



Kotlin

Android



<https://kotlinlang.org/>



Create Android Project

Application name**Company domain****Project location****Package name**

- Include C++ support
- Include Kotlin support



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)

Cancel

Previous

Next

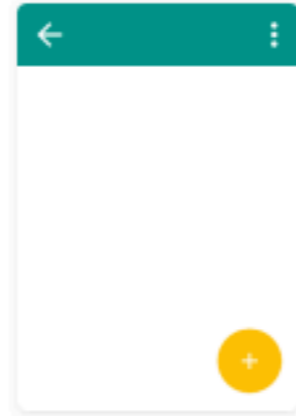
Finish



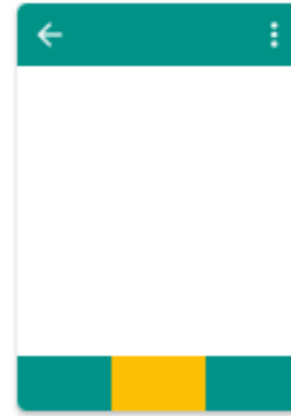
Add an Activity to Mobile



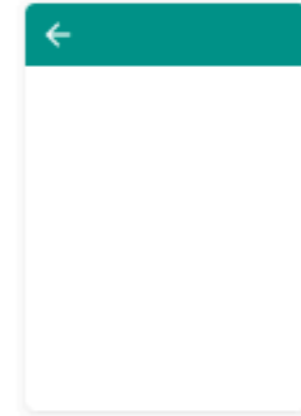
Add No Activity



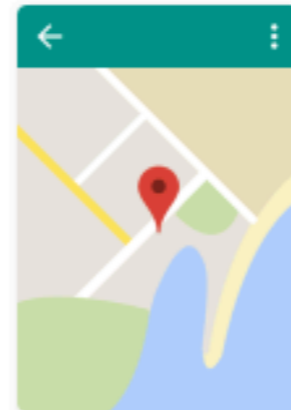
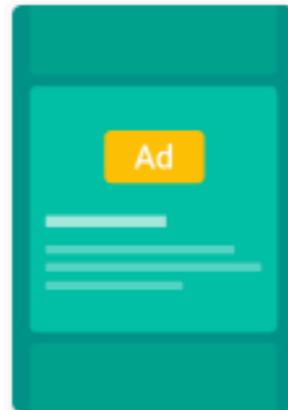
Basic Activity



Bottom Navigation Activity



Empty Activity



Cancel

Previous

Next

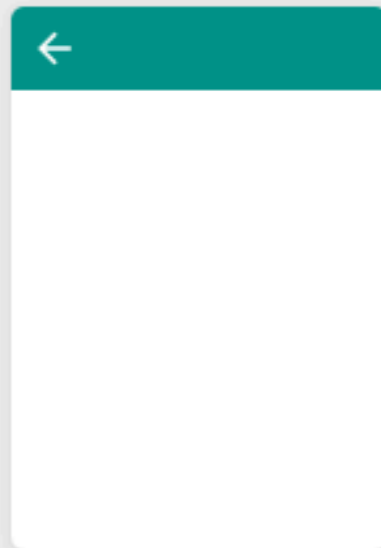
Finish



Configure Activity



Creates a new empty activity



Activity Name

MainActivity

Generate Layout File

Layout Name

activity_main

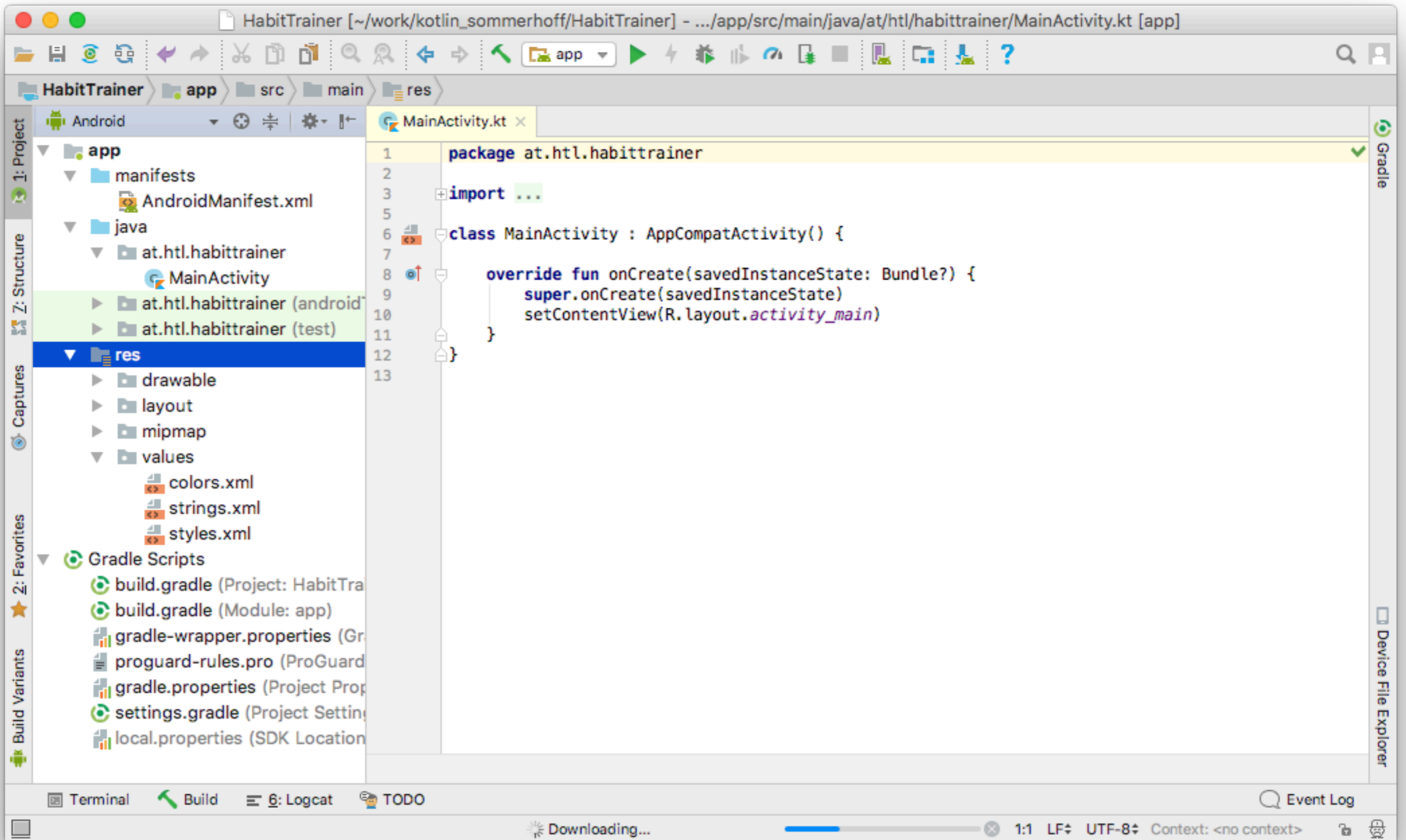
Backwards Compatibility (AppCompat)

Cancel

Previous

Next

Finish

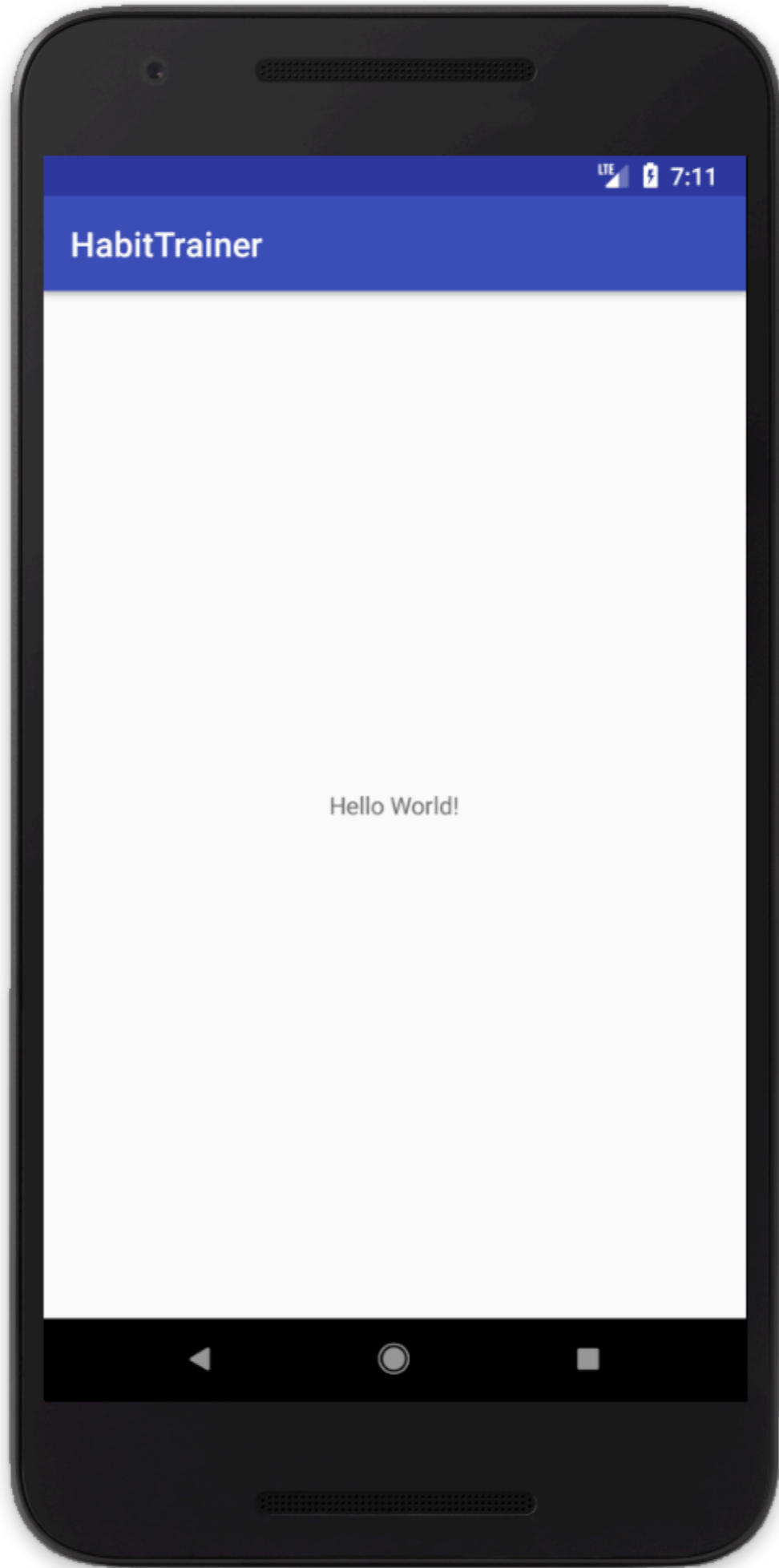


The screenshot shows an IDE window titled "HabitTrainer [~/work/kotlin_sommerhoff/HabitTrainer] - app". The main editor displays the `build.gradle` file for the `app` module. The code is as follows:

```
1  apply plugin: 'com.android.application'
2
3  apply plugin: 'kotlin-android'
4
5  apply plugin: 'kotlin-android-extensions'
6
7  android {
8      compileSdkVersion 27
9      defaultConfig {
10         applicationId "at.htl.habittrainer"
11         minSdkVersion 16
12         targetSdkVersion 27
13         versionCode 1
14         versionName "1.0"
15         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
16     }
17     buildTypes {
18         release {
19             minifyEnabled false
20             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
21         }
22     }
23 }
24
25 dependencies {
26     implementation fileTree(dir: 'libs', include: ['*.jar'])
27     implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
28     implementation 'com.android.support:appcompat-v7:27.0.2'
29     implementation 'com.android.support.constraint:constraint-layout:1.0.2'
30     testImplementation 'junit:junit:4.12'
```

The `minSdkVersion 16` line is highlighted in yellow. The IDE interface includes a Project view on the left showing the `app` module structure, a Gradle view on the right, and a bottom toolbar with options like Terminal, Build, Logcat, Android Profiler, Run, and TODO. A status bar at the bottom indicates "Gradle build finished in 26s 881ms (3 minutes ago)" and the system time is 11:25.

ev.




```
<?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" />

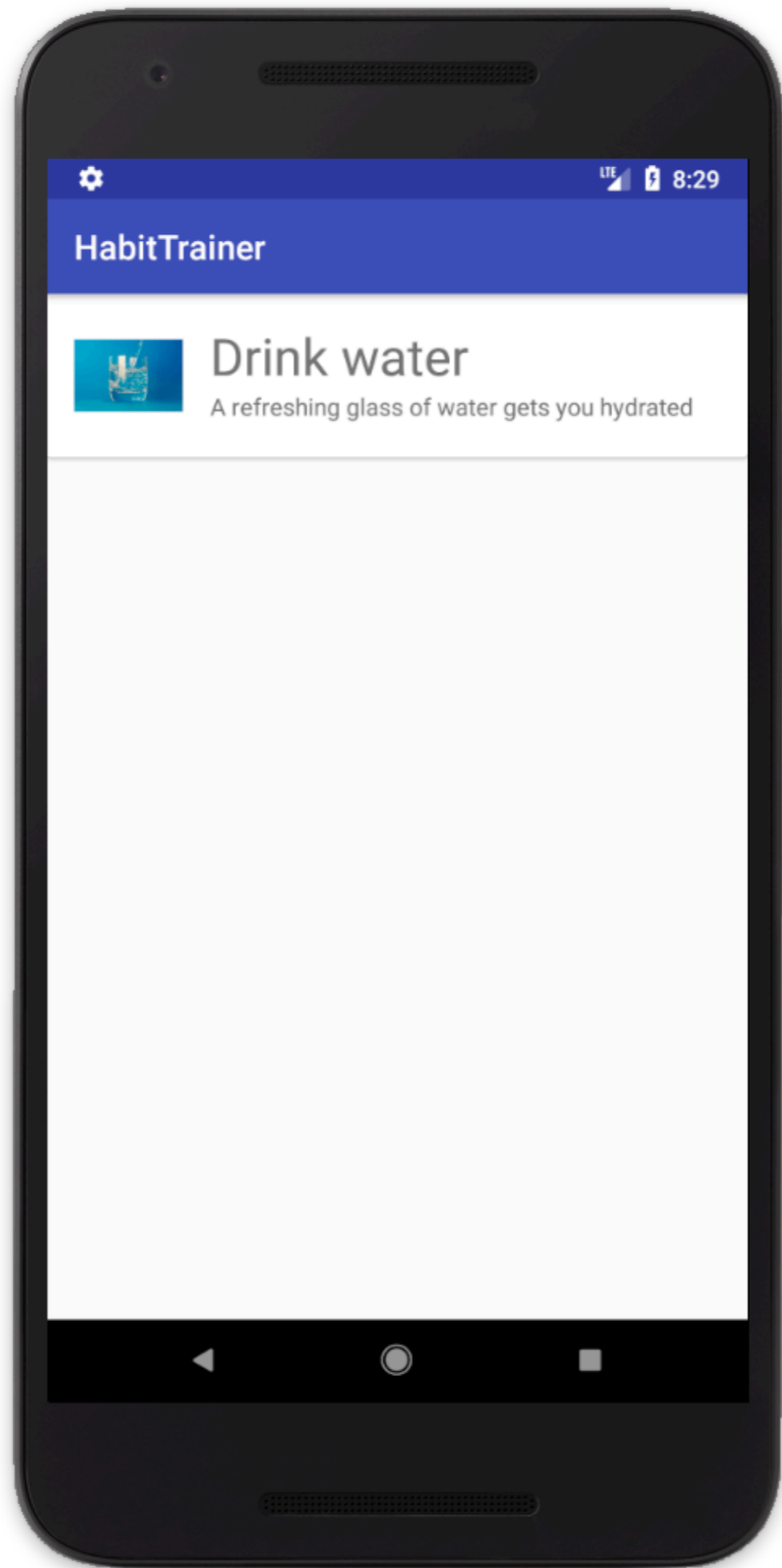
            <TextView
                android:id="@+id/tv_description"
                android:text="A refreshing glass of water gets you hydrated"
                android:layout_toRightOf="@+id/iv_icon"
                android:layout_below="@+id/tv_title"
                android:layout_width="wrap_content"
   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

        </RelativeLayout>

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

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

activity_main.xml



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"
```

```
}
```

