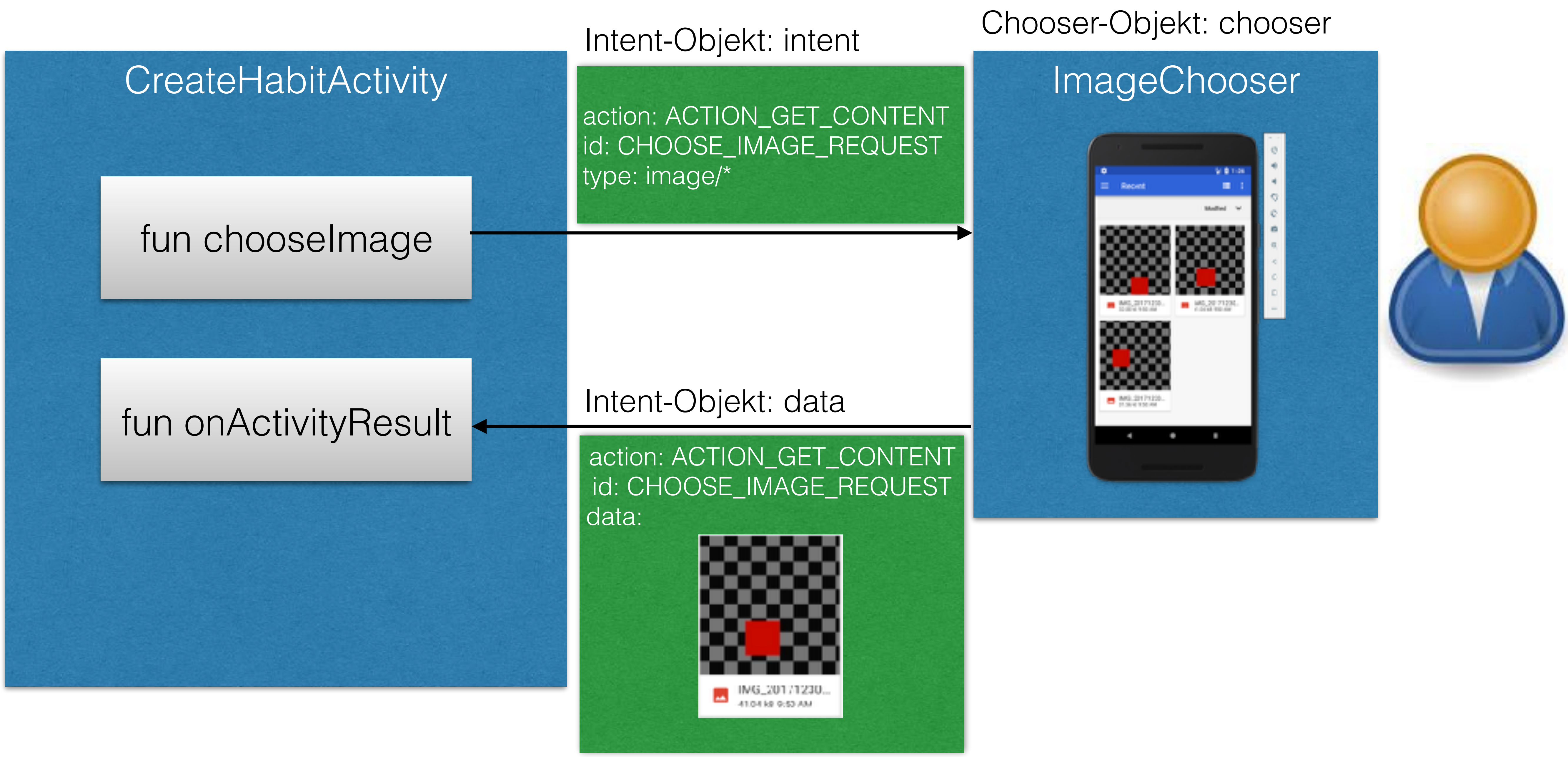
Drink a glass of water
A refreshing glass of water gets you hydrated



Eat an apple

An apple a day keeps the doctor away

Intents


CreateHabitActivity
fun chooseImage
fun onActivityResult
Intent-Objekt: intent
action: ACTION_GET_CONTENT
id: CHOOSE_IMAGE_REQUEST
type: image/*
Chooser-Objekt: chooser
ImageChooser
Intent-Objekt: data
action: ACTION_GET_CONTENT
id: CHOOSE_IMAGE_REQUEST
data:

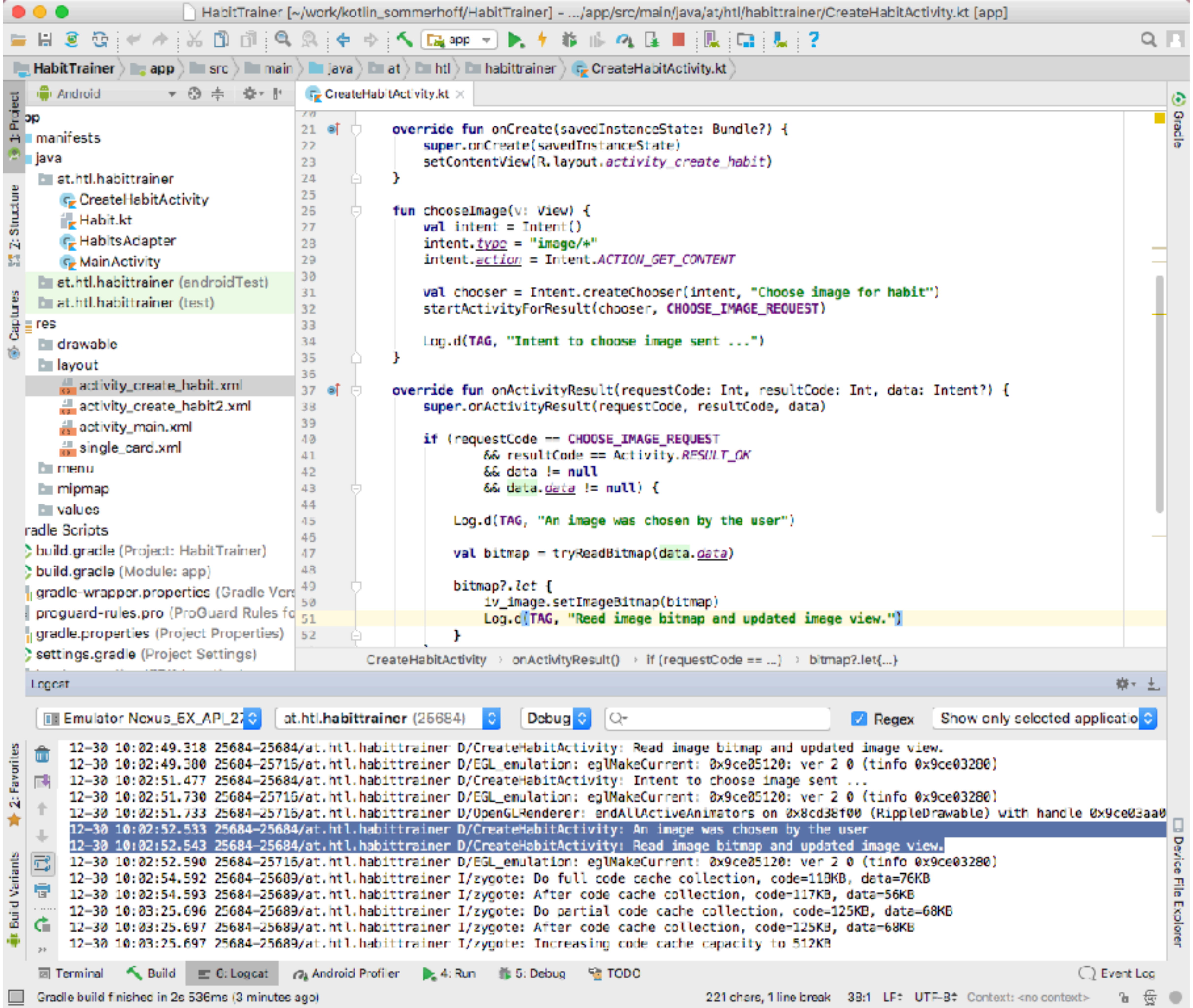
onClick-Methode 1

```kotlin
package at.htl.habittrainer

...
import android.os.Bundle
import android.provider.MediaStore
import android.util.Log
import android.view.View
import kotlinx.android.synthetic.main.activity_create_habit.*
import java.io.IOException

class CreateHabitActivity : AppCompatActivity() {

    private val TAG = CreateHabitActivity::class.java.simpleName

    private val CHOOSE_IMAGE_REQUEST = 4711

    override fun onCreate(savedInstanceState: Bundle?) {
        ...
    }

    fun chooseImage(v: View) {
        val intent = Intent()
        intent.type = "image/*"
        intent.action = Intent.ACTION_GET_CONTENT

        val chooser = Intent.createChooser(intent, "Choose image for habit")
        startActivityForResult(chooser, CHOOSE_IMAGE_REQUEST)

        Log.d(TAG, "Intent to choose image sent ...")
    }

    ...
}
```