Hier wird sichergestellt,
dass das Objekt != null
ist

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

Zugriff auf Views - the Kotlin

way

1

Zeile löschen

```
<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" />
```

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

3

```
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)
```

2

```
        ? trainer.R.id.tv_description? (multiple choices...) \n↵
```

```
        tv_description.text = "A refreshing glass of water gets you hydrated"
```

```
    }
```

```
}
```

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

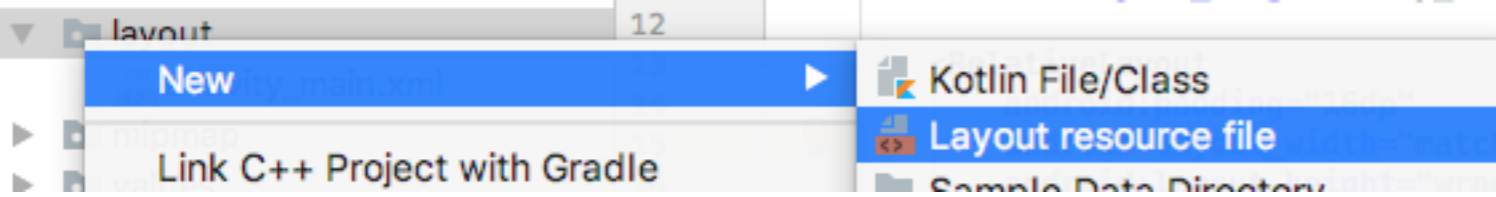
```
<RelativeLayout
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <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 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>
```

? <http://schemas.android.com/apk/res/android?> ↵

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]

activity_main.xml

Palette

- Common
 - Ab TextView
 - Button
- Text
 - ImageView
- Buttons
 - RecyclerView
- Widgets
- Layouts
- Containers
- Google
- Legacy

Attributes

ID rv

layout_width match_parent

layout_height match_parent

RecyclerView

- scrollbars
- listitem
- background
- clipToPadding
- clipChildren

Favorite Attributes

- visibility none

View all attributes

Component Tree

- ConstraintLayout
 - rv (RecyclerView)

Design Text

Terminal Build Logcat Android Profiler Run TODO

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

Context: <no context>

activity_main.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

```
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

```
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

```
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)
```

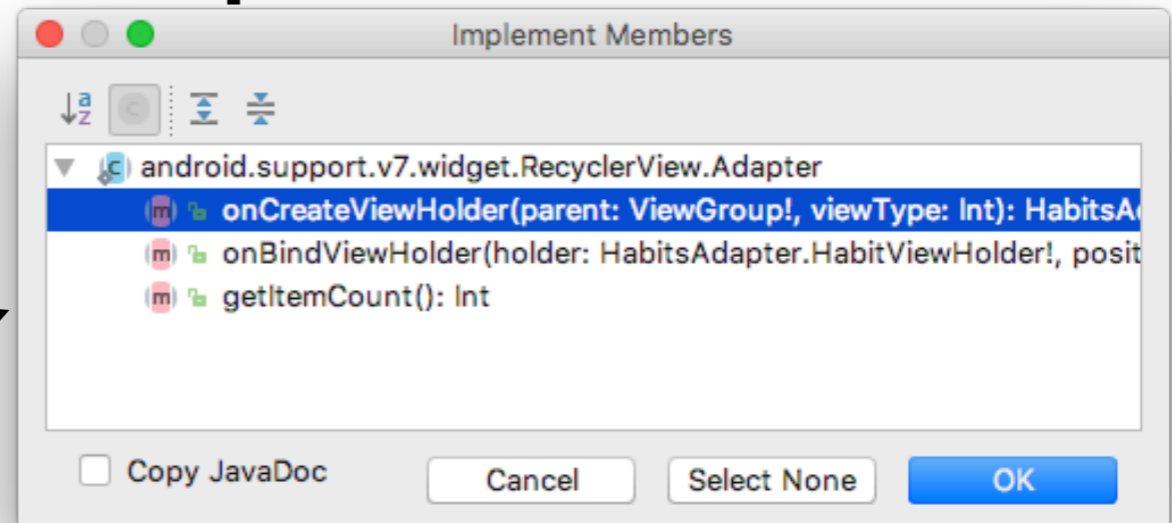
```
    override fun onBindViewHolder(holder: HabitViewHolder?, position: Int) {  
        TODO("not implemented")  
    }
```

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

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

```
}
```

Ctrl-I



kann verkürzt werden auf

```
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)
```

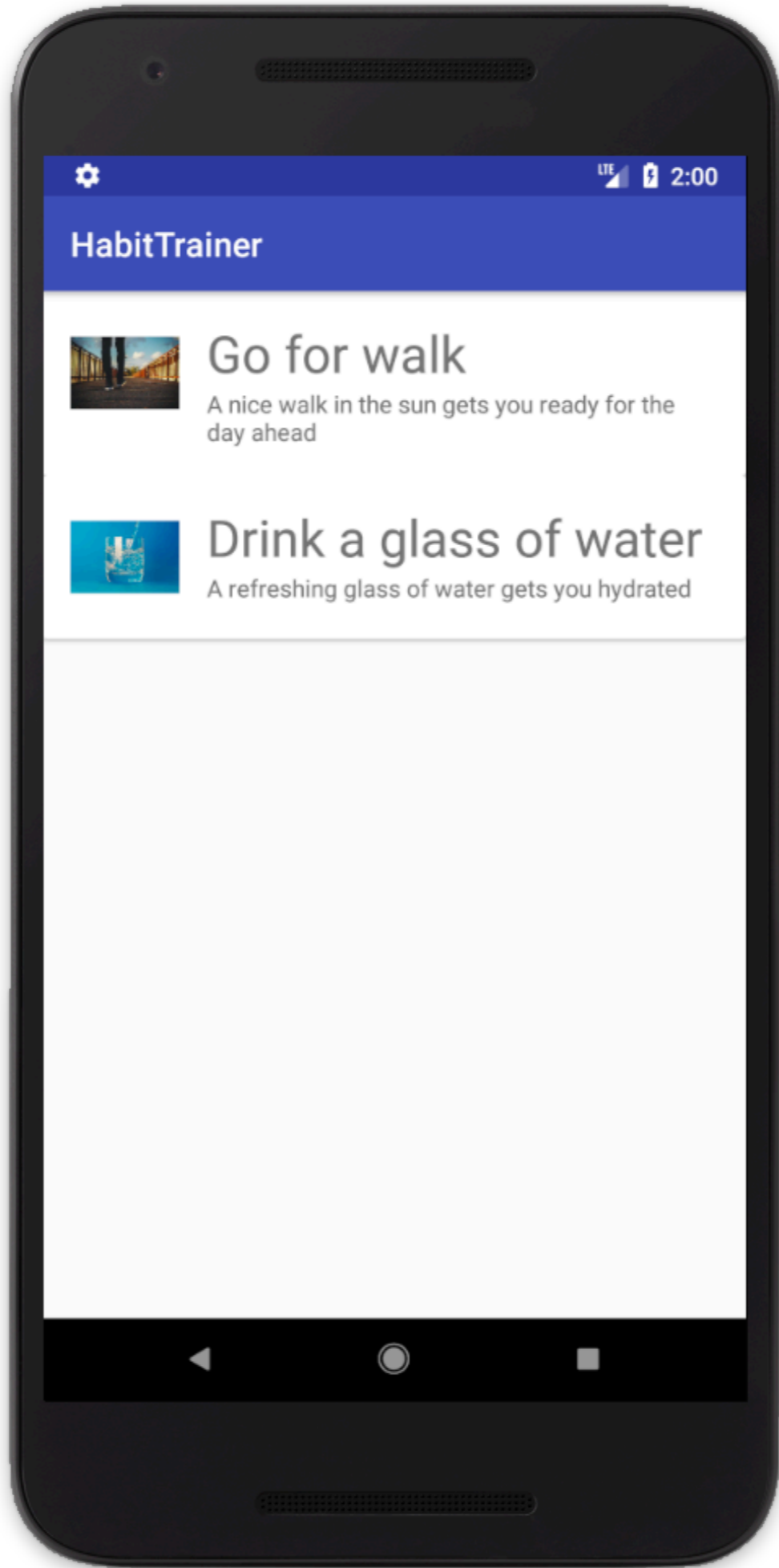
```
            .inflate(R.layout.single_card, parent, false)
```

```
        return HabitViewHolder(view)
```

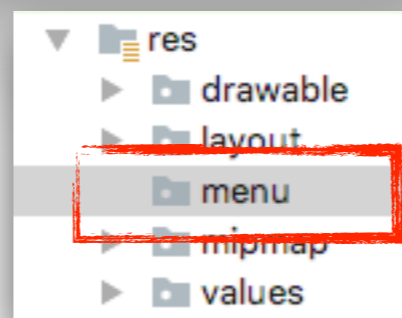
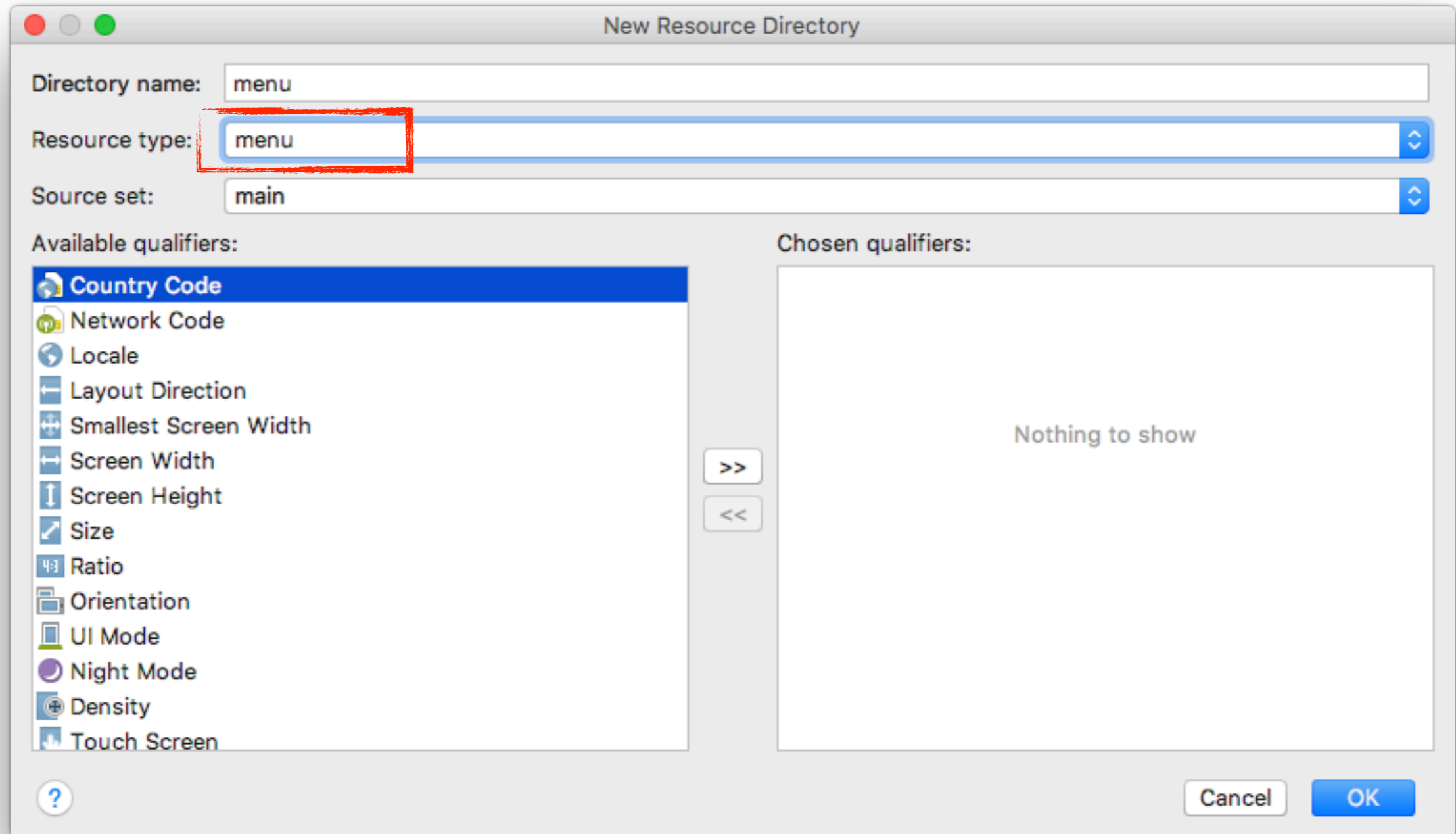
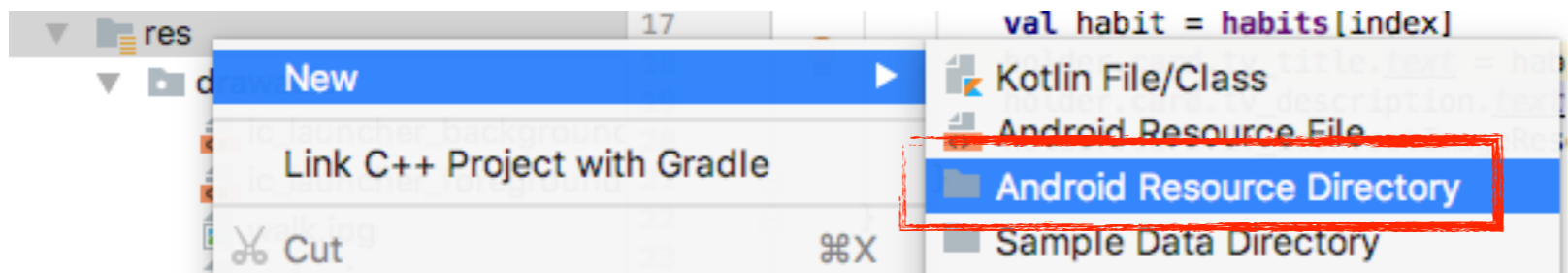
```
    }
```

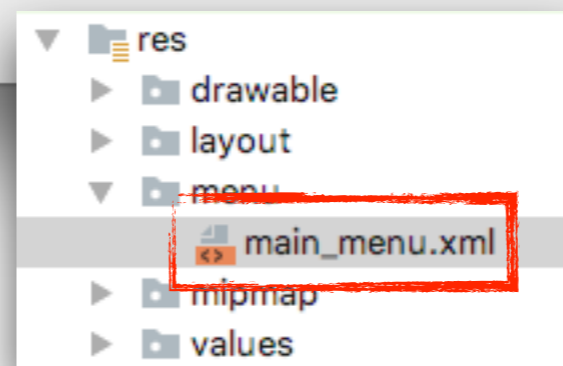
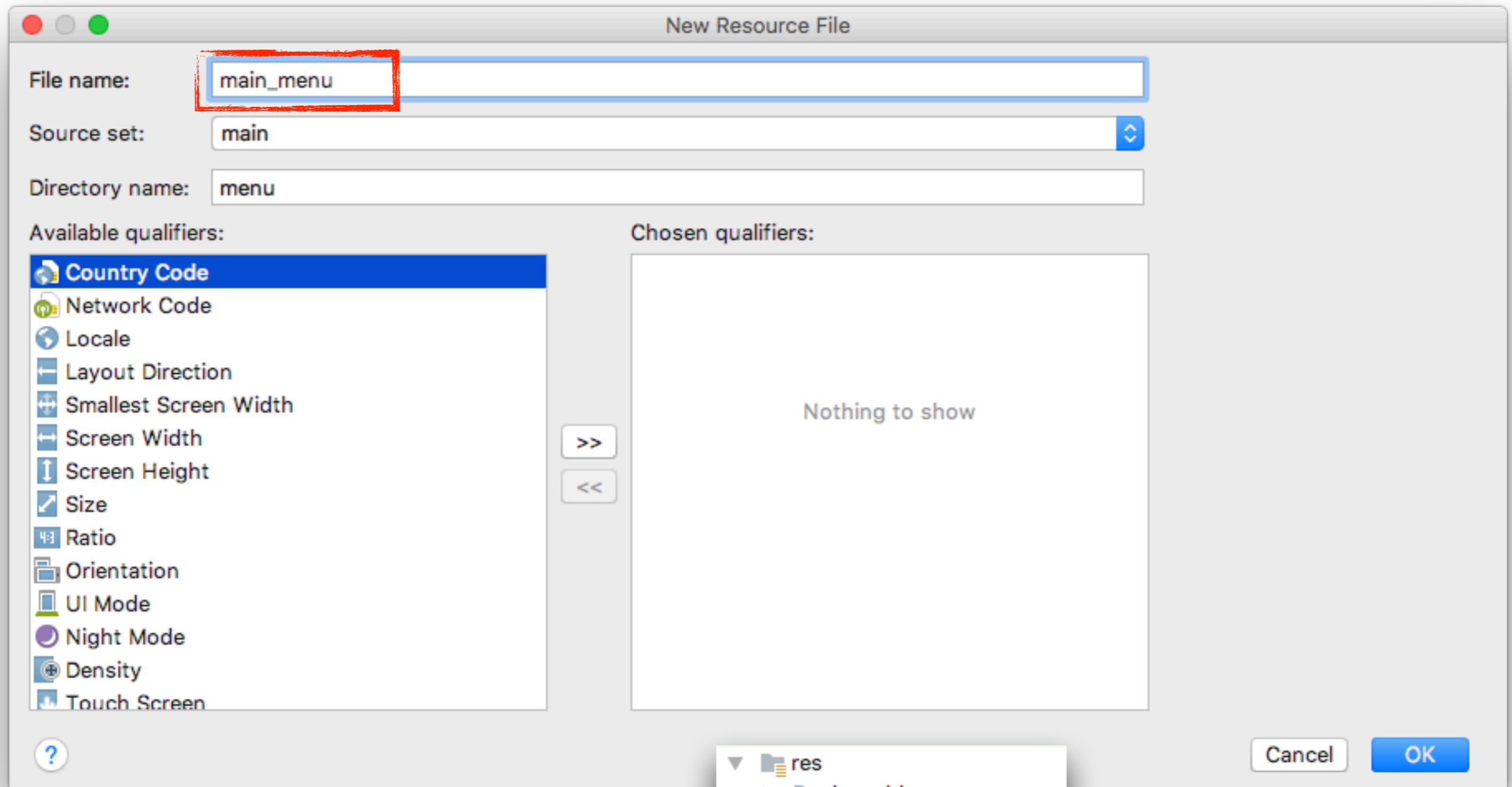
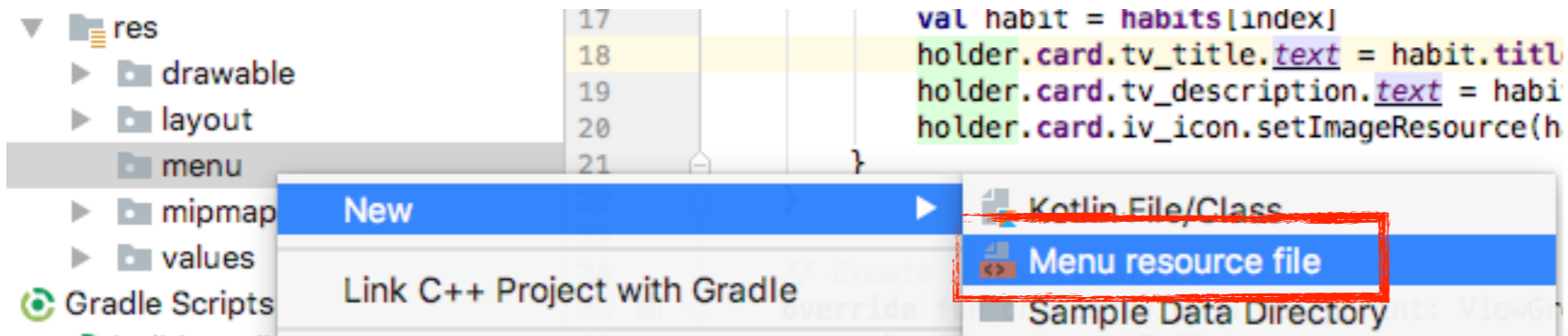
```
    override fun getItemCount() = habits.size
```

```
}
```

Creating a Menu





1

Menu Item

2

add_habit

3

Begin new habit ...

6

4

Add new resource

add_habit (Begin new habit)

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

5

OK

ifRoom

- never
- ifRoom
- always
- withText
- collapseActionView

OK

showAsAction

visible

enabled

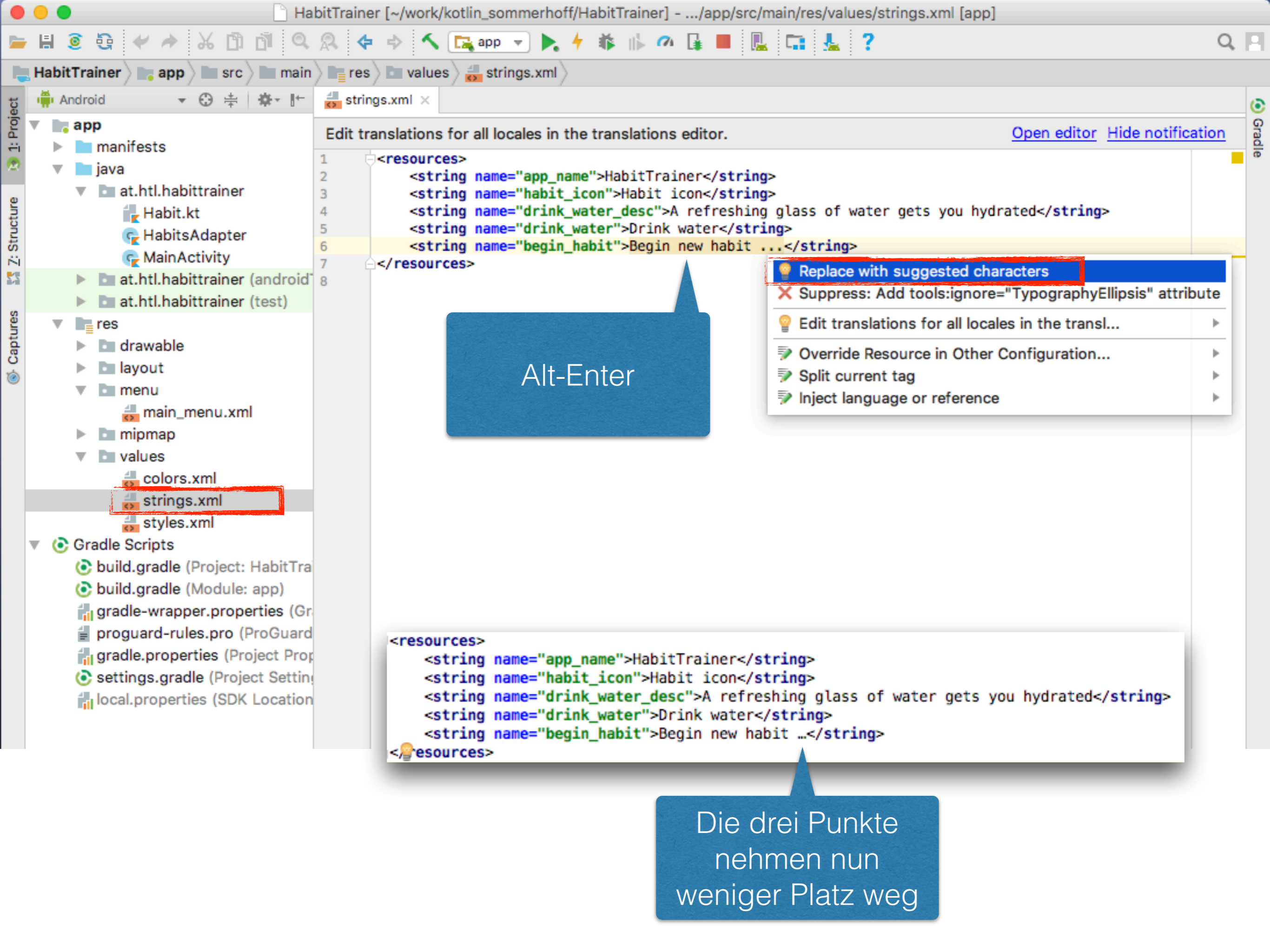
checkable

Project	Resource
app_name	HabitTrain
drink_water	Drink water
drink_water_desc	A refreshing
habit_icon	Habit icon
android	
android.support.design.widget.Snackbar\$SnackbarTextView\$SnackbarTextViewErrorButton	OK
android.support.design.widget.Snackbar\$SnackbarTextView\$SnackbarTextViewErrorTextInvalid	This video
android.support.design.widget.Snackbar\$SnackbarTextView\$SnackbarTextViewErrorTextUnknown	Can't play
android.support.design.widget.Snackbar\$SnackbarTextView\$SnackbarTextViewErrorTitle	Video pro

Preview

Cancel OK

Attributes



Project Structure

- app
 - manifests
 - java
 - at.htl.habittrainer
 - Habit.kt
 - HabitsAdapter
 - MainActivity
 - at.htl.habittrainer (android)
 - at.htl.habittrainer (test)
 - res
 - drawable
 - layout
 - menu
 - main_menu.xml
 - mipmap
 - values
 - colors.xml
 - strings.xml
 - styles.xml
 - Gradle Scripts
 - build.gradle (Project: HabitTra)
 - build.gradle (Module: app)
 - gradle-wrapper.properties (Gr)
 - proguard-rules.pro (ProGuard)
 - gradle.properties (Project Prop)
 - settings.gradle (Project Setting)
 - local.properties (SDK Location)

Edit translations for all locales in the translations editor. [Open editor](#) [Hide notification](#)

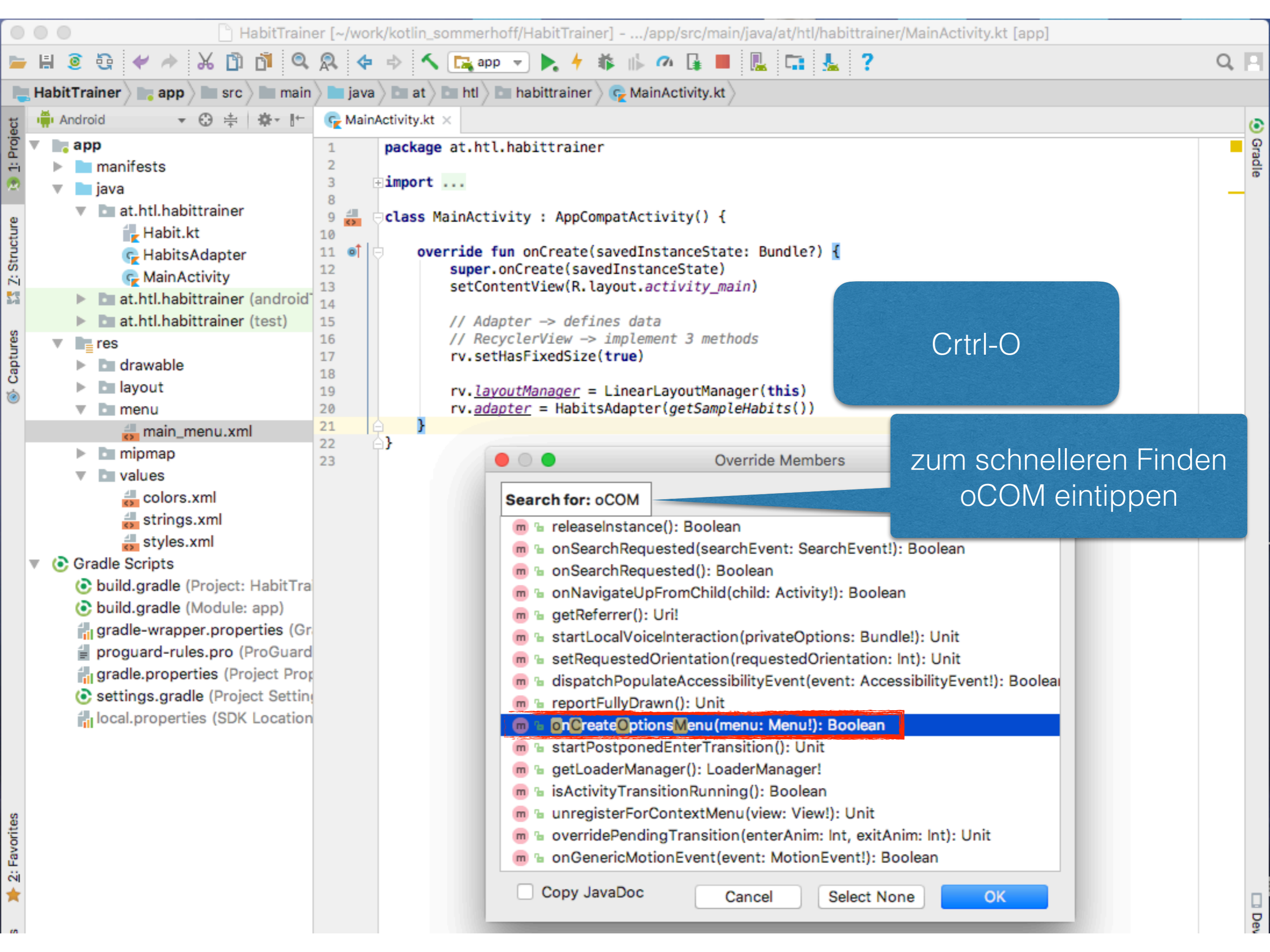
```
1 <resources>
2   <string name="app_name">HabitTrainer</string>
3   <string name="habit_icon">Habit icon</string>
4   <string name="drink_water_desc">A refreshing glass of water gets you hydrated</string>
5   <string name="drink_water">Drink water</string>
6   <string name="begin_habit">Begin new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

Alt-Enter

- 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