onClick-Methode 2

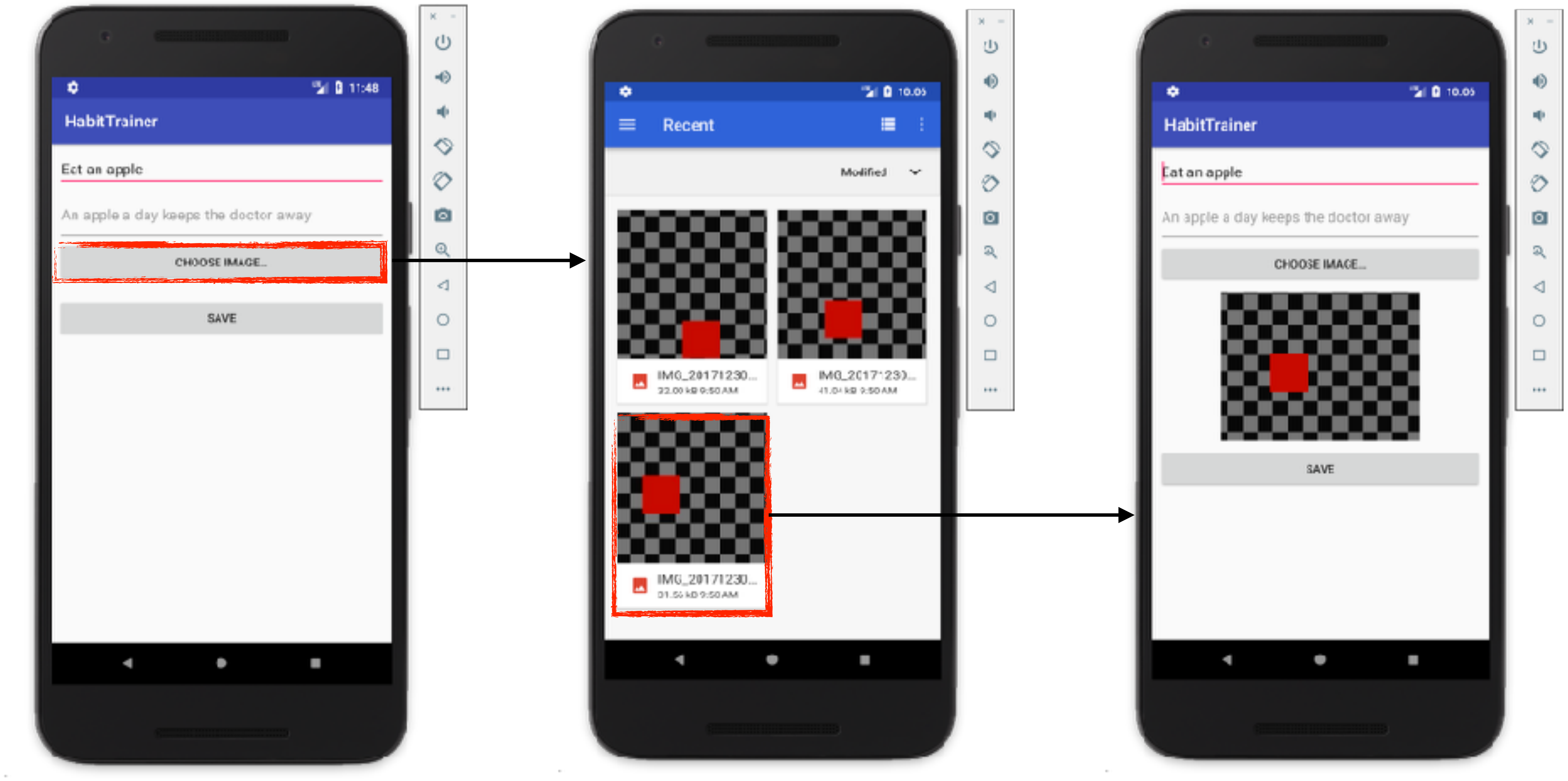
```
...

class CreateHabitActivity : AppCompatActivity() {

    ...

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)

        if (requestCode == CHOOSE_IMAGE_REQUEST
                && resultCode == Activity.RESULT_OK
                && data != null
                && data.data != null) {

            Log.d(TAG, "An image was chosen by the user")

            val bitmap = tryReadBitmap(data.data)

            bitmap?.let {
                iv_image.setImageBitmap(bitmap)
                Log.d(TAG, "Read image bitmap and updated image view.")
            }
        }
    }

    fun tryReadBitmap(data: Uri): Bitmap? {
        return try {
            MediaStore.Images.Media.getBitmap(contentResolver, data)
        } catch (e: IOException) {
            e.printStackTrace()
            null
        }
    }
}
```

Der Code im let-Block wird nur ausgeführt, wenn bitmap != null ist


HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - .../app/src/main/java/at/htl/habittrainer/CreateHabitActivity.kt [app]
HabitTrainer › app › src › main › java › at › htl › habittrainer › CreateHabitActivity.kt
Android
CreateHabitActivity.kt
manifests
java
at.htl.habittrainer
CreateHabitActivity
Habit.kt
HabitsAdapter
MainActivity
at.htl.habittrainer (androidTest)
at.htl.habittrainer (test)
res
drawable
layout
activity_create_habit.xml
activity_create_habit2.xml
activity_main.xml
single_card.xml
menu
mipmap
values
Gradle Scripts
build.gradle (Project: HabitTrainer)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
CreateHabitActivity › onActivityResult() › if (requestCode == ...) › bitmap?.let{...}
Logcat
Emulator Nexus_5X_API_27
at.htl.habittrainer (25684)
Debug
Regex
Show only selected application
Terminal
Build
6: Logcat
Android Profiler
4: Run
5: Debug
TODO
Event Log
Gradle build finished in 2s 536ms (3 minutes ago)
221 chars, 1 line break 38:1 LF: UTF-8: Context: <no context>

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_create_habit)
    }

    fun chooseImage(v: View) {
        val intent = Intent()
        intent.type = "image/*"
        intent.action = Intent.ACTION_GET_CONTENT

        val chooser = Intent.createChooser(intent, "Choose image for habit")
        startActivityForResult(chooser, CHOOSE_IMAGE_REQUEST)

        Log.d(TAG, "Intent to choose image sent ...")
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)

        if (requestCode == CHOOSE_IMAGE_REQUEST
                && resultCode == Activity.RESULT_OK
                && data != null
                && data.data != null) {

            Log.d(TAG, "An image was chosen by the user")

            val bitmap = tryReadBitmap(data.data)

            bitmap?.let {
                iv_image.setImageBitmap(bitmap)
                Log.d(TAG, "Read image bitmap and updated image view.")
            }
```

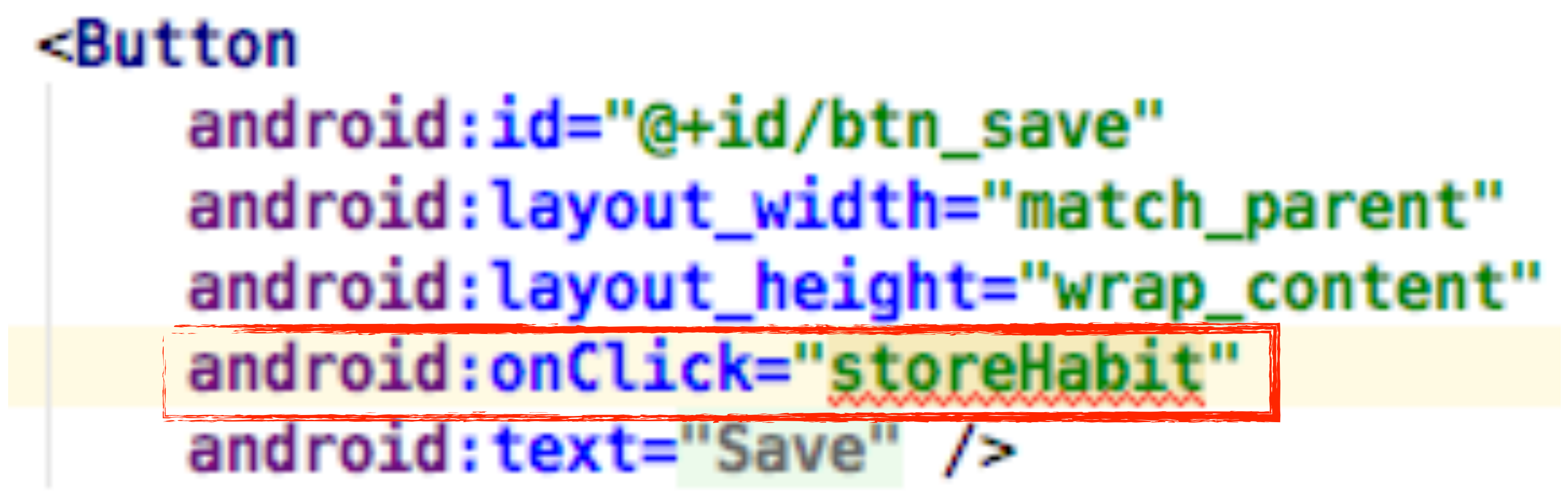
```
12-30 10:02:49.318 25684-25684/at.htl.habittrainer D/CreateHabitActivity: Read image bitmap and updated image view.
12-30 10:02:49.380 25684-25716/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0x9ce05120: ver 2 0 (tinfo 0x9ce03280)
12-30 10:02:51.477 25684-25684/at.htl.habittrainer D/CreateHabitActivity: Intent to choose image sent ...
12-30 10:02:51.730 25684-25716/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0x9ce05120: ver 2 0 (tinfo 0x9ce03280)
12-30 10:02:51.733 25684-25716/at.htl.habittrainer D/OpenGLRenderer: endAllActiveAnimators on 0x8cd38f00 (RippleDrawable) with handle 0x9ce03aa0
12-30 10:02:52.533 25684-25684/at.htl.habittrainer D/CreateHabitActivity: An image was chosen by the user
12-30 10:02:52.543 25684-25684/at.htl.habittrainer D/CreateHabitActivity: Read image bitmap and updated image view.
12-30 10:02:52.590 25684-25716/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0x9ce05120: ver 2 0 (tinfo 0x9ce03280)
12-30 10:02:54.592 25684-25689/at.htl.habittrainer I/zygote: Do full code cache collection, code=118KB, data=76KB
12-30 10:02:54.593 25684-25689/at.htl.habittrainer I/zygote: After code cache collection, code=117KB, data=56KB
12-30 10:03:25.696 25684-25689/at.htl.habittrainer I/zygote: Do partial code cache collection, code=125KB, data=68KB
12-30 10:03:25.697 25684-25689/at.htl.habittrainer I/zygote: After code cache collection, code=125KB, data=68KB
12-30 10:03:25.697 25684-25689/at.htl.habittrainer I/zygote: Increasing code cache capacity to 512KB
```


HabitTrainer

Eat an apple

An apple a day keeps the doctor away

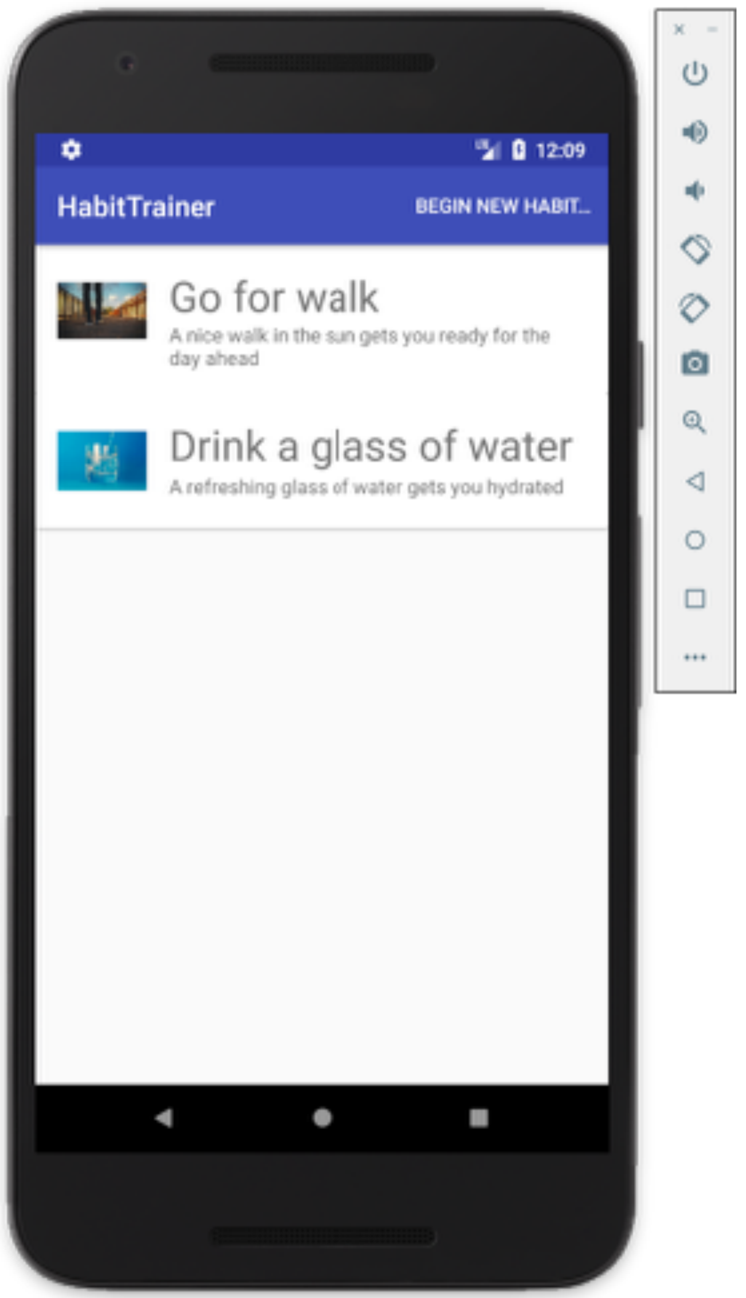
Save-Button



```
<Button
    android:id="@+id/btn_save"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="storeHabit"
    android:text="Save" />
```

CreateHabitActivity

```kotlin
class CreateHabitActivity : AppCompatActivity() {

    private val TAG = CreateHabitActivity::class.java.simpleName

    private val CHOOSE_IMAGE_REQUEST = 4711

    private var imageBitmap: Bitmap? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        ...
    }

    fun storeHabit(v: View) {
        if (et_title.text.toString().isBlank()
                || et_descr.text.toString().isBlank()) {
            Log.d(TAG, "No habit stored: title or description missing.")
            displayErrorMessage("Your habit needs an engaging title and description.")
            return
        } else if (imageBitmap == null) {
            Log.d(TAG, "No habit stored: image missing.")
            displayErrorMessage("Add a motivating picture to your habit")
            return
        }

        // store the habit in database ...
        tv_error.visibility = View.INVISIBLE
    }

    private fun displayErrorMessage(message: String) {
        tv_error.text = message
        tv_error.visibility = View.VISIBLE
    }

    fun chooseImage(v: View) {
        ...
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)

        if (requestCode == CHOOSE_IMAGE_REQUEST
                && resultCode == Activity.RESULT_OK
                && data != null
                && data.data != null) {

            Log.d(TAG, "An image was chosen by the user")

            val bitmap = tryReadBitmap(data.data)

            bitmap?.let {
                this.imageBitmap = bitmap
                iv_image.setImageBitmap(bitmap)
                Log.d(TAG, "Read image bitmap and updated image view.")
            }
        }
    }

    fun tryReadBitmap(data: Uri): Bitmap? {
        ...
    }
}
```

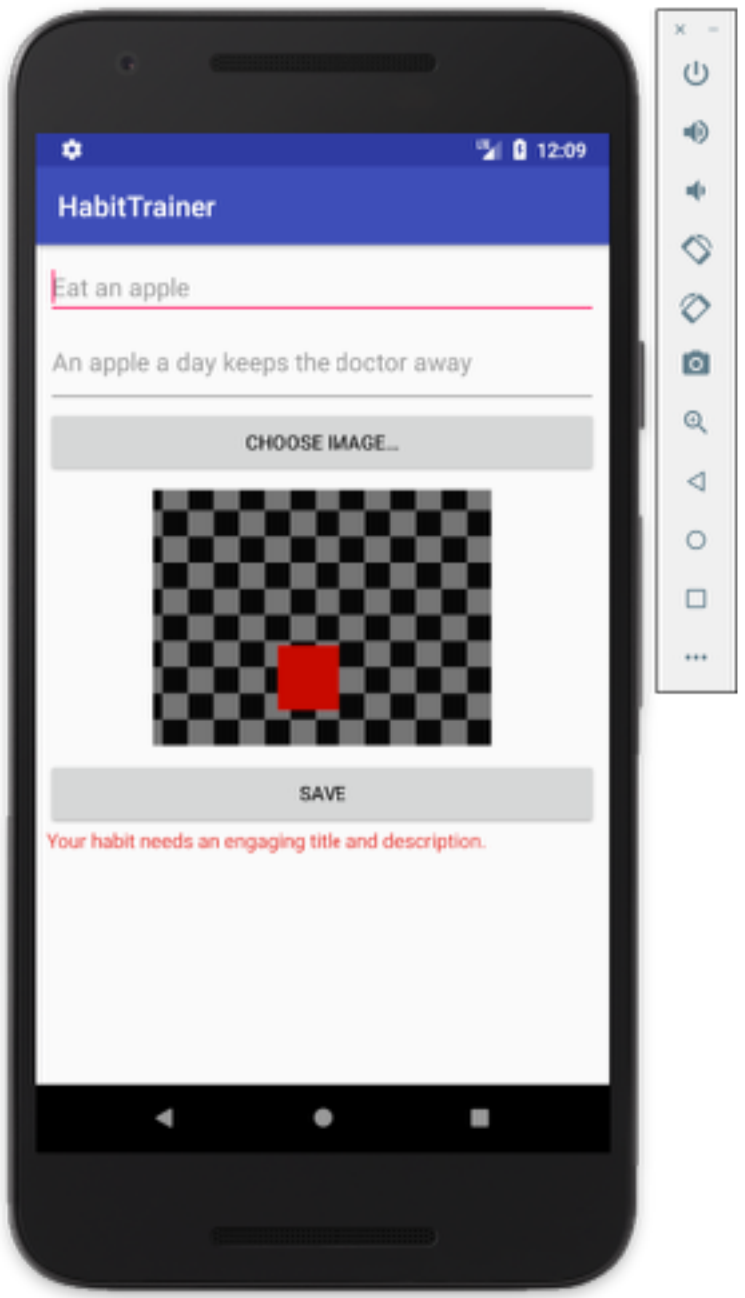


Go for walk

A nice walk in the sun gets you ready for the day ahead

Drink a glass of water

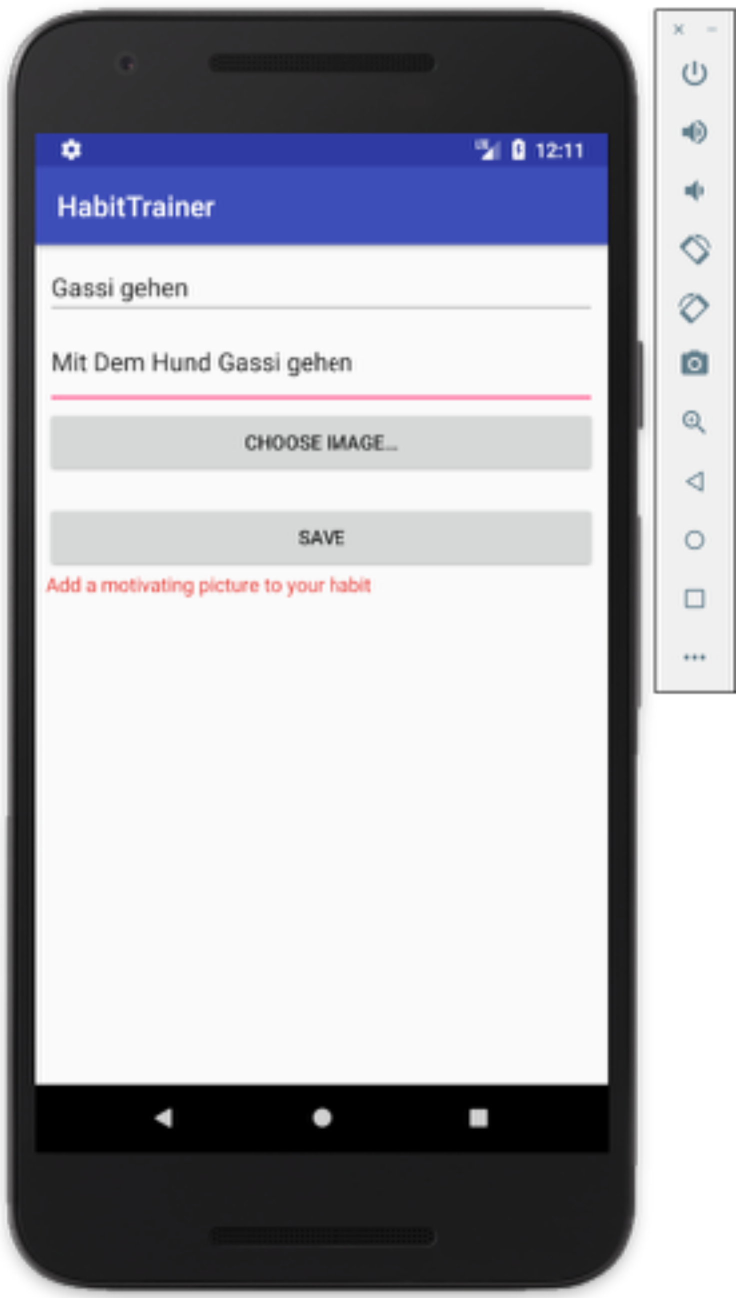
A refreshing glass of water gets you hydrated



Eat an apple

An apple a day keeps the doctor away

Your habit needs an engaging title and description.



Gassi gehen

Mit Dem Hund Gassi gehen

Add a motivating picture to your habit

Extension Functions

```
class CreateHabitActivity : AppCompatActivity() {
    ...

    fun storeHabit(v: View) {
        if (et_title.isBlank() || et_descr.isBlank()) {
            Log.d(TAG, "No habit stored: title or description missing.")
            displayErrorMessage("Your habit needs an engaging title and description.")
            return
        } else if (imageBitmap == null) {
            Log.d(TAG, "No habit stored: image missing.")
            displayErrorMessage("Add a motivating picture to your habit")
            return
        }

        // store the habit in database ...
        tv_error.visibility = View.INVISIBLE
    }

  ...
}

//    private fun EditText.isBlank(): Boolean {
//        if (this.text.toString().isBlank()) {
//            return true
//        }
//        return false
//    }