```
<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



```
1 package at.htl.habittrainer
2
3 import ...
4
5
6
7
8
9 class MainActivity : AppCompatActivity() {
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_main)
14
15         // Adapter -> defines data
16         // RecyclerView -> implement 3 methods
17         rv.setHasFixedSize(true)
18
19         rv.layoutManager = LinearLayoutManager(this)
20         rv.adapter = HabitsAdapter(getSampleHabits())
21     }
22 }
23
```

Ctrl-O

zum schnelleren Finden
oCOM eintippen

Override Members

Search for: oCOM

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

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)
```

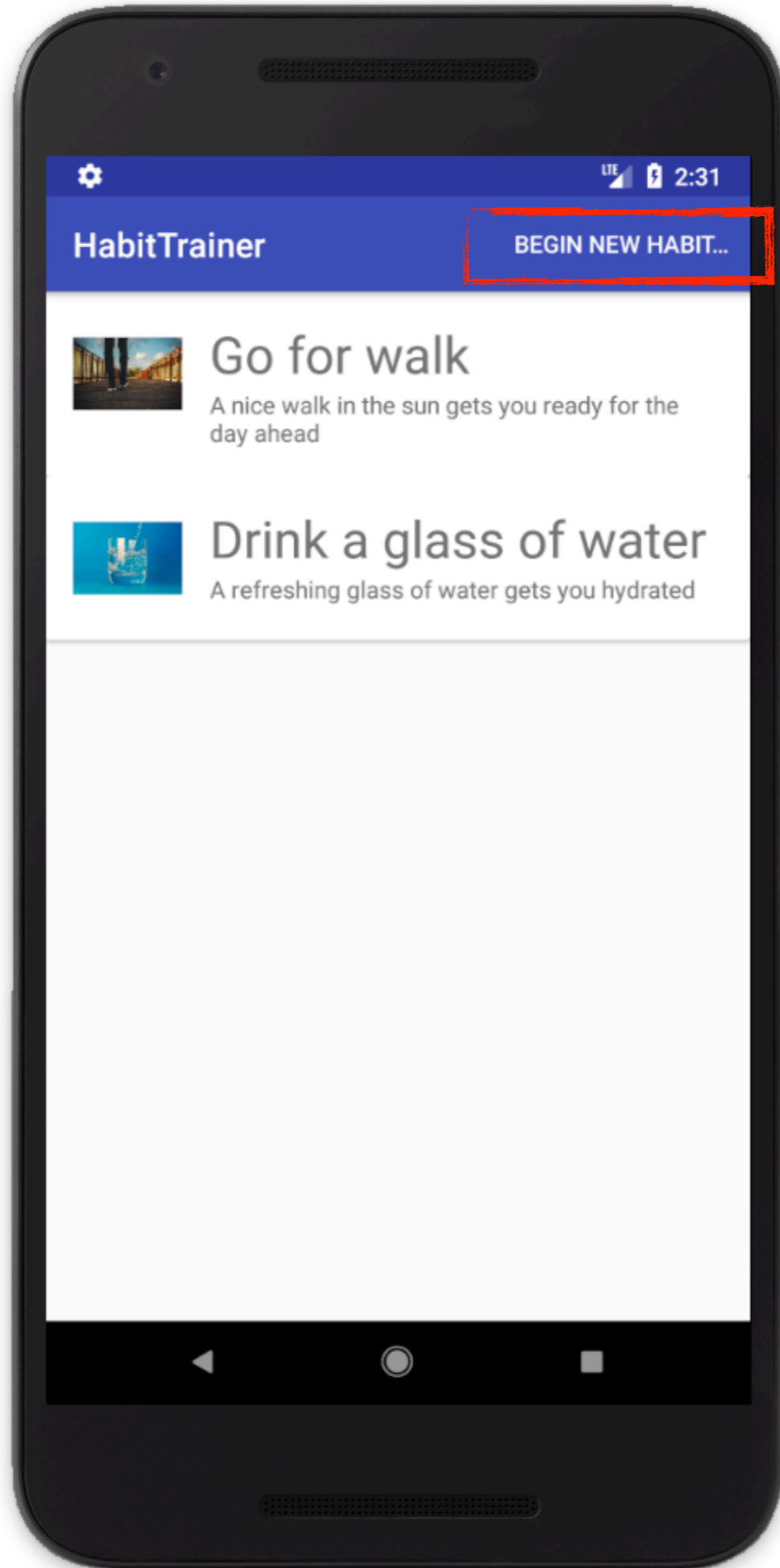
```
        rv.layoutManager = LinearLayoutManager(this)
        rv.adapter = HabitsAdapter(getSampleHabits())
```

```
    }
```

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

```
    }
```

```
}
```

BEGIN NEW HABIT...



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

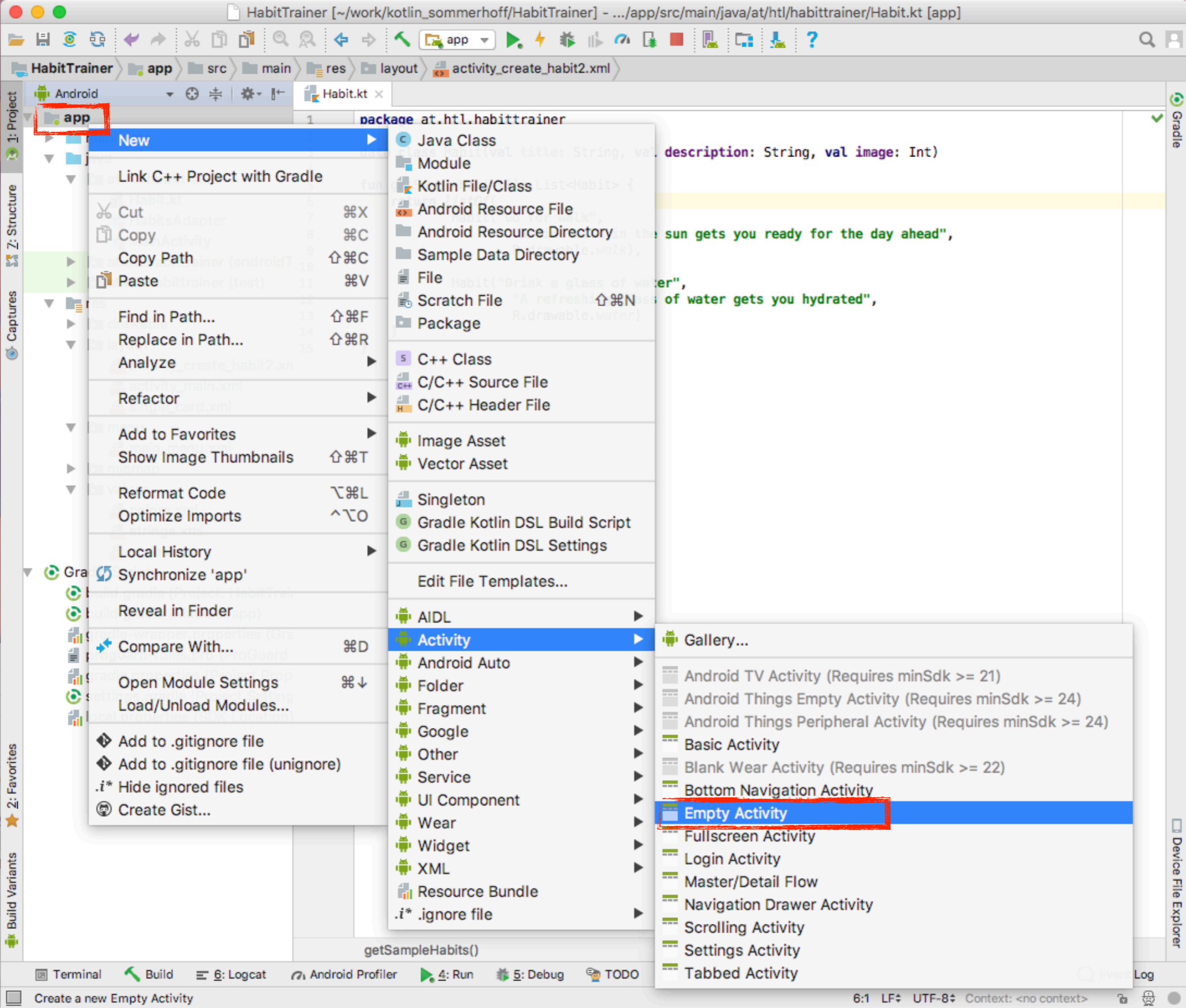
- Try to build a layout (no functionality, just XML) with
 - An `EditText` for the title
 - An `EditText` for the description
 - A `Button` to choose an image
 - An `ImageView` to preview the selected image
 - A `Button` to save the new habit
 - A `TextView` to show possible error messages to the user
 - --
 - 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



New

Link C++ Project with Gradle

Cut

Copy

Copy Path

Paste

Find in Path...

Replace in Path...

Analyze

Refactor

Add to Favorites

Show Image Thumbnails

Reformat Code

Optimize Imports

Local History

Synchronize 'app'

Reveal in Finder

Compare With...

Open Module Settings

Load/Unload Modules...

Add to .gitignore file

Add to .gitignore file (unignore)

.i* Hide ignored files

Create Gist...

Java Class

Module

Kotlin File/Class

Android Resource File

Android Resource Directory

Sample Data Directory

File

Scratch File

Package

C++ Class

C/C++ Source File

C/C++ Header File

Image Asset

Vector Asset

Singleton

Gradle Kotlin DSL Build Script

Gradle Kotlin DSL Settings

Edit File Templates...

AIDL

Activity

Android Auto

Folder

Fragment

Google

Other

Service

UI Component

Wear

Widget

XML

Resource Bundle

.i* .ignore file

description: String, val image: Int)

sun gets you ready for the day ahead",

of water gets you hydrated",

Gallery...

Android TV Activity (Requires minSdk >= 21)

Android Things Empty Activity (Requires minSdk >= 24)

Android Things Peripheral Activity (Requires minSdk >= 24)

Basic Activity

Blank Wear Activity (Requires minSdk >= 22)

Bottom Navigation Activity

Empty Activity

Fullscreen Activity

Login Activity

Master/Detail Flow

Navigation Drawer Activity

Scrolling Activity

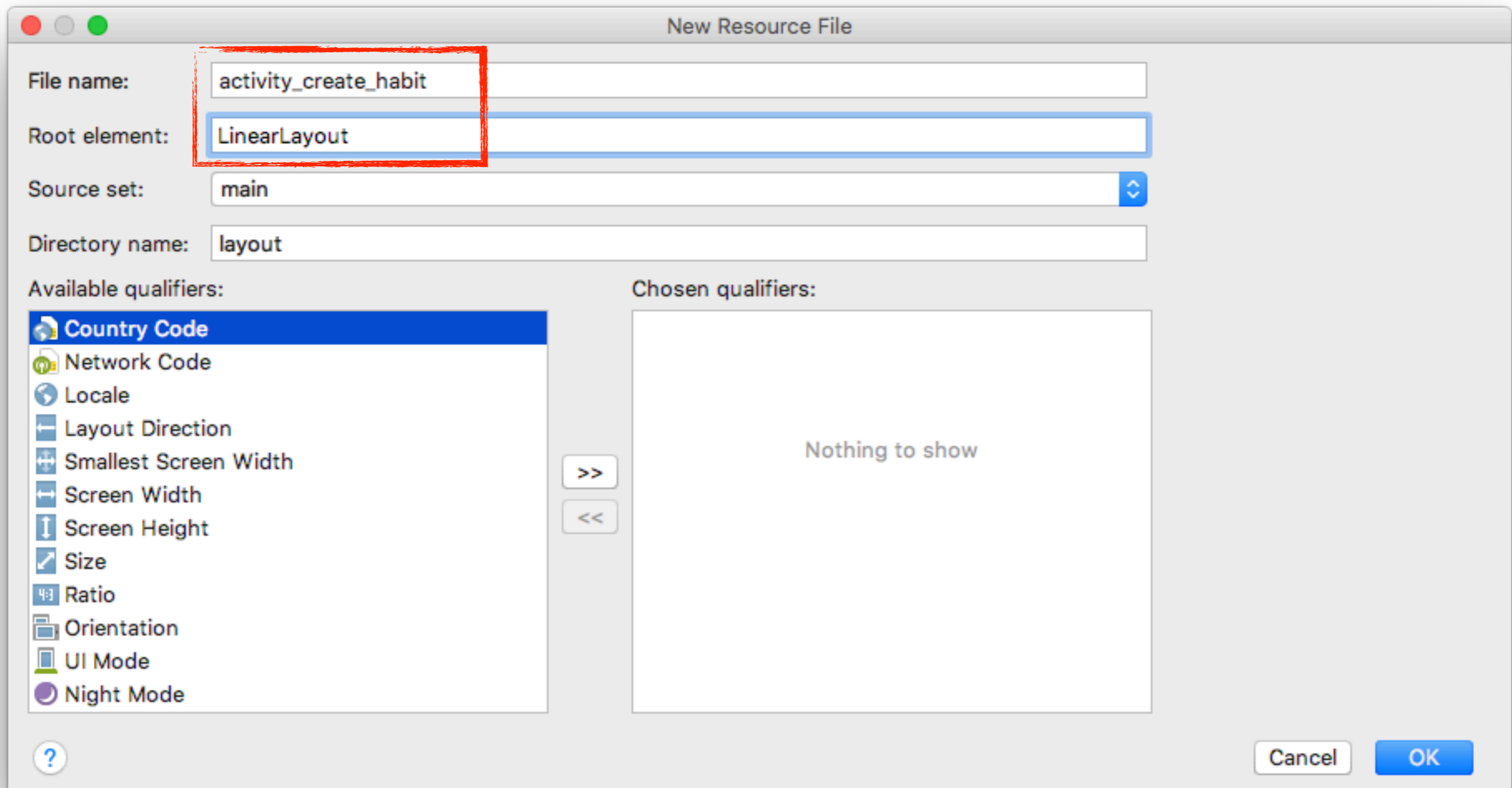
Settings Activity

Tabbed Activity

getSampleHabits()

6:1 LF+ UTF-8+ Context: <no context>

Layout File anlegen



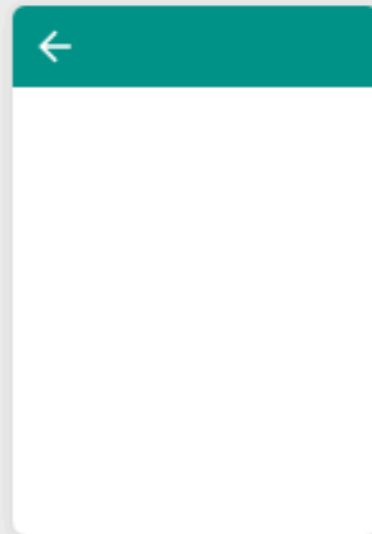


Configure Activity

Android Studio



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

Cancel

Previous

Next

Finish

Project Structure:

- app
 - 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)
 - proguard-rules.pro (ProGuard Rule)
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)
 - local.properties (SDK Location)