private fun EditText.isBlank() = this.text.isBlank()
```

Auch diese Schreibweise wäre möglich

SQL-Datenbank

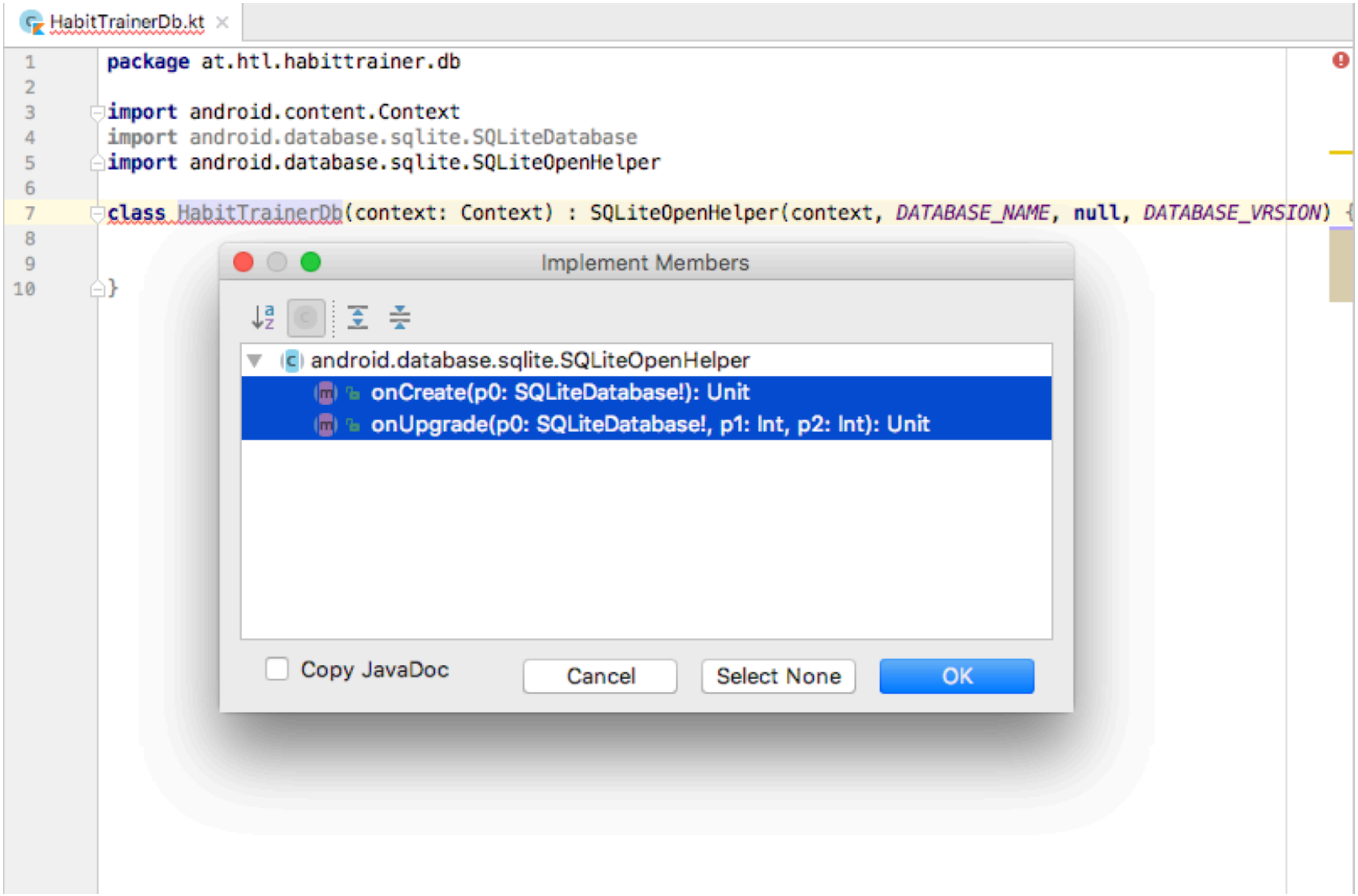
Contracts.kt

```
package at.htl.habittrainer.db

import android.provider.BaseColumns

val DATABASE_NAME = "habittrainer.db"
val DATABASE_VRSION = 10

object HabitEntry : BaseColumns {
    val TABLE_NAME = "habit"
    val _ID = "id"
    val TITLE_COL = "title"
    val DESCR_COL = "description"
    val IMAGE_COL = "image"
}
```

```
package android.provider;

public interface BaseColumns {
    String _COUNT = "_count";
    String _ID = "_id";
}
```

Diese Spalte sollte eigentlich vom Interface BaseColumns zur Verfügung gestellt werden, doch Kotlin kennt das Konzept der static fields nicht, daher hat man keinen Zugriff auf _ID des Interfaces

HabitTrainerDb.kt



HabitTrainerDb.kt

```kotlin
package at.htl.habittrainer.db

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class HabitTrainerDb(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VRSION) {


}
```

Implement Members

```
android.database.sqlite.SQLiteOpenHelper
    onCreate(p0: SQLiteDatabase!): Unit
    onUpgrade(p0: SQLiteDatabase!, p1: Int, p2: Int): Unit
```

HabitTrainerDb.kt

```kotlin
package at.htl.habittrainer.db

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import android.provider.BaseColumns

class HabitTrainerDb(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VRSION) {

    private val SQL_CREATE_ENTRIES = "CREATE TABLE ${HabitEntry.TABLE_NAME} (" +
            "${HabitEntry._ID} INTEGER PRIMARY KEY, “ +
            "${HabitEntry.TITLE_COL} TEXT, " +
            "${HabitEntry.DESCR_COL} TEXT, " +
            "${HabitEntry.IMAGE_COL} BLOB " +
            ")"

    private val SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS ${HabitEntry.TABLE_NAME}"

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(SQL_CREATE_ENTRIES)
    }

    override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
        db.execSQL(SQL_DELETE_ENTRIES)
        onCreate(db)
    }
}
```

HabitDbTable.kt

```kotlin
package at.htl.habittrainer.db

import android.content.ContentValues
import android.content.ContentValues.TAG
import android.content.Context
import android.graphics.Bitmap
import android.util.Log
import at.htl.habittrainer.Habit
import java.io.ByteArrayOutputStream

class HabitDbTable(context: Context) {

    private val dbHelper = HabitTrainerDb(context)

    fun store(habit: Habit): Long {
        val db = dbHelper.writableDatabase

        val values = ContentValues()
        values.put(HabitEntry.TITLE_COL, habit.title)
        values.put(HabitEntry.DESCR_COL, habit.description)
        values.put(HabitEntry.IMAGE_COL, toByteArray(habit.image))

        val id = db.insert(HabitEntry.TABLE_NAME, null, values)

        db.close()

        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    private fun toByteArray(bitmap: Bitmap): ByteArray {
        val stream = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.PNG, 0, stream)
        return stream.toByteArray()
    }
}
```

ContentValues sind ein key/value-store in dem die Spaltenwerte übergeben werden

db.insert() speichert den Datensatz in der DB-Tabelle

Jetzt muss noch in der data class der korrekte Datentyp für das image eingegeben werden

Habit.kt

```kotlin
package at.htl.habittrainer

import android.graphics.Bitmap

data class Habit(val title: String, val description: String, val image: Bitmap)

//fun getSampleHabits(): List<Habit> {
//    return listOf(
//            Habit("Go for walk",
//                    "A nice walk in the sun gets you ready for the day
ahead",
//                    R.drawable.walk),
//
//            Habit("Drink a glass of water",
//                    "A refreshing glass of water gets you hydrated",
//                    R.drawable.water)
//    )
//}
```

unsere Dateninitialisierung paßt jetzt auch nicht mehr

MainActivity.kt

```
class MainActivity : AppCompatActivity() {

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

        // Adapter -> defines data
        // RecyclerView -> implement 3 methods
        rv.setHasFixedSize(true)

        rv.layoutManager = LinearLayoutManager(this)
//          rv.adapter = HabitsAdapter(getSampleHabits())
    }

    ...
}
```

Auch unser Adapter paßt nicht mehr

Datenbank-
Transaktionen

HabitDbTable.kt

```
class HabitDbTable(context: Context) {

    private val dbHelper = HabitTrainerDb(context)

    fun store(habit: Habit): Long {
        val db = dbHelper.writableDatabase

        val values = ContentValues()
        values.put(HabitEntry.TITLE_COL, habit.title)
        values.put(HabitEntry.DESCR_COL, habit.description)
        values.put(HabitEntry.IMAGE_COL, toByteArray(habit.image))

        db.beginTransaction()
        val id = try {
            val returnValue = db.insert(HabitEntry.TABLE_NAME, null, valu
            db.setTransactionSuccessful()

            returnValue
        } finally {
            db.endTransaction()
        }
        db.close()

        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    private fun toByteArray(bitmap: Bitmap): ByteArray {
        val stream = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.PNG, 0, str
        return stream.toByteArray()
    }
}
```

Der try-Block wird als expression verwendet

Dies ist eigentlich umfangreich. Wir werden versuchen dies mit Kotlin einfacher zu gestalten, mit weniger Boilerplate-code

Transaktion als extension function

```kotlin
class HabitDbTable(context: Context) {

    private val dbHelper = HabitTrainerDb(context

    fun store(habit: Habit): Long {
        val db = dbHelper.writableDatabase

        val values = ContentValues()
        values.put(HabitEntry.TITLE_COL, habit.title)
        values.put(HabitEntry.DESCR_COL, habit.description)
        values.put(HabitEntry.IMAGE_COL, toByteArray(habit.image))

        db.transaction {
            db.insert(HabitEntry.TABLE_NAME, null, values)
        }

        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    private fun toByteArray(bitmap: Bitmap): ByteArray {
        val stream = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.PNG, 0, stream)
        return stream.toByteArray()
    }
}

private fun SQLiteDatabase.transaction(function: () -> Unit) {
    beginTransaction()
    try {
        function()
        setTransactionSuccessful()
    } finally {
        endTransaction()
    }
    close()
}
```

2 Nun kann die Datenbankoperation gekapselt werden

3 Ein Problem bleibt noch. Es wird keine id zurückgegeben

1 Ein extension function wird erstellt. Unit bedeutet, dass es KEINEN Rückgabewert gibt

db als Parameter

```
class HabitDbTable(context: Context) {

    private val dbHelper = HabitTrainerDb(context)

    fun store(habit: Habit): Long {
        val db = dbHelper.writableDatabase

        val values = ContentValues()
        values.put(HabitEntry.TITLE_COL, habit.title)
        values.put(HabitEntry.DESCR_COL, habit.description)
        values.put(HabitEntry.IMAGE_COL, toByteArray(habit.image))

        db.transaction {
            it.insert(HabitEntry.TABLE_NAME, null, values)
        }


        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    private fun toByteArray(bitmap: Bitmap): ByteArray {
        val stream = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.PNG, 0, stream)
        return stream.toByteArray()
    }
}

private fun SQLiteDatabase.transaction(function: (SQLiteDatabase) -> Unit) {
    beginTransaction()
    try {
        function(this)
        setTransactionSuccessful()
    } finally {
        endTransaction()
    }
    close()
}
```

Somit wird immer die korrekte db verwendet.

extension function in der extension function

```
class HabitDbTable(context: Context) {

    ...

        db.transaction {
            insert(HabitEntry.TABLE_NAME, null, values)
        }

        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    private fun toByteArray(bitmap: Bitmap): ByteArray {
        val stream = ByteArrayOutputStream()
        bitmap.compress(Bitmap.CompressFormat.PNG, 0, stream)
        return stream.toByteArray()
    }
}

private fun SQLiteDatabase.transaction(function: SQLiteDatabase.() -> Unit) {
    beginTransaction()
    try {
        function()
        setTransactionSuccessful()
    } finally {
        endTransaction()
    }
    close()
}
```

Das it. kann nun weggelassen werden

Die als Parameter übergebene db wird nun als extension function von SQLiteDatabase deklariert

Da die db eine extension ist, kann nun die übergebene Funktion ganz einfach aufgerufen werden

inline und return-Value

```
class HabitDbTable(context: Context) {

    ...

        val id = db.transaction {
            insert(HabitEntry.TABLE_NAME, null, values)
        }


        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }

    ...
}

private inline fun <T> SQLiteDatabase.transaction(function: SQLiteDatabase.() -> T): T {
    beginTransaction()
    val result = try {
        val returnValue = function()
        setTransactionSuccessful()

        returnValue
    } finally {
        endTransaction()
    }
    close()

    return result
}
```

Die inline Funktion sorgt dafür, dass im Bytecode die extension function beim Aufruf den transaction-Block ersetzt. Der Code sieht also wieder genau wie am Anfang aus. Der Code ist modularisiert und trotzdem performant

Der Rückgabewert ist generisch. Beim Aufruf der Transaktion ist die id automatisch vom Typ long

with-Klausel

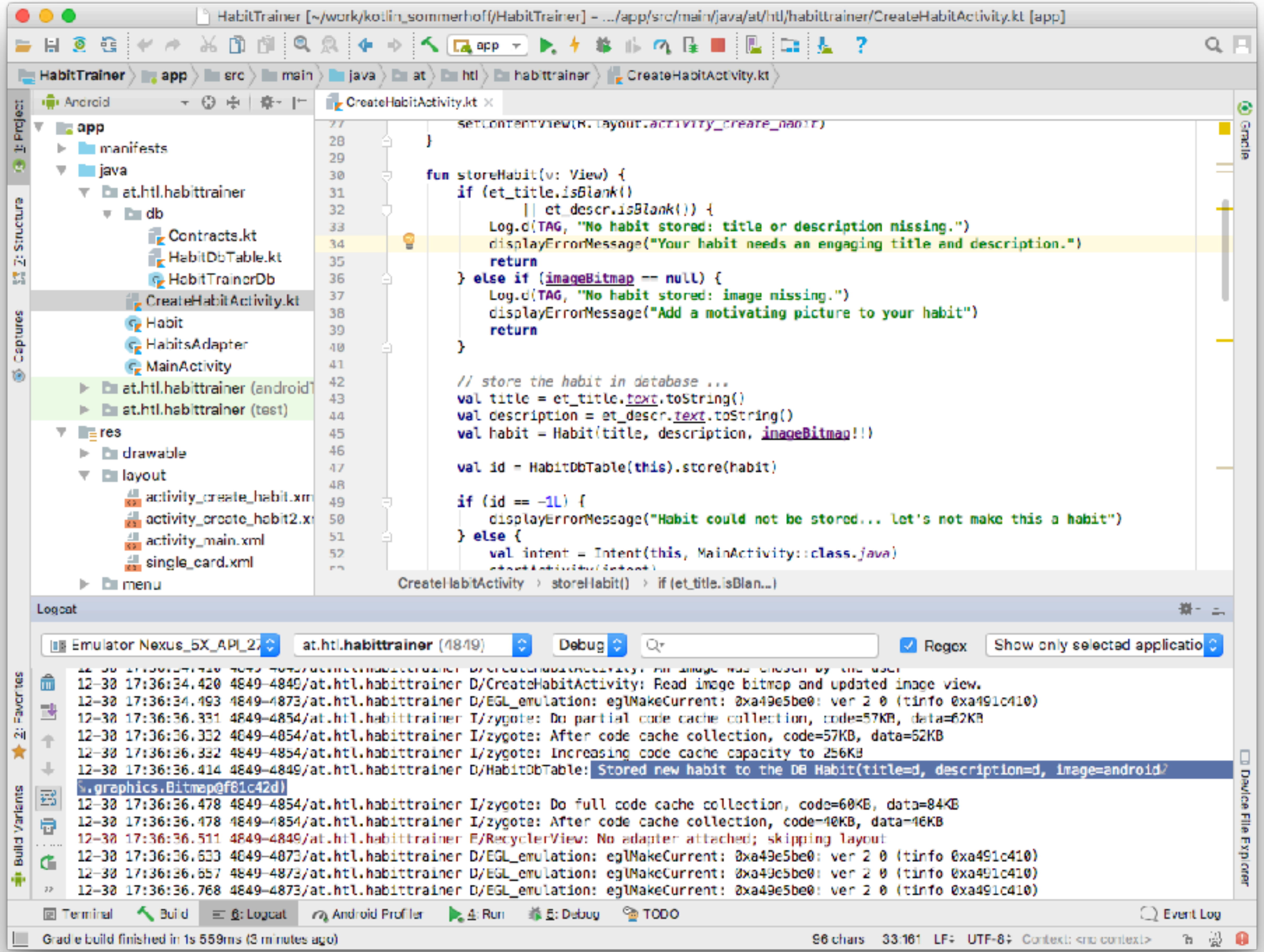
```
class HabitDbTable(context: Context) {

    private val TAG = HabitDbTable::class.java.simpleName

    private val dbHelper = HabitTrainerDb(context)

    fun store(habit: Habit): Long {
        val db = dbHelper.writableDatabase

        val values = ContentValues()
        with(values) {
            put(HabitEntry.TITLE_COL, habit.title)
            put(HabitEntry.DESCR_COL, habit.description)
            put(HabitEntry.IMAGE_COL, toByteArray(habit.image))
        }

        val id = db.transaction {
            insert(HabitEntry.TABLE_NAME, null, values)
        }

        Log.d(TAG, "Stored new habit to the DB $habit")

        return id
    }
}
```

CreateHabitActivity.kt

```kotlin
fun storeHabit(v: View) {
    if (et_title.isBlank()
            || et_descr.isBlank()) {
        Log.d(TAG, "No habit stored: title or description missing.")
        displayErrorMessage("Your habit needs an engaging title and description.")
        return
    } else if (imageBitmap == null) {
        Log.d(TAG, "No habit stored: image missing.")
        displayErrorMessage("Add a motivating picture to your habit")
        return
    }

    // store the habit in database ...
    val title = et_title.text.toString()
    val description = et_descr.text.toString()
    val habit = Habit(title, description, imageBitmap!!)

    val id = HabitDbTable(this).store(habit)

    if (id == -1L) {
        displayErrorMessage("Habit could not be stored... let's not make this a habit")
    } else {
        val intent = Intent(this, MainActivity::class.java)
        startActivity(intent)
    }
}
```

HabitsAdapter.kt

```
// Specifies the contents for the shown habit
override fun onBindViewHolder(holder: HabitViewHolder?, index: Int) {
    if (holder != null) {  // if wegen SmartCast
        val habit = habits[index]
        holder.card.tv_title.text = habit.title
        holder.card.tv_description.text = habit.description
        holder.card.iv_icon.setImageBitmap(habit.image)
    }
}
```



```kotlin
    }

    fun storeHabit(v: View) {
        if (et_title.isBlank()
                || et_descr.isBlank()) {
            Log.d(TAG, "No habit stored: title or description missing.")
            displayErrorMessage("Your habit needs an engaging title and description.")
            return
        } else if (imageBitmap == null) {
            Log.d(TAG, "No habit stored: image missing.")
            displayErrorMessage("Add a motivating picture to your habit")
            return
        }

        // store the habit in database ...
        val title = et_title.text.toString()
        val description = et_descr.text.toString()
        val habit = Habit(title, description, imageBitmap!!)

        val id = HabitDbTable(this).store(habit)