```
1 package at.htl.habittrainer
2
3 import ...
4
5
6
7
8
9
10 class MainActivity : AppCompatActivity() {
11
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15
16         // Adapter -> defines data
17         // RecyclerView -> implement 3 methods
18         rv.setHasFixedSize(true)
19
20         rv.layoutManager = LinearLayoutManager(this)
21         rv.adapter = HabitsAdapter(getSampleHabits())
22     }
23
24     override fun onCreateOptionsMenu(): Boolean {
25         menuInflater.inflate(R.menu.menu_main, this)
26         return true
27     }
28
29
30
31
32 }
```

In MainActivity.kt
<Ctrl>-O zum
Überschreiben

Override Members

Search for: oois

- m onWindowStartingActionMode(callback: ActionMode.Callback!, type: Int): Boolean
- m getLocalClassName(): String!
- m getPreferences(mode: Int): SharedPreferences!
- m getCurrentFocus(): View!
- m onRestart(): Unit
- m startActivityIfNeeded(intent: Intent!, requestCode: Int): Boolean
- m startActivityIfNeeded(intent: Intent!, requestCode: Int, options: Bundle!)
- m **onOptionsItemSelected(): Boolean**
- m setIntent(newIntent: Intent!): Unit
- m getFragmentManager(): FragmentManager!
- m getCallingPackage(): String!
- m showAssist(args: Bundle!): Boolean
- m startSearch(initialQuery: String!, selectInitialQuery: Boolean, appSearchDialog: AppSearchDialog): Boolean
- m finishAfterTransition(): Unit
- m onRestoreInstanceState(savedInstanceState: Bundle!): Unit
- m onRestoreInstanceState(savedInstanceState: Bundle!, persistentState: Parcelable): Unit

Copy JavaDoc

Cancel Select None OK

Intents

- implizite Intents
- explizite Intents

MainActivity.kt

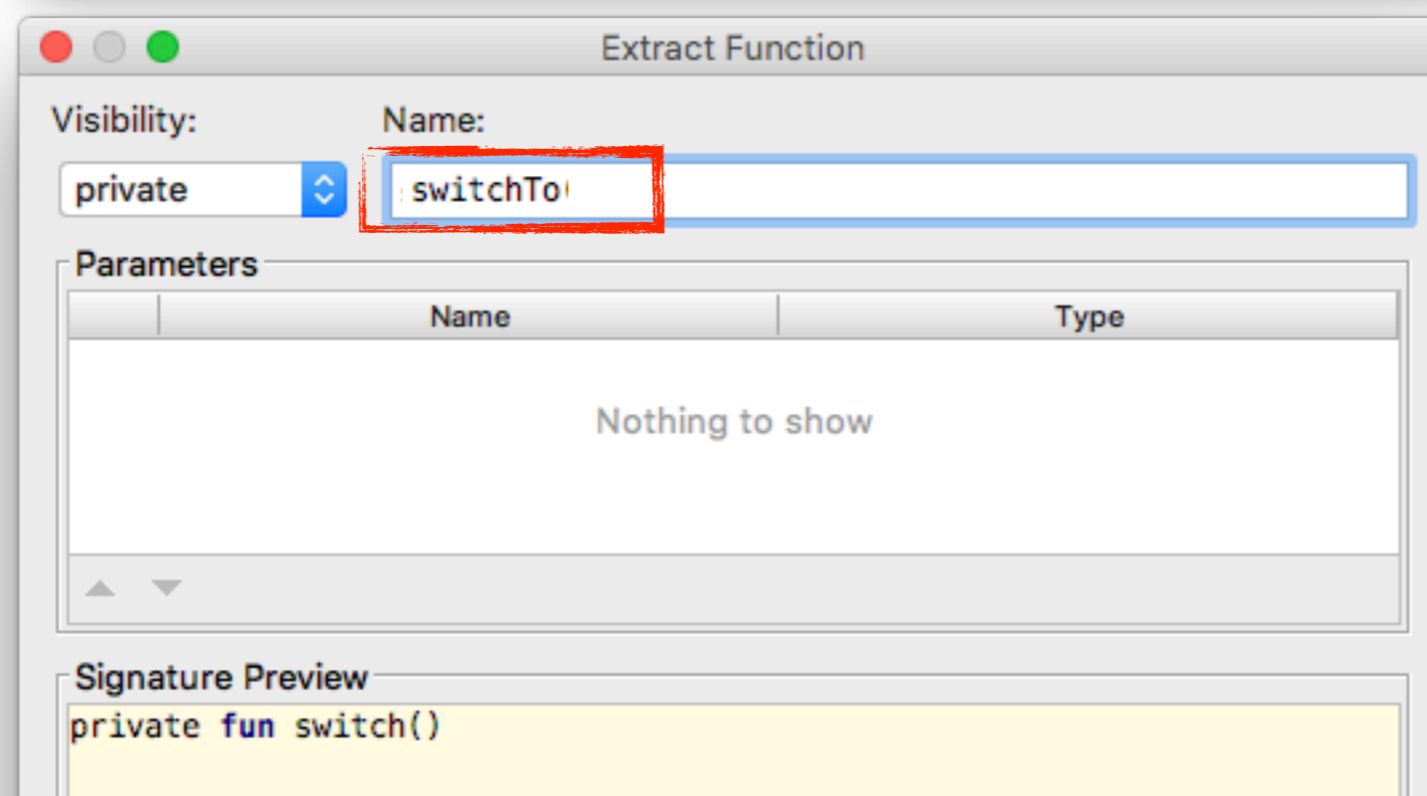
```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        // Adapter -> defines data  
        // RecyclerView -> implement 3 methods  
        rv.setHasFixedSize(true)  
  
        rv.layoutManager = LinearLayoutManager(this)  
        rv.adapter = HabitsAdapter(getSampleHabits())  
    }  
  
    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

```
MainActivity.kt x
12 class MainActivity : AppCompatActivity() {
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_main)
17
18         // Adapter -> defines data
19         // RecyclerView -> implement 3 methods
20         rv.setHasFixedSize(true)
21
22         rv.layoutManager = LinearLayoutManager(this)
23         rv.adapter = HabitsAdapter(getSampleHabits())
24     }
25
26     override fun onCreateOptionsMenu(menu: Menu?): Boolean {
27         menuInflater.inflate(R.menu.main_menu, menu)
28         return true
29     }
30
31     override fun onOptionsItemSelected(item: MenuItem): Boolean {
32         if (item.itemId == R.id.add_habit) {
33             val intent = Intent(this, CreateHabitActivity::class.java)
34             startActivity(intent)
35         }
36         return true
37     }
38 }
39 }
```

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 $\text{⌘}M$ (Ctrl+Alt+M for Win/Linux)



MainActivity.kt - switchTo()

```
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

```

<?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>

```

activity_create _habit.xml

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>

```

```

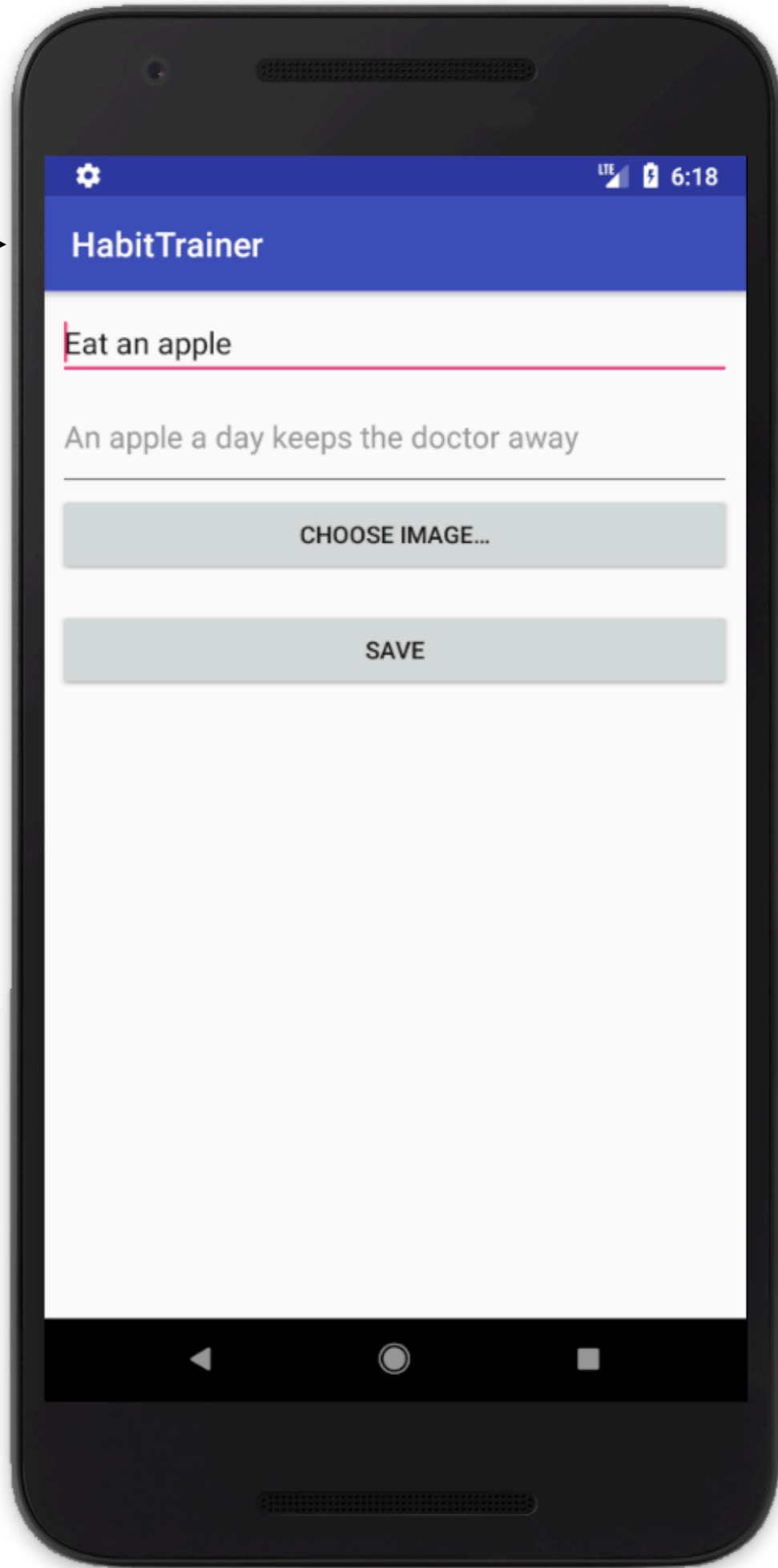
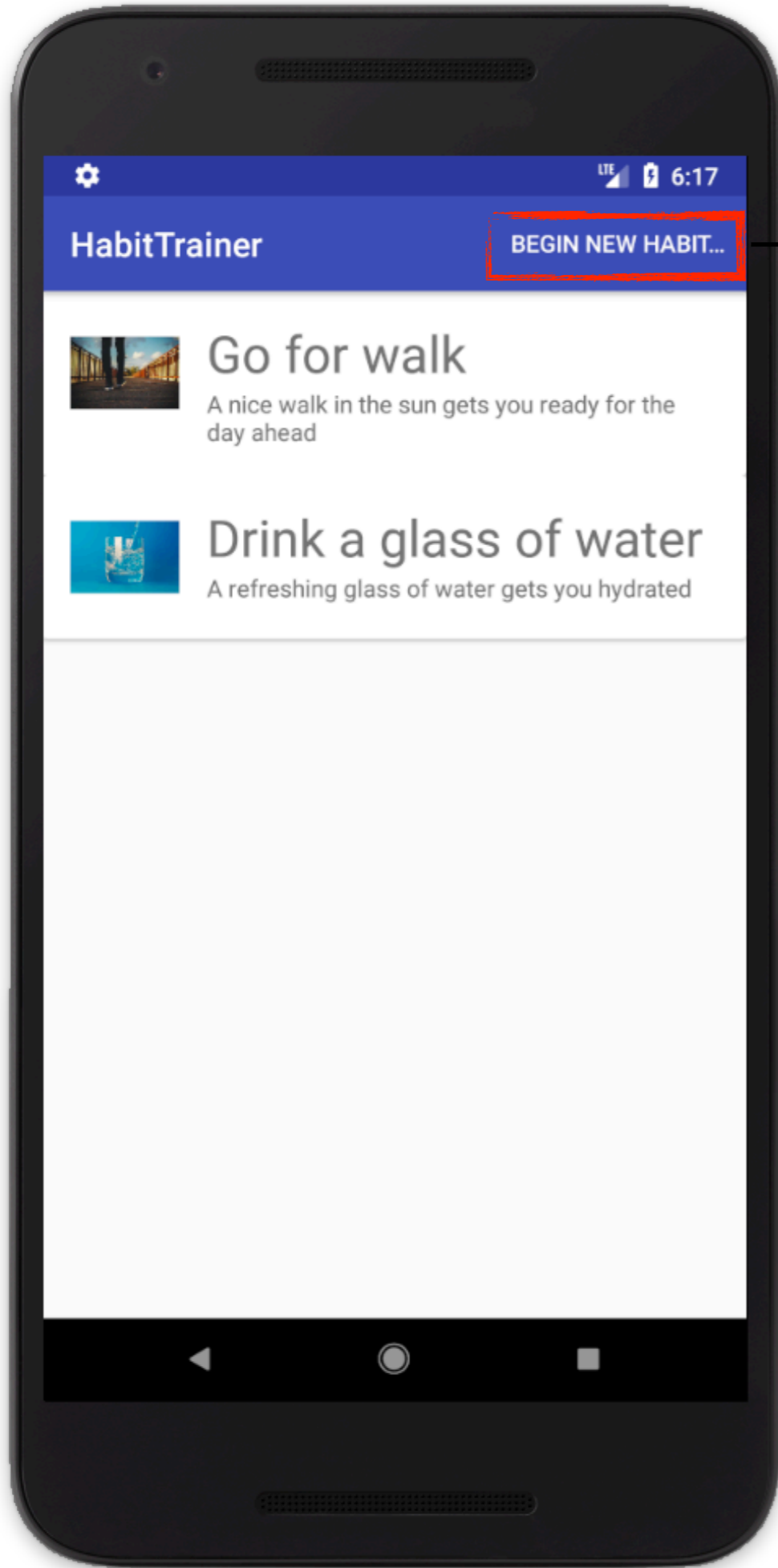
<string name="choose_image">Choose image...</string>
resources>