        if (id == -1L) {
            displayErrorMessage("Habit could not be stored... let's not make this a habit")
        } else {
            val intent = Intent(this, MainActivity::class.java)
```

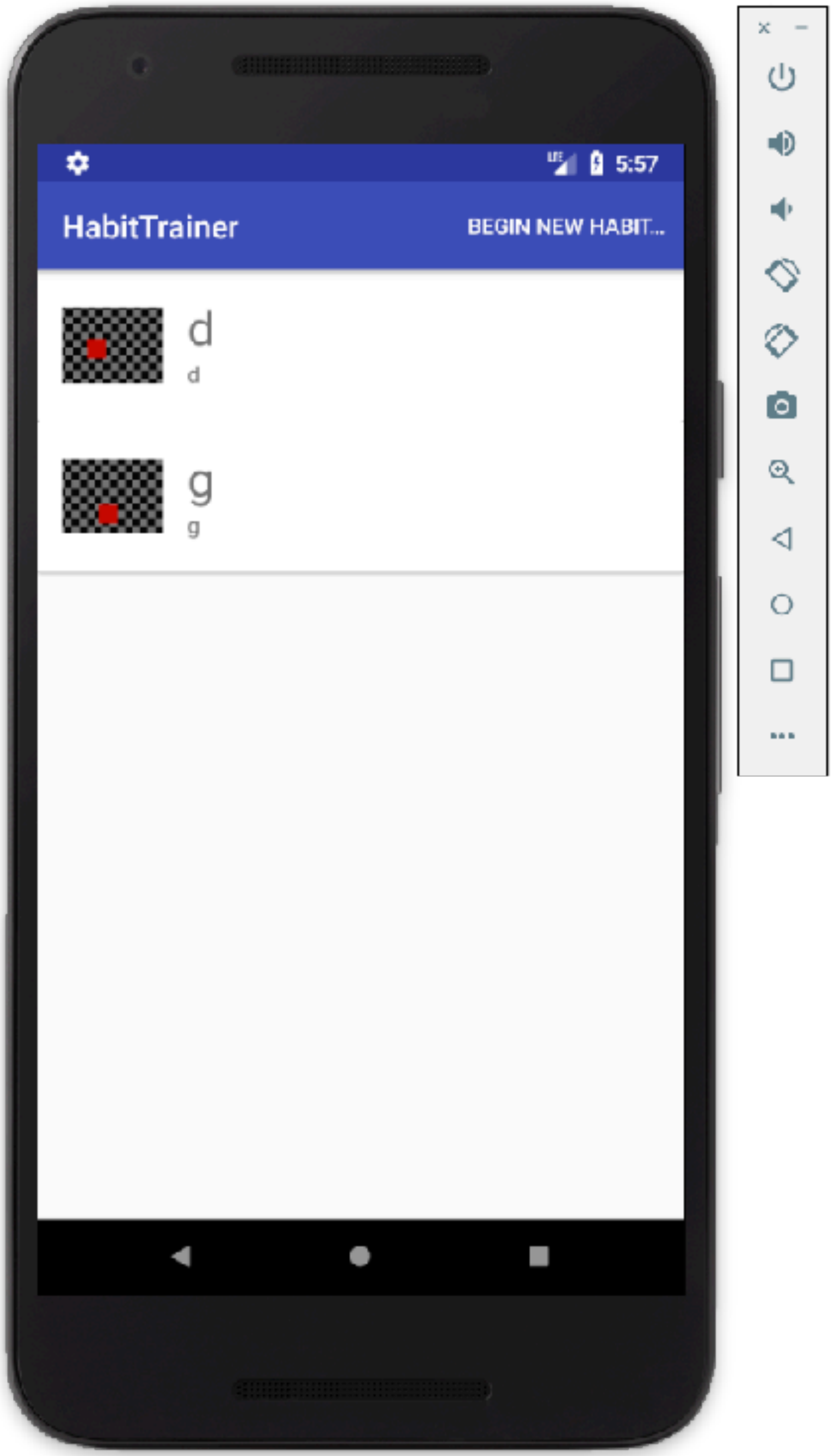
```
12-30 17:36:34.420 4849-4849/at.htl.habittrainer D/CreateHabitActivity: Read image bitmap and updated image view.
12-30 17:36:34.493 4849-4873/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0xa49e5be0: ver 2 0 (tinfo 0xa491c410)
12-30 17:36:36.331 4849-4854/at.htl.habittrainer I/zygote: Do partial code cache collection, code=57KB, data=62KB
12-30 17:36:36.332 4849-4854/at.htl.habittrainer I/zygote: After code cache collection, code=57KB, data=62KB
12-30 17:36:36.332 4849-4854/at.htl.habittrainer I/zygote: Increasing code cache capacity to 256KB
12-30 17:36:36.414 4849-4849/at.htl.habittrainer D/HabitDbTable: Stored new habit to the DB Habit(title=d, description=d, image=android.graphics.Bitmap@f61c42d)
12-30 17:36:36.478 4849-4854/at.htl.habittrainer I/zygote: Do full code cache collection, code=60KB, data=84KB
12-30 17:36:36.478 4849-4854/at.htl.habittrainer I/zygote: After code cache collection, code=40KB, data=46KB
12-30 17:36:36.511 4849-4849/at.htl.habittrainer E/RecyclerView: No adapter attached; skipping layout
12-30 17:36:36.633 4849-4873/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0xa49e5be0: ver 2 0 (tinfo 0xa491c410)
12-30 17:36:36.657 4849-4873/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0xa49e5be0: ver 2 0 (tinfo 0xa491c410)
12-30 17:36:36.768 4849-4873/at.htl.habittrainer D/EGL_emulation: eglMakeCurrent: 0xa49e5be0: ver 2 0 (tinfo 0xa491c410)
```

HabitDbTable.kt

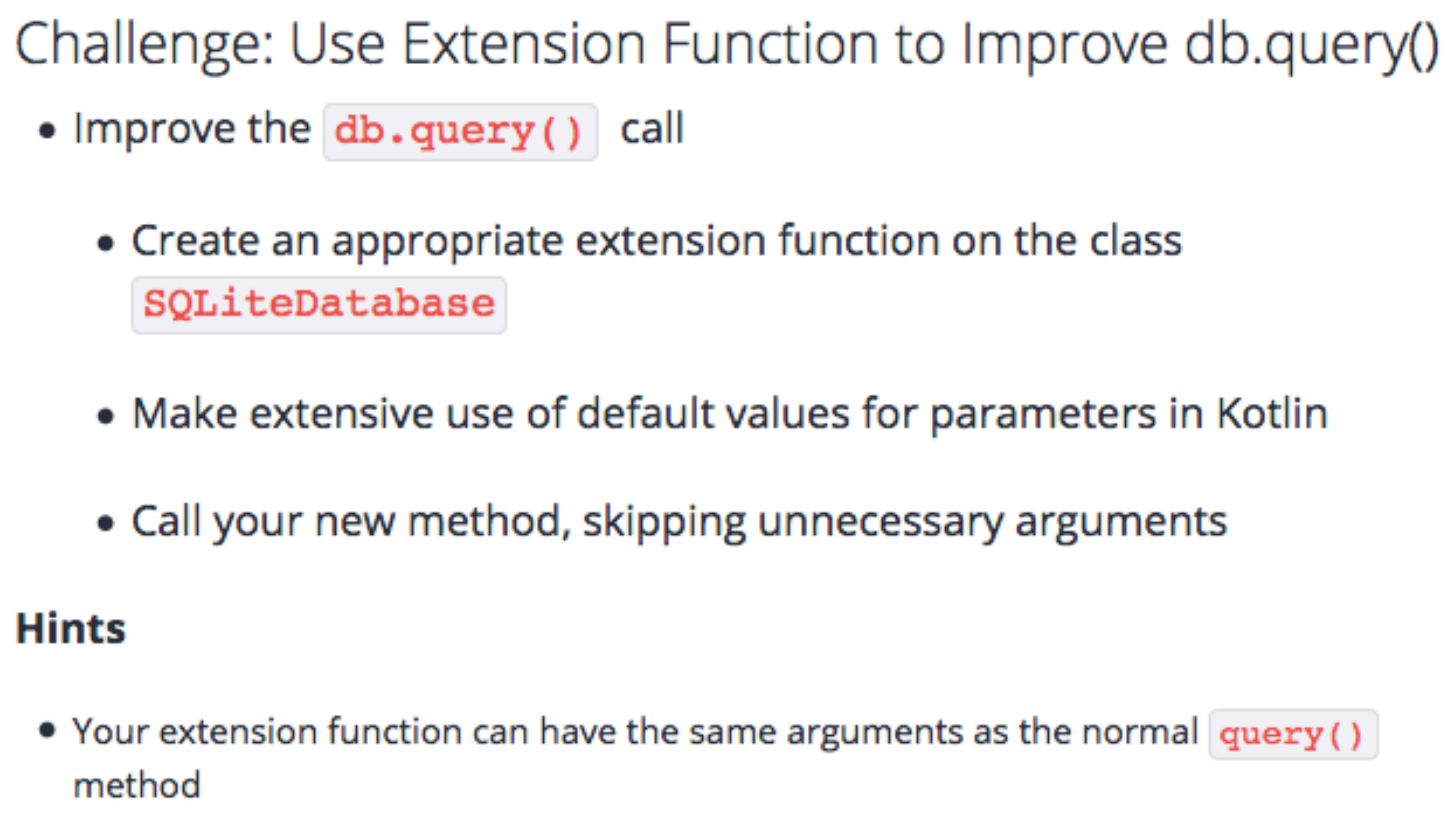
```
fun readAllHabits(): List<Habit> {

    val columns = arrayOf(HabitEntry._ID, HabitEntry.TITLE_COL,
            HabitEntry.DESCR_COL, HabitEntry.IMAGE_COL)

    val order = "${HabitEntry._ID} ASC"

    val db = dbHelper.readableDatabase

    val cursor = db.query(HabitEntry.TABLE_NAME, columns, null, null, null, null,
order)

    val habits = mutableListOf<Habit>()
    while (cursor.moveToNext()) {
        val title = cursor.getString(cursor.getColumnIndex((HabitEntry.TITLE_COL)))
        val desc = cursor.getString(cursor.getColumnIndex((HabitEntry.DESCR_COL)))
        val byteArray = cursor.getBlob(cursor.getColumnIndex((HabitEntry.IMAGE_COL)))
        val bitmap = BitmapFactory.decodeByteArray(byteArray, 0, byteArray.size)
        habits.add(Habit(title, desc, bitmap))
    }
    cursor.close()

    return habits
}
```

MainActivity.kt

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

    // Adapter -> defines data
    // RecyclerView -> implement 3 methods
    rv.setHasFixedSize(true)
    rv.layoutManager = LinearLayoutManager(this)
    rv.adapter = HabitsAdapter(HabitDbTable(this).readAllHabits())
}
```


5:57
HabitTrainer
BEGIN NEW HABIT...
d
d
g
g

Challenge: Improve SQLiteDatabase.query(...)



Challenge: Use Extension Function to Improve db.query()

- Improve the `db.query()` call
  - Create an appropriate extension function on the class `SQLiteDatabase`
  - Make extensive use of default values for parameters in Kotlin
  - Call your new method, skipping unnecessary arguments

Hints

- Your extension function can have the same arguments as the normal `query()` method

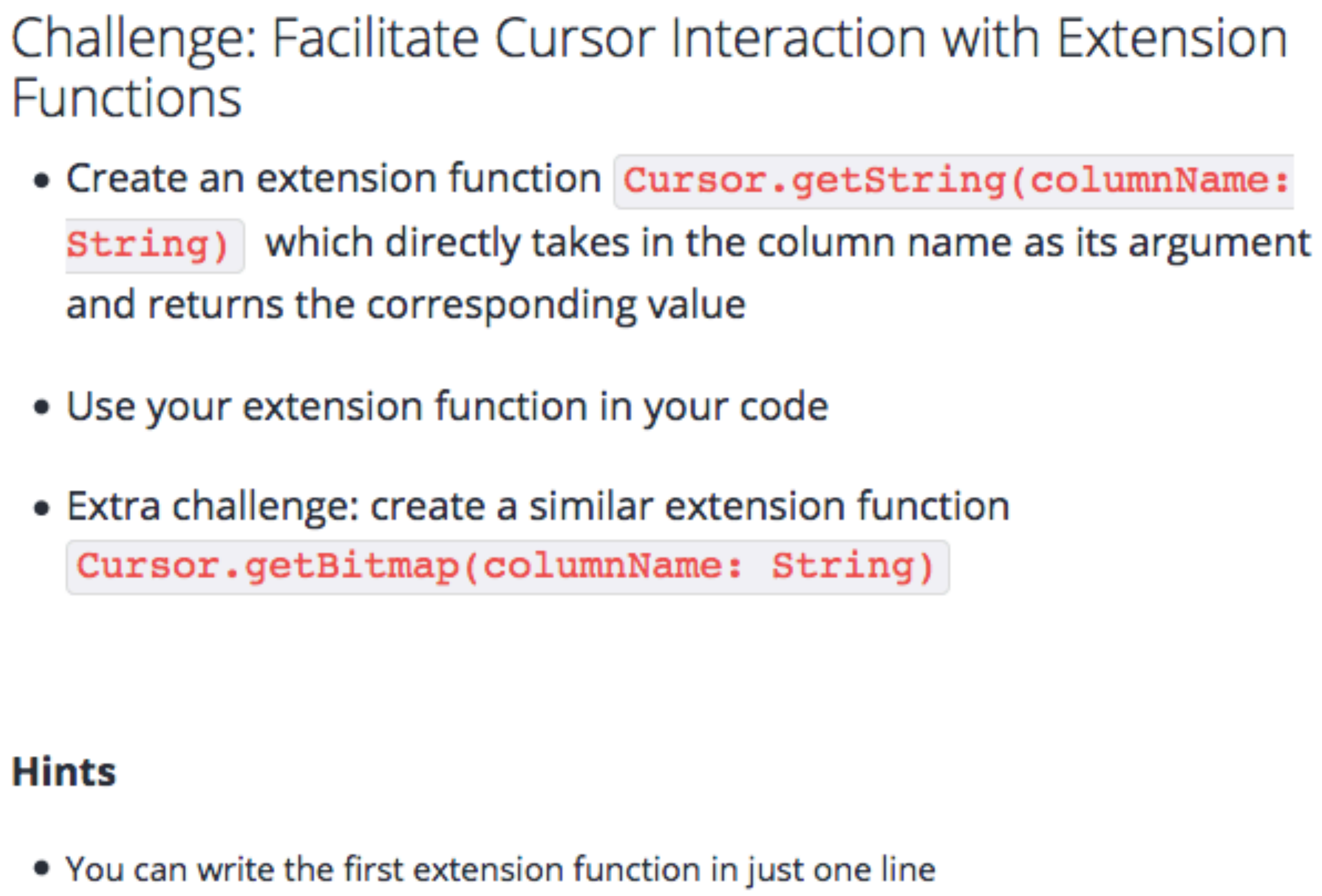
Default Values for Parameters

```
class HabitDbTable(context: Context) {

    ...

    fun readAllHabits(): List<Habit> {

        val columns = arrayOf(HabitEntry._ID, HabitEntry.TITLE_COL,
                HabitEntry.DESCR_COL, HabitEntry.IMAGE_COL)

        val order = "${HabitEntry._ID} ASC"

        val db = dbHelper.readableDatabase

        val cursor = db.doQuery(HabitEntry.TABLE_NAME, columns, orderBy = order)

        val habits = mutableListOf<Habit>()
        while (cursor.moveToNext()) {
            val title = cursor.getString(cursor.getColumnIndex((HabitEntry.TITLE_COL)))
            val desc = cursor.getString(cursor.getColumnIndex((HabitEntry.DESCR_COL)))
            val byteArray = cursor.getBlob(cursor.getColumnIndex((HabitEntry.IMAGE_COL)))
            val bitmap = BitmapFactory.decodeByteArray(byteArray, 0, byteArray.size)
            habits.add(Habit(title, desc, bitmap))
        }
        cursor.close()

        return habits
    }
}

private fun toByteArray(bitmap: Bitmap): ByteArray {
    ...
}

private fun SQLiteDatabase.doQuery(table: String, columns: Array<String>, selection: String? = null,
                                   selectionArgs: Array<String>? = null, groupBy: String? = null,
                                   having: String? = null, orderBy: String? = null): Cursor {
    return query(table, columns, selection, selectionArgs, groupBy, having, orderBy)
}
```

Wir verwenden den Namen doQuery(), da query() mit gleichen PArametern bereits existiert

Durch die Verwendung der default parameters, können die nicht gebrauchten Parameter weggelassen werden

Challenge: Improve Cursor.getString(…)



Challenge: Facilitate Cursor Interaction with Extension Functions

- Create an extension function `Cursor.getString(columnName: String)` which directly takes in the column name as its argument and returns the corresponding value
- Use your extension function in your code
- Extra challenge: create a similar extension function `Cursor.getBitmap(columnName: String)`

Hints

- You can write the first extension function in just one line

Dieser Ausdruck soll vereinfacht werden:

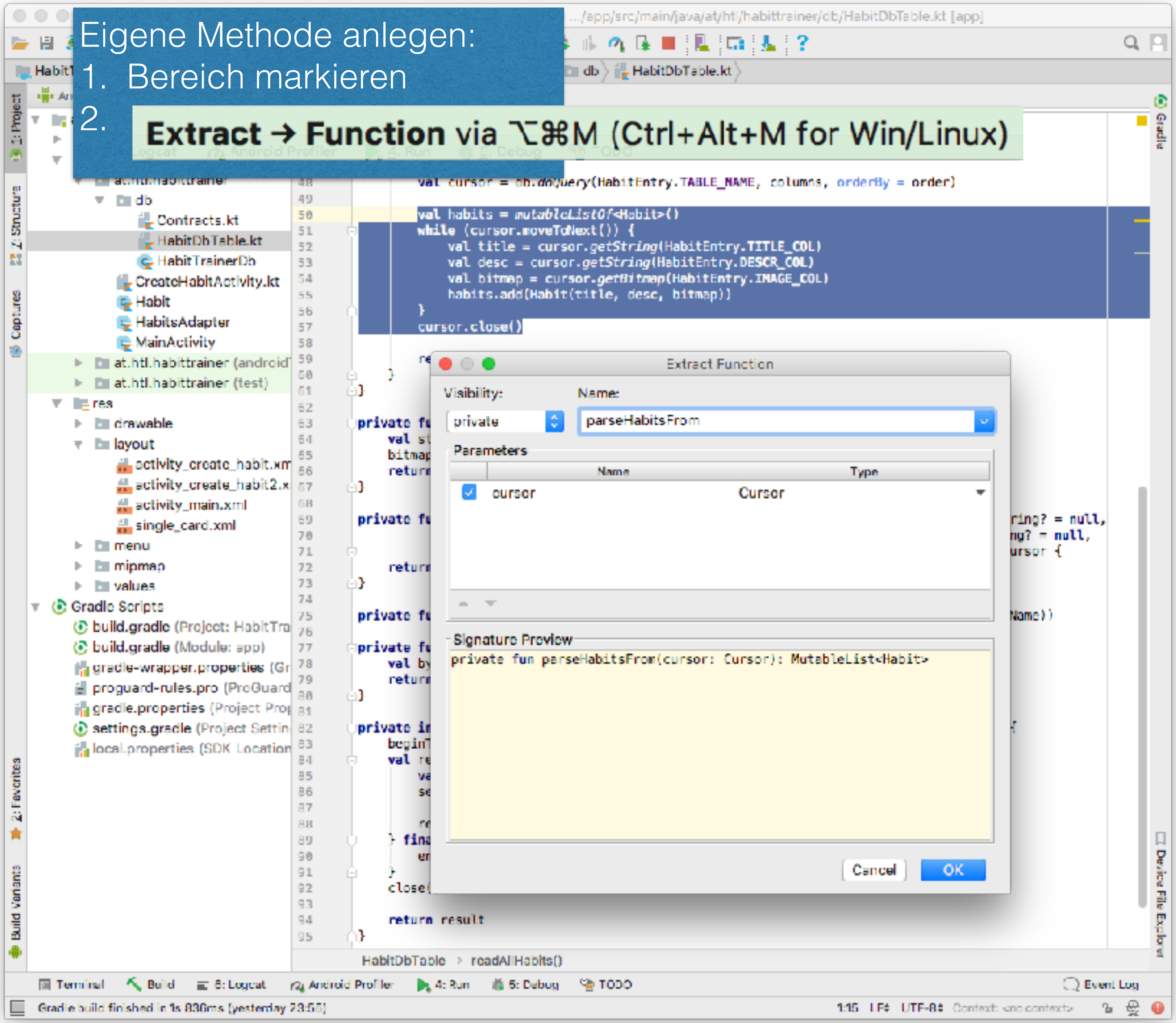
```
val title = cursor.getString(cursor.getColumnIndex((HabitEntry.TITLE_COL)))
```

```
class HabitDbTable(context: Context) {

    ...

    fun readAllHabits(): List<Habit> {

        val columns = arrayOf(HabitEntry._ID, HabitEntry.TITLE_COL,
                HabitEntry.DESCR_COL, HabitEntry.IMAGE_COL)

        val order = "${HabitEntry._ID} ASC"

        val db = dbHelper.readableDatabase

        val cursor = db.doQuery(HabitEntry.TABLE_NAME, columns, orderBy = order)

        val habits = mutableListOf<Habit>()
        while (cursor.moveToNext()) {
            val title = cursor.getString(HabitEntry.TITLE_COL)
            val desc = cursor.getString(HabitEntry.DESCR_COL)
            val bitmap = cursor.getBitmap(HabitEntry.IMAGE_COL)
            habits.add(Habit(title, desc, bitmap))
        }
        cursor.close()

        return habits
    }
}

private fun toByteArray(bitmap: Bitmap): ByteArray { ... }

private fun SQLiteDatabase.doQuery(table: String, columns: Array<String>, selection: String? = null,
                                    selectionArgs: Array<String>? = null, groupBy: String? = null,
                                    having: String? = null, orderBy: String? = null): Cursor { ... }

private fun Cursor.getString(columnName: String) = this.getString(getColumnIndex(columnName))

private fun Cursor.getBitmap(columnName: String): Bitmap {
    val bytes = getBlob(getColumnIndex(columnName))
    return BitmapFactory.decodeByteArray(bytes, 0, bytes.size)
}

private inline fun <T> SQLiteDatabase.transaction(function: SQLiteDatabase.() -> T): T { ... }
```

Auch hier vereinfachen extension functions die Aufrufe sehr

Eigene Methode anlegen:

1. Bereich markieren
2. **Extract → Function via ⌥⌘M (Ctrl+Alt+M for Win/Linux)**



```
val cursor = db.doQuery(HabitEntry.TABLE_NAME, columns, orderBy = order)

val habits = mutableListOf<Habit>()
while (cursor.moveToNext()) {
    val title = cursor.getString(HabitEntry.TITLE_COL)
    val desc = cursor.getString(HabitEntry.DESCR_COL)
    val bitmap = cursor.getBitmap(HabitEntry.IMAGE_COL)
    habits.add(Habit(title, desc, bitmap))
}
cursor.close()
```

Extract Function

Visibility: private

Name: parseHabitsFrom

Parameters

	Name	Type
✓	cursor	Cursor

Signature Preview

```
private fun parseHabitsFrom(cursor: Cursor): MutableList<Habit>
```

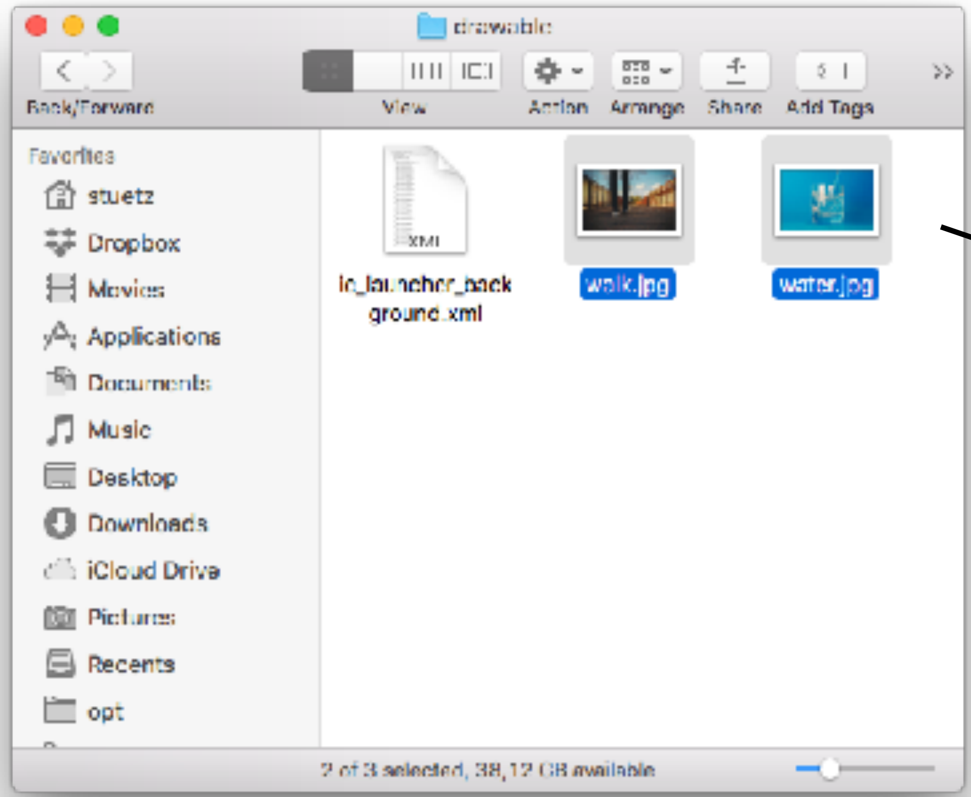
Cancel
OK
.../app/src/main/java/at/htl/habittrainer/db/HabitDbTable.kt [app]
HabitDbTable > readAllHabits()
Gradle build finished in 1s 830ms (yesterday 23:55)