```

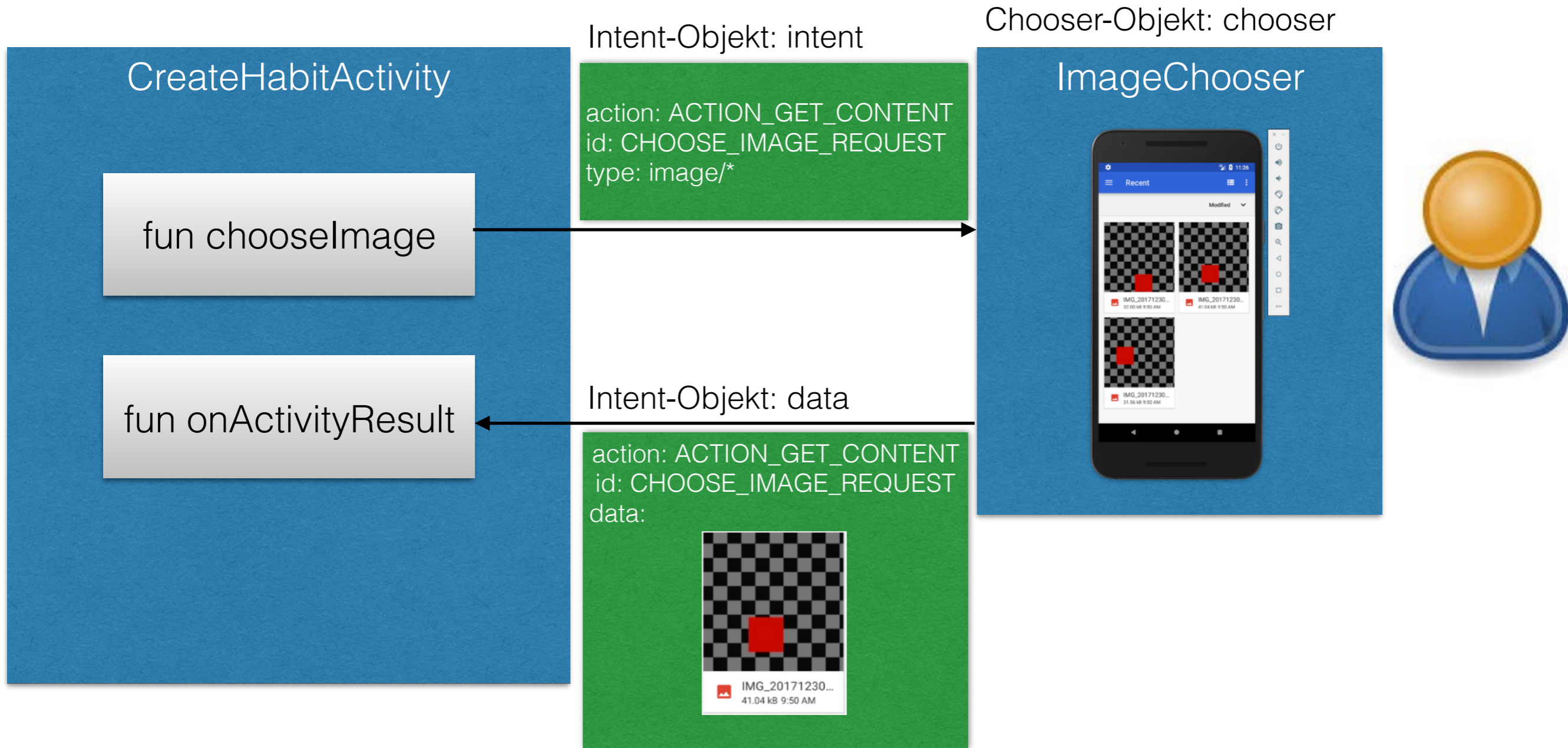
Replace with suggested characters

Guidelines für Layout

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



Intents



onClick-Methode 1

```
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

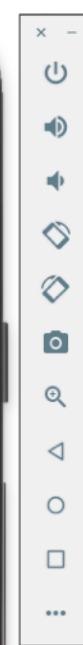
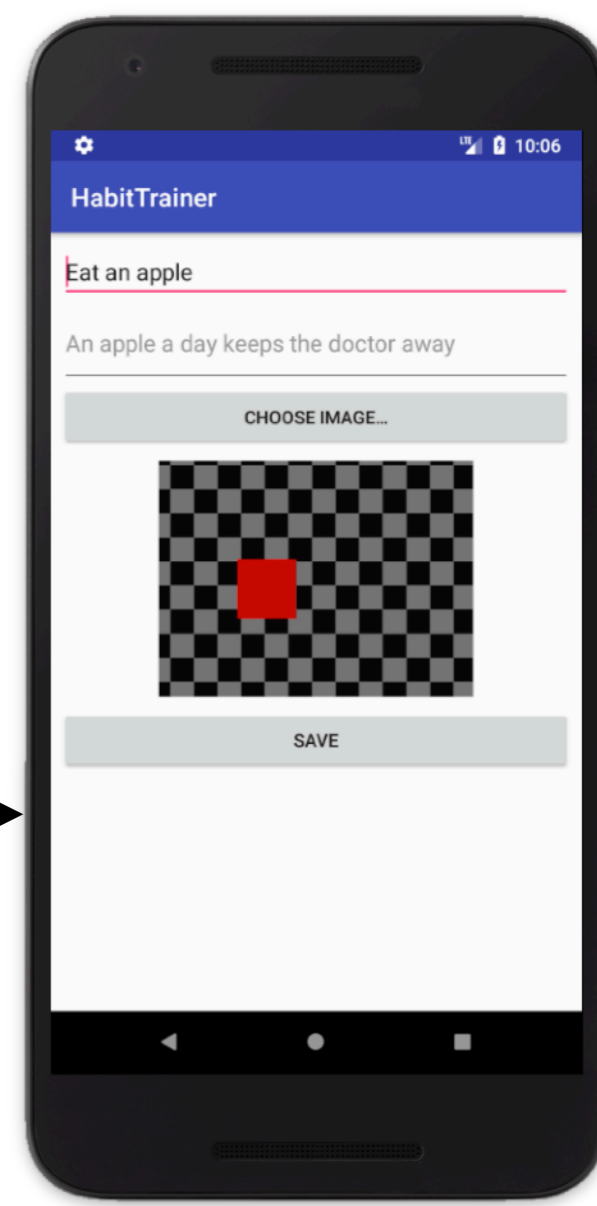
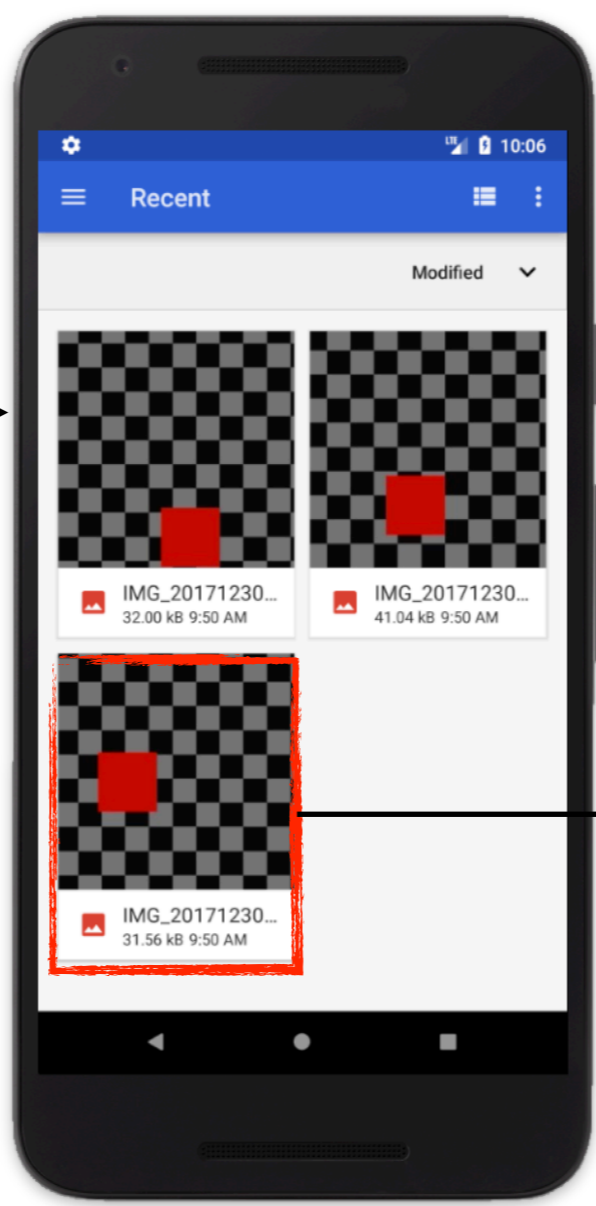
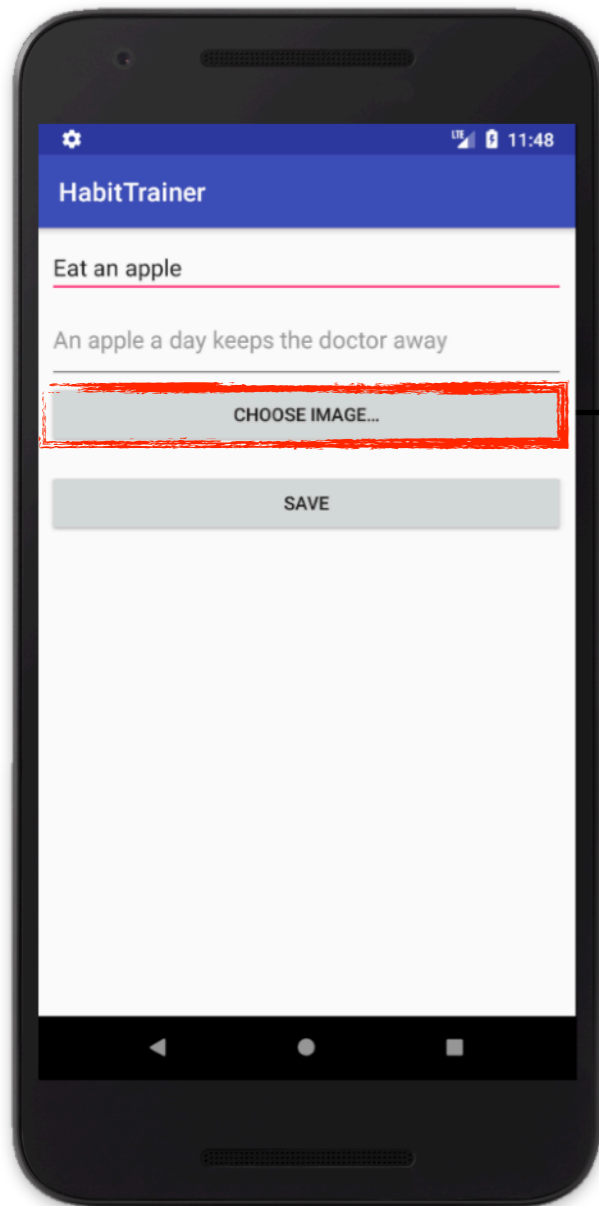
Project Structure

- app
 - 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 Vers)
 - proguard-rules.pro (ProGuard Rules fo
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)

```
20
21 override fun onCreate(savedInstanceState: Bundle?) {
22     super.onCreate(savedInstanceState)
23     setContentView(R.layout.activity_create_habit)
24 }
25
26 fun chooseImage(v: View) {
27     val intent = Intent()
28     intent.type = "image/*"
29     intent.action = Intent.ACTION_GET_CONTENT
30
31     val chooser = Intent.createChooser(intent, "Choose image for habit")
32     startActivityForResult(chooser, CHOOSE_IMAGE_REQUEST)
33
34     Log.d(TAG, "Intent to choose image sent ...")
35 }
36
37 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
38     super.onActivityResult(requestCode, resultCode, data)
39
40     if (requestCode == CHOOSE_IMAGE_REQUEST
41         && resultCode == Activity.RESULT_OK
42         && data != null
43         && data.data != null) {
44
45         Log.d(TAG, "An image was chosen by the user")
46
47         val bitmap = tryReadBitmap(data.data)
48
49         bitmap?.let {
50             iv_image.setImageBitmap(bitmap)
51             Log.d(TAG, "Read image bitmap and updated image view.")
52         }
53     }
54 }
```

Logcat

```
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
```

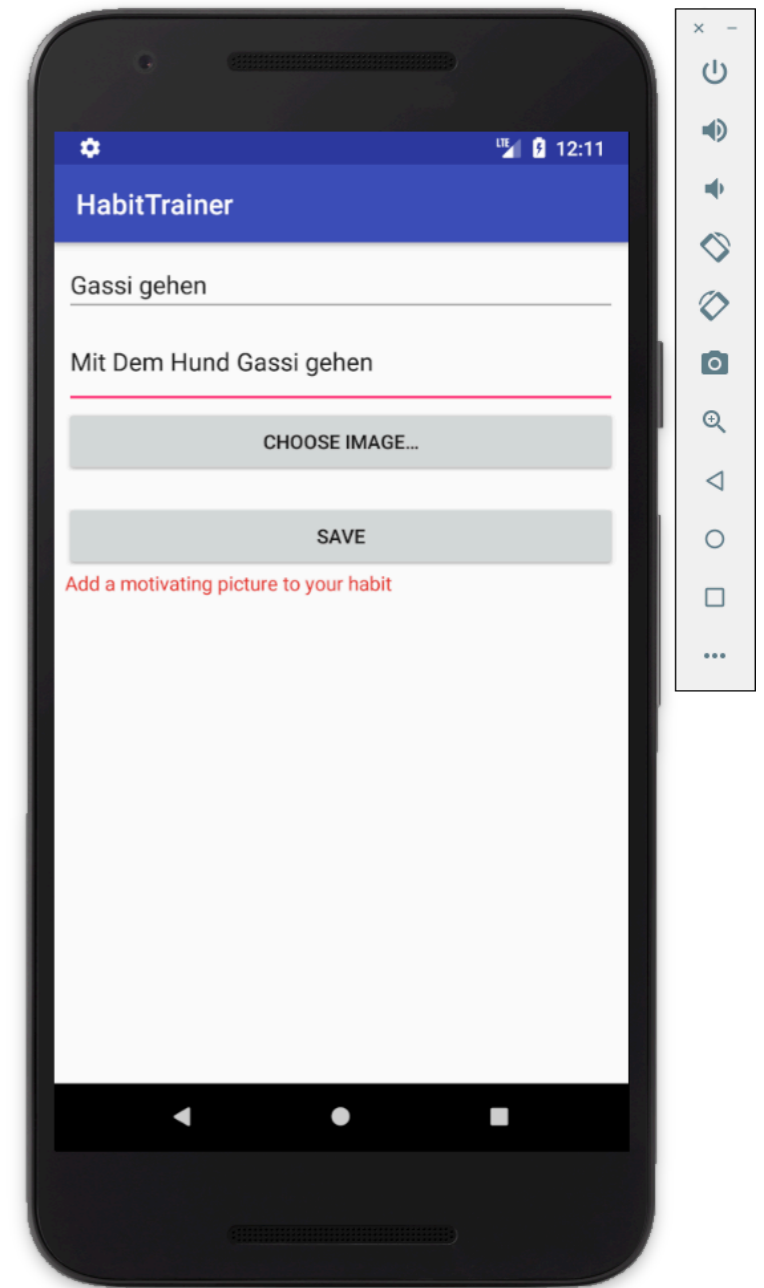
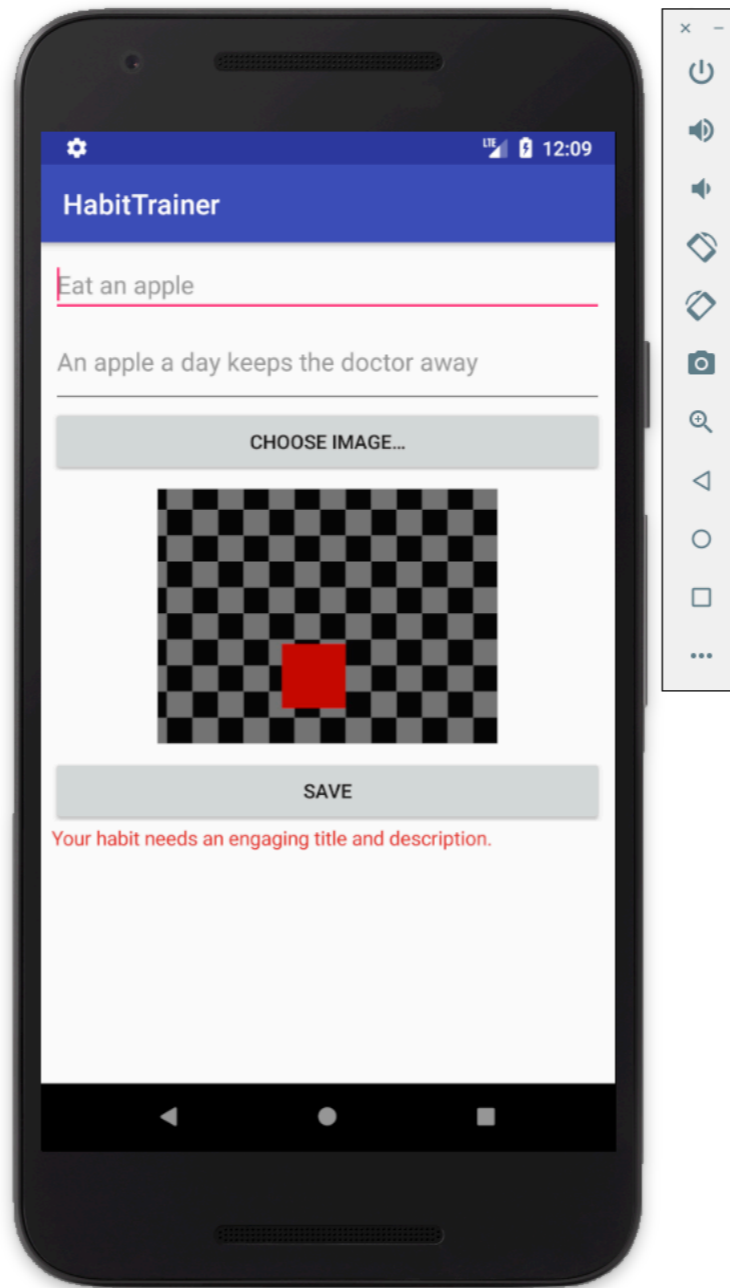
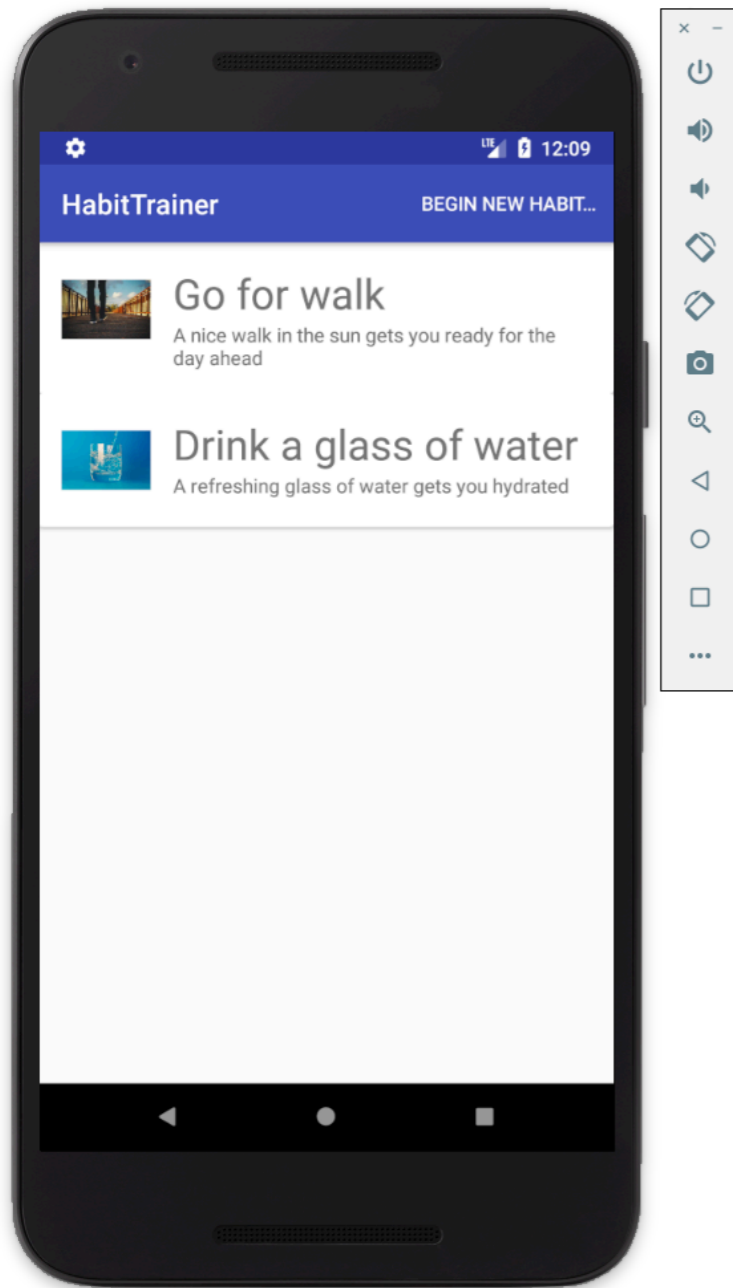


Save-Button

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

CreateHabitActivity

```
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? {  
        ...  
    }  
}
```



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
// }
```

Auch diese Schreibweise
wäre möglich

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


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

```
1 package at.htl.habittrainer.db
2
3 import android.content.Context
4 import android.database.sqlite.SQLiteDatabase
5 import android.database.sqlite.SQLiteOpenHelper
6
7 class HabitTrainerDb(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VRSION) {
8
9
10 }
```

Implement Members

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

Copy JavaDoc

HabitTrainerDb.kt

```
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

```
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

```
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, values)  
            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, stream)  
        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

```
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

```
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)
    }
}
```


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

Project: 1: HabitTrainer

- app
 - manifests
 - java
 - at.htl.habittrainer
 - db
 - Contracts.kt
 - HabitDbTable.kt
 - HabitTrainerDb
 - CreateHabitActivity.kt
 - Habit
 - HabitsAdapter
 - MainActivity
 - at.htl.habittrainer (android)
 - at.htl.habittrainer (test)
 - res
 - drawable
 - layout
 - activity_create_habit.xml
 - activity_create_habit2.xml
 - activity_main.xml
 - single_card.xml
 - menu

Structure: Z: Structure

Captures

Gradle

```
27 setContentView(R.layout.activity_create_habit)
28 }
29
30 fun storeHabit(v: View) {
31     if (et_title.isBlank()
32         || et_descr.isBlank()) {
33         Log.d(TAG, "No habit stored: title or description missing.")
34         displayErrorMessage("Your habit needs an engaging title and description.")
35         return
36     } else if (imageBitmap == null) {
37         Log.d(TAG, "No habit stored: image missing.")
38         displayErrorMessage("Add a motivating picture to your habit")
39         return
40     }
41
42     // store the habit in database ...
43     val title = et_title.text.toString()
44     val description = et_descr.text.toString()
45     val habit = Habit(title, description, imageBitmap!!)
46
47     val id = HabitDbTable(this).store(habit)
48
49     if (id == -1L) {
50         displayErrorMessage("Habit could not be stored... let's not make this a habit")
51     } else {
52         val intent = Intent(this, MainActivity::class.java)
53         startActivity(intent)
54     }
55 }
```

Logcat

Emulator Nexus_5X_API_27 | at.htl.habittrainer (4849) | Debug | Regex | Show only selected applicatio

```
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@f81c42d)
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)
```

Build Variants: 2: Favorites

Terminal | Build | 6: Logcat | Android Profiler | 4: Run | 5: Debug | TODO | Event Log

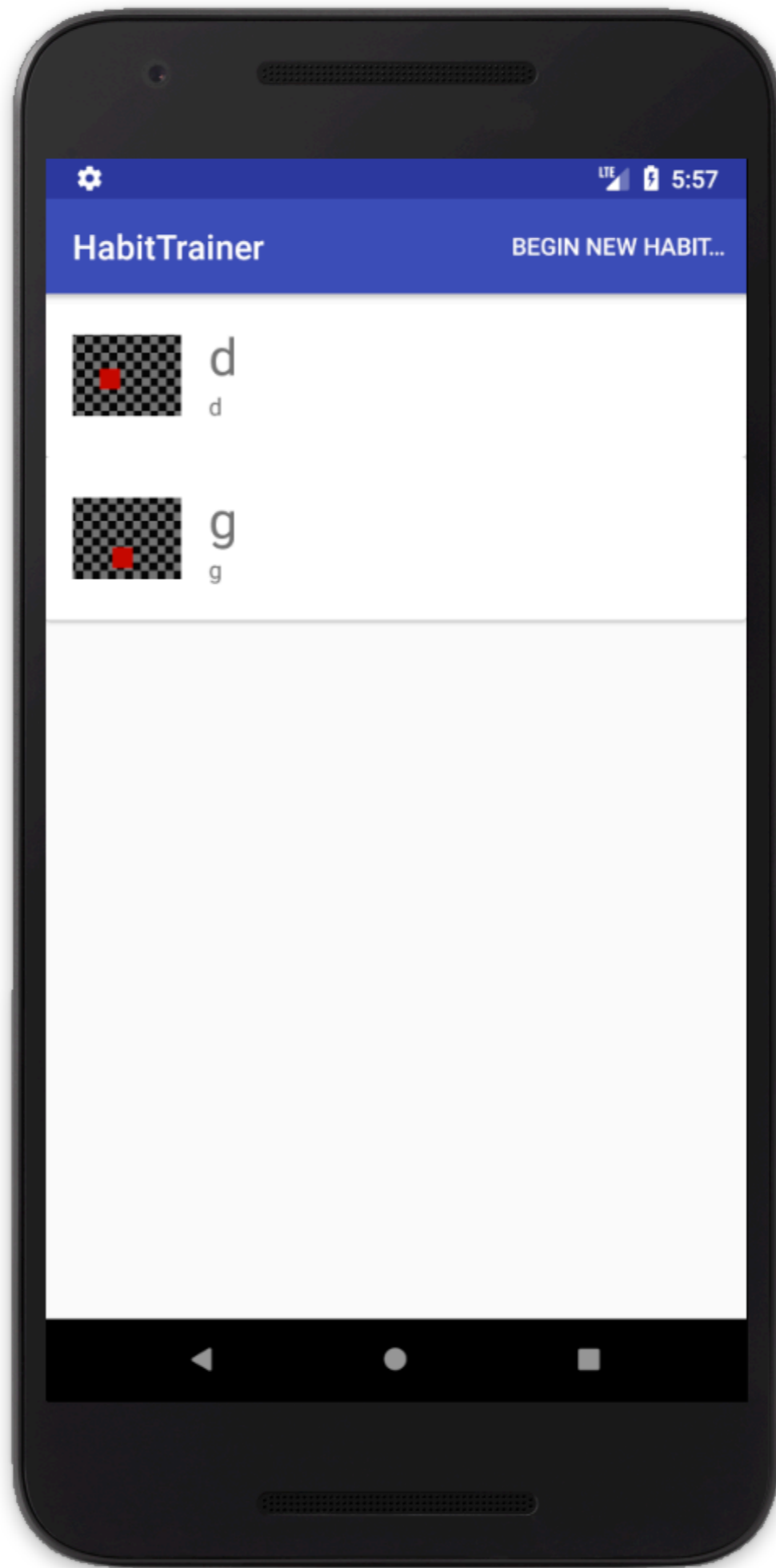
Gradle build finished in 1s 559ms (3 minutes ago) | 96 chars | 33:161 | LF | UTF-8 | Context: <no context>

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())  
}
```



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
    }
}

```

Auch hier vereinfachen extension functions die Aufrufe sehr

```
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)
}

```

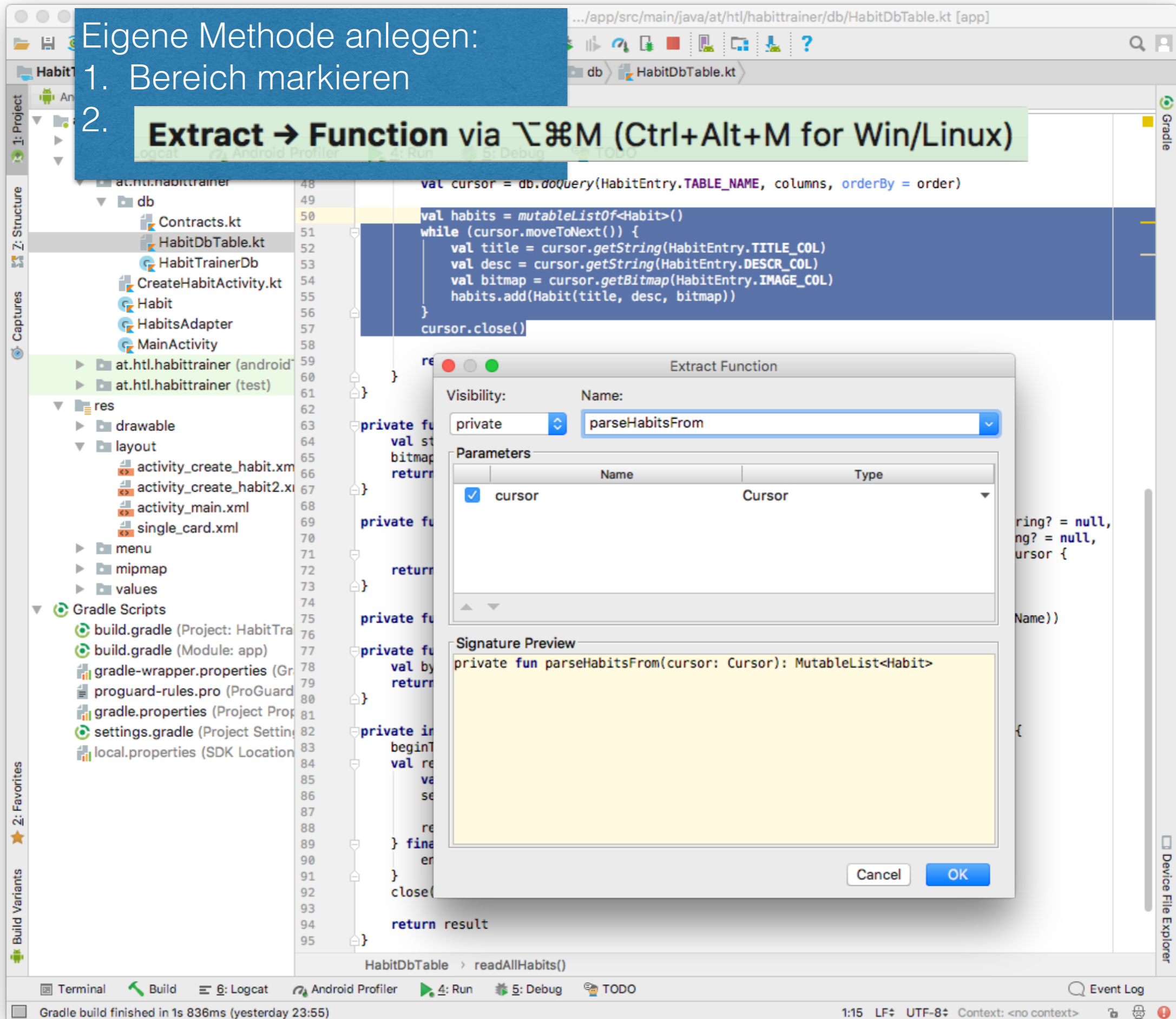
```
private inline fun <T> SQLiteDatabase.transaction(function: SQLiteDatabase.() -> T): T { ... }
```


Eigene Methode anlegen:

1. Bereich markieren

2.