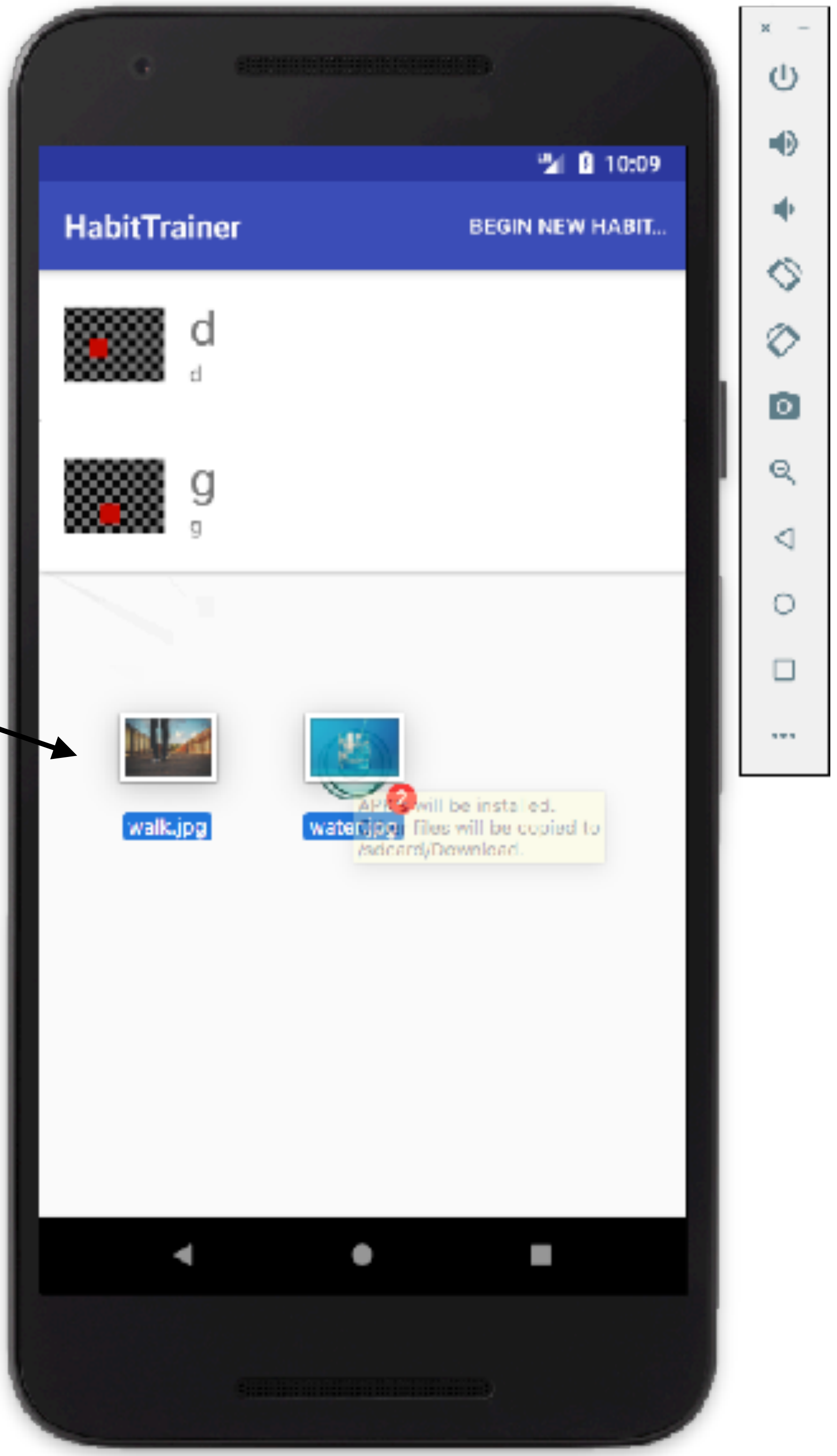
Methode extrahieren

```
fun readAllHabits(): List<Habit> {

    val columns = arrayOf(HabitEntry._ID, HabitEntry.TITLE_COL,
            HabitEntry.DESCR_COL, HabitEntry.IMAGE_COL)

    val order = "${HabitEntry._ID} ASC"

    val db = dbHelper.readableDatabase

    val cursor = db.doQuery(HabitEntry.TABLE_NAME, columns, orderBy = order)

    val habits = parseHabitsFrom(cursor)

    return habits
  }

  private fun parseHabitsFrom(cursor: Cursor): MutableList<Habit> {
    val habits = mutableListOf<Habit>()
    while (cursor.moveToNext()) {
        val title = cursor.getString(HabitEntry.TITLE_COL)
        val desc = cursor.getString(HabitEntry.DESCR_COL)
        val bitmap = cursor.getBitmap(HabitEntry.IMAGE_COL)
        habits.add(Habit(title, desc, bitmap))
    }
    cursor.close()
    return habits
  }
}
```

Das Ergebnis ist ein durchaus lesbarer Code

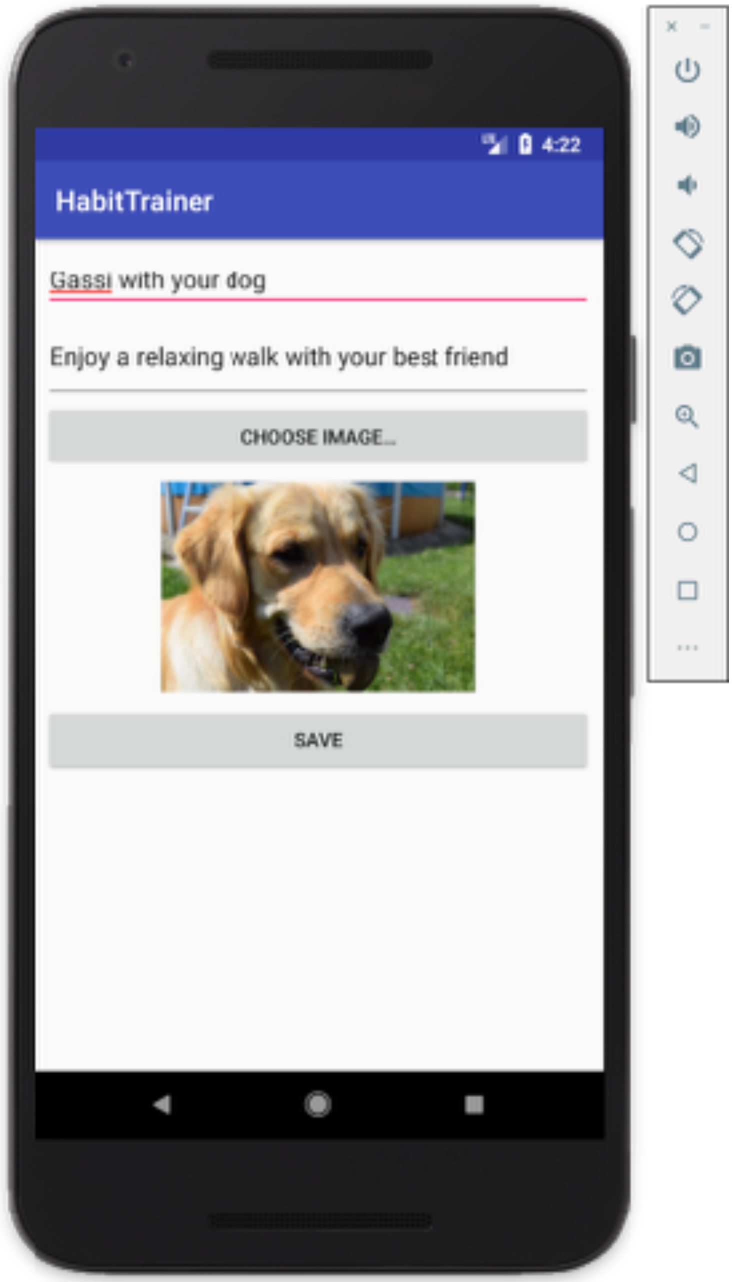
Kopieren von Files auf Device




10:09
HabitTrainer
BEGIN NEW HABIT...
d
d
g
g
walk.jpg
water.jpg
APKs will be installed. Files will be copied to /sdcard/Download.

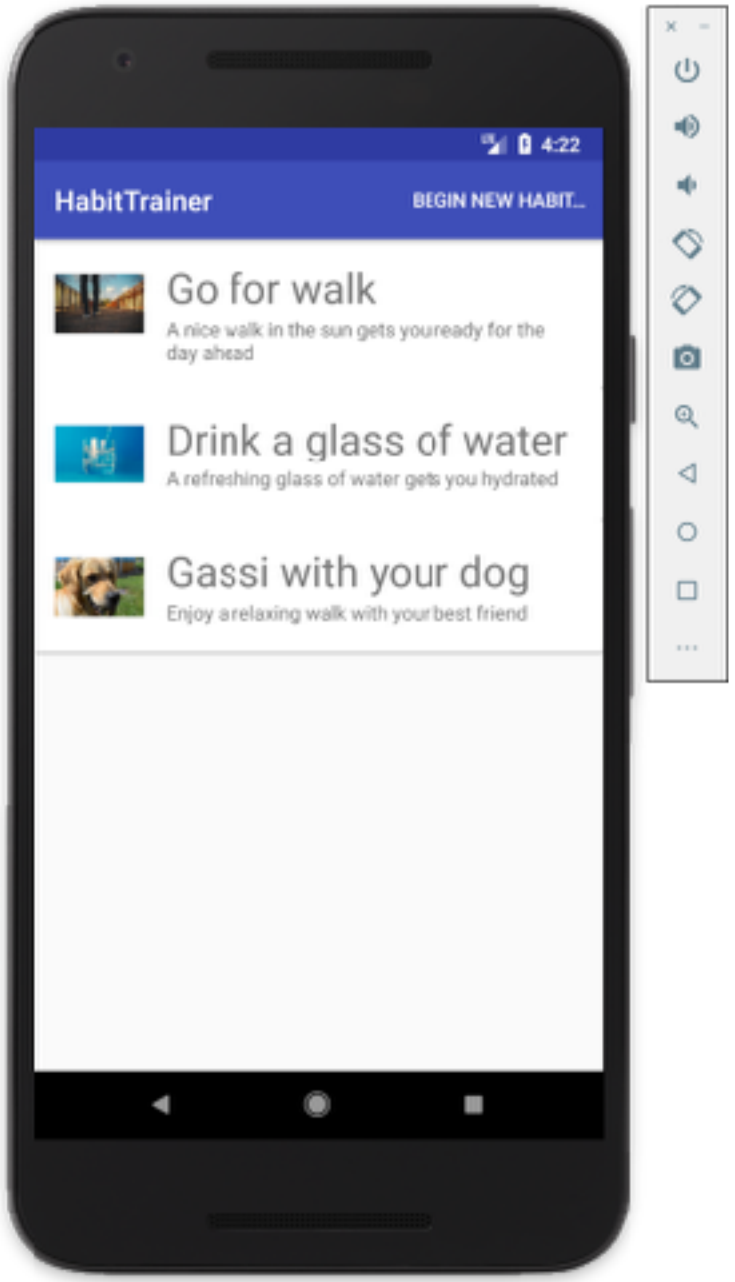
Durch Drag'n Drop kann man Files direkt in den Download-Ordner kopieren

Great - you did it



Gassi with your dog

Enjoy a relaxing walk with your best friend



Go for walk
A nice walk in the sun gets youready for the day ahead

Drink a glass of water
A refreshing glass of water gets you hydrated

Gassi with your dog
Enjoy arelaxing walk with your best friend

- https://antonioleiva.com/kotlin-awesome-tricks-for-android/
- https://github.com/petersommerhoff/kotlin-android

Kotlin

Noch
Fragen?