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



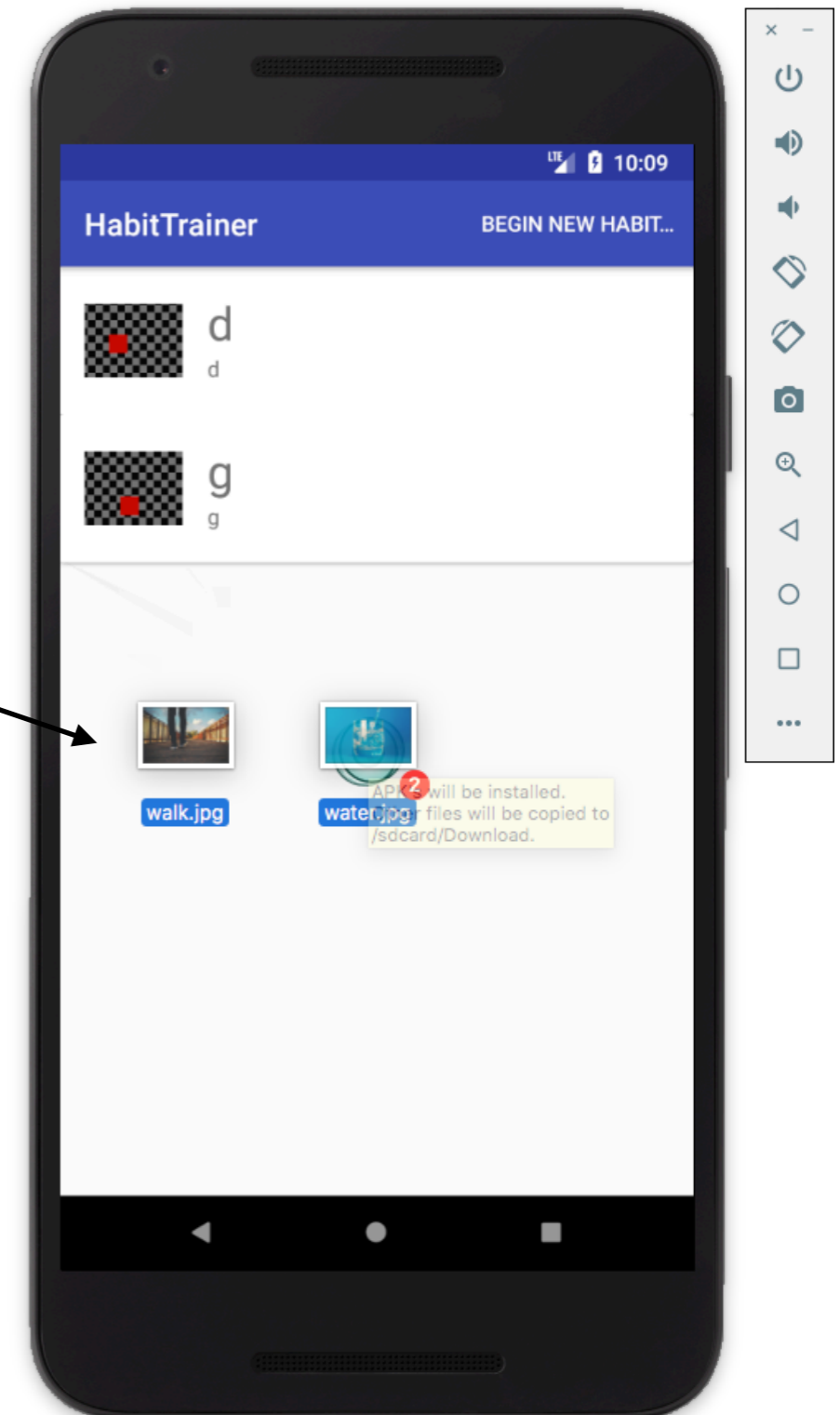
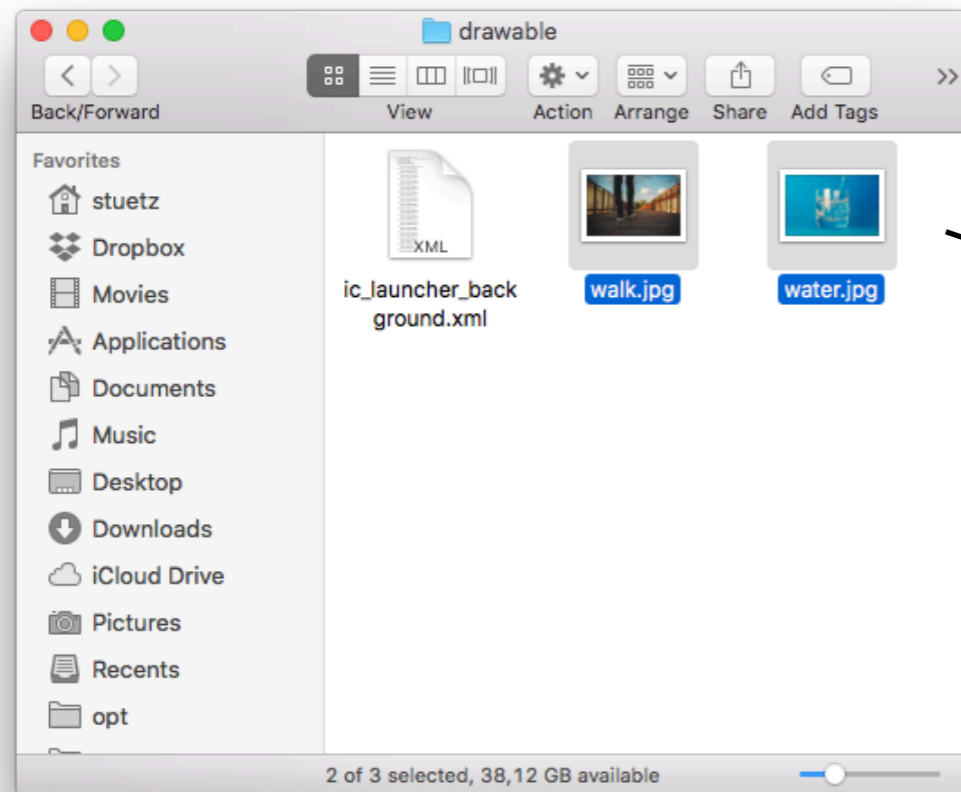
Methode extrahieren

```
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  
}
```

Das Ergebnis ist ein
durchaus lesbarer Code

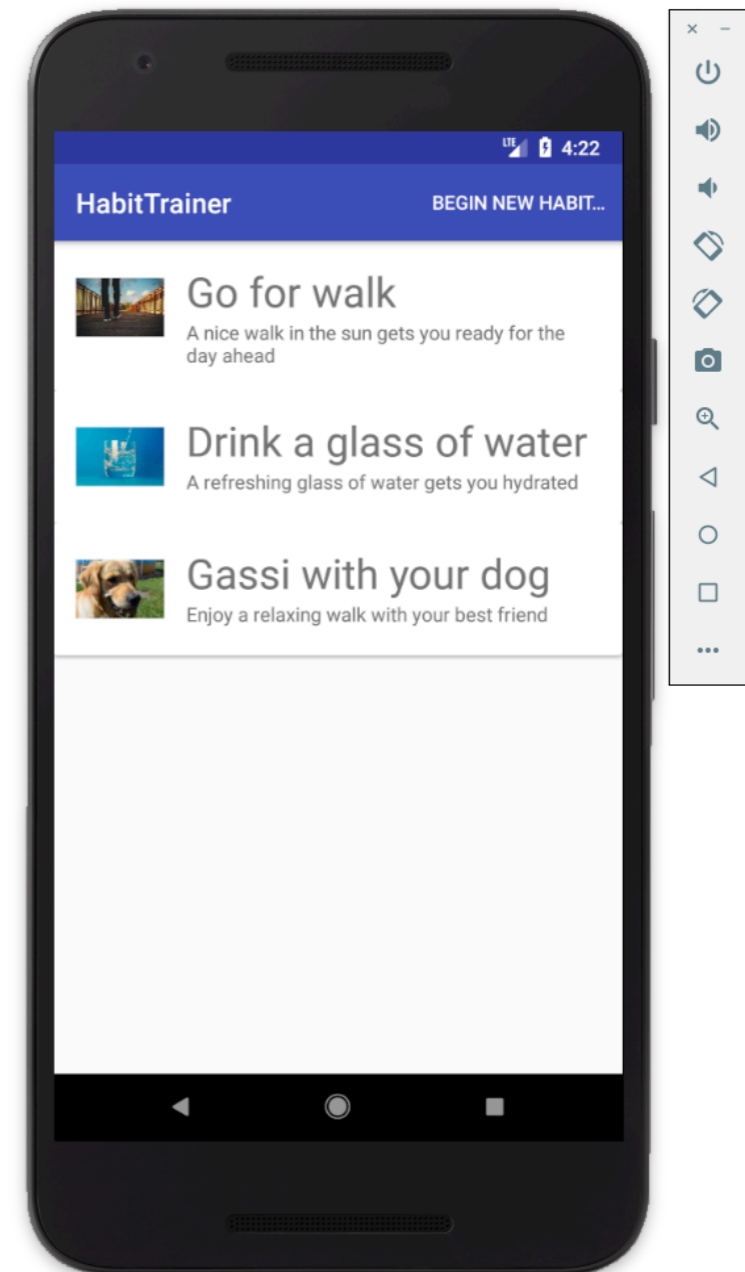
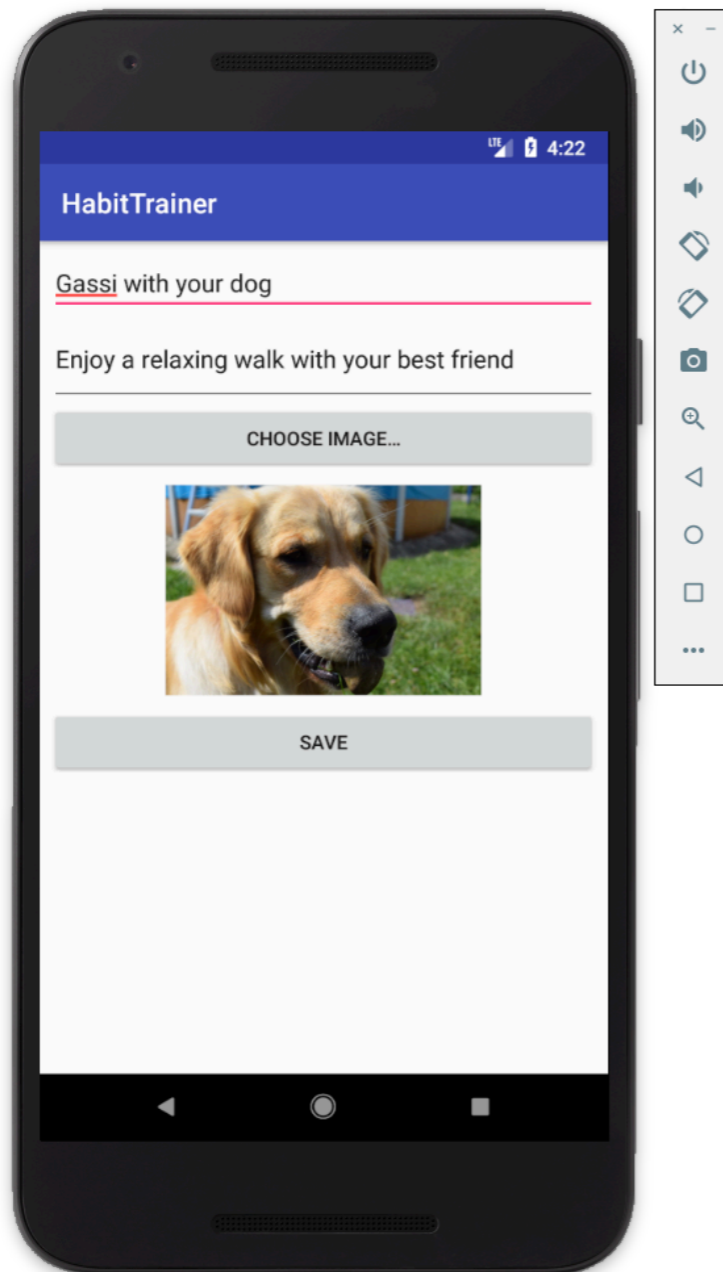
```
private fun parseHabitsFrom(cursor: Cursor): MutableList<Habit> {  
    val habits = mutableListOf<Habit>()  
    while (cursor.moveToNext()) {  
        val title = cursor.getString(HabitEntry.TITLE_COL)  
        val desc = cursor.getString(HabitEntry.DESCR_COL)  
        val bitmap = cursor.getBitmap(HabitEntry.IMAGE_COL)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

Kopieren von Files auf Device



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

Great - you did it



- <https://antoniroleiva.com/kotlin-awesome-tricks-for-android/>
- <https://github.com/petersommerhoff/kotlin-android>



Kotlin





Noch
Fragen?