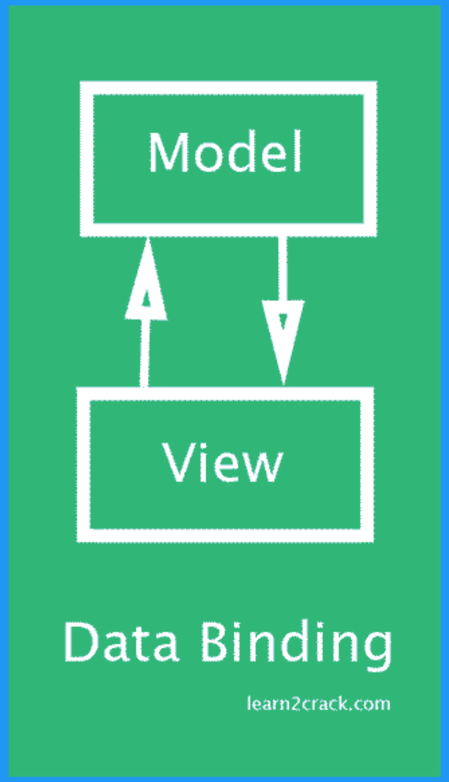


Android Data Binding

Was ist Data Binding?

Als Datenbindung (engl. Data Binding) bezeichnet man die automatische Weitergabe von Daten zwischen Objekten. Typischerweise werden Daten aus einem Datenobjekt an ein Steuerelement der Benutzeroberfläche weitergegeben. Aber auch zwischen Steuerelementen ist Datenbindung in einigen Frameworks möglich.

Beispiel

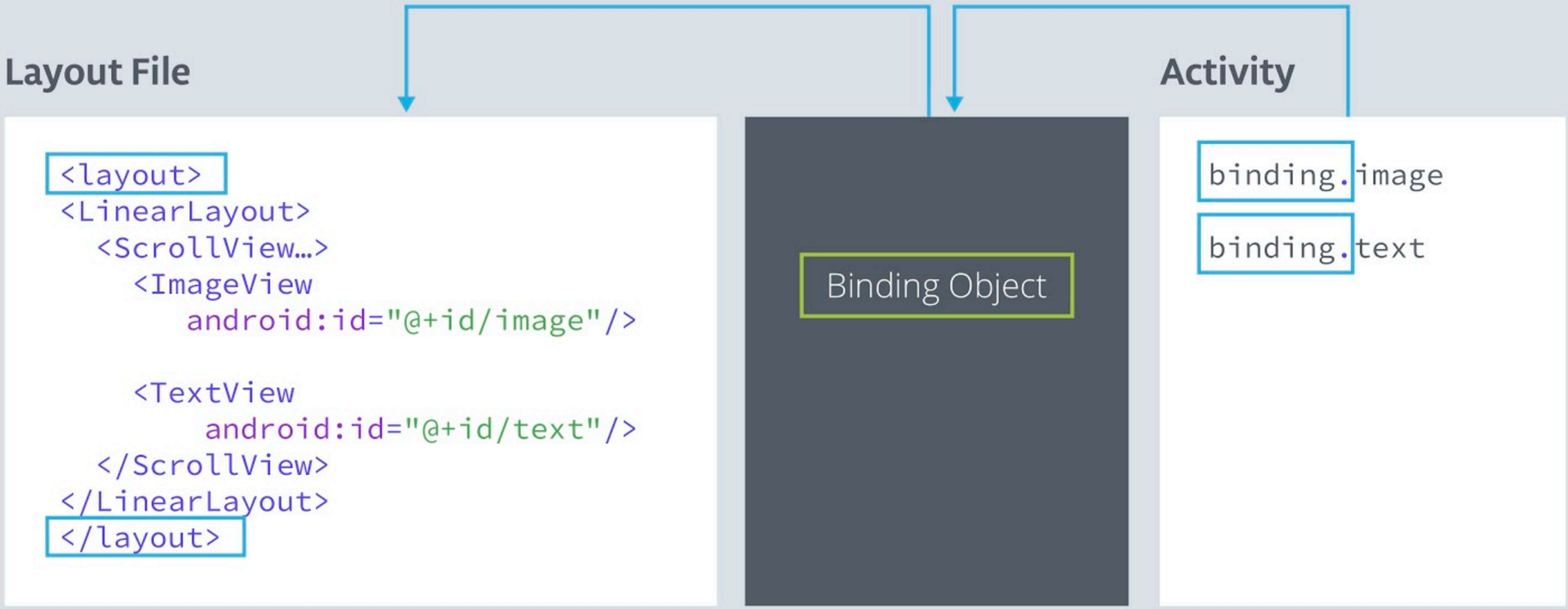
Wenn man eine Liste von Kunden auf dem Bildschirm in einer Tabelle ausgeben will, würde man klassischerweise eine Schleife über die Kundenliste ausführen (z.B. foreach) und dann für jedes Kundenelement eine Tabellenzeile mit den einzelnen Spalten erzeugen. Bei der Datenbindung entfällt die Schleife und die Erstellung der einzelnen Spalten. Vielmehr schreibt der Entwickler so etwas wie:

Steuerelement.DataSource = KundenListe

Das Steuerelement führt dann die Schleife selbst aus. In dem Steuerelement ist konfiguriert, welche Spalten es darstellen soll. Alternativ können einige Steuerelemente alle Eigenschaften des Kunden ausgeben, sofern es keine explizite Definition gibt.

https://www.it-visions.de/glossar/alle/6875/Data_Binding.aspx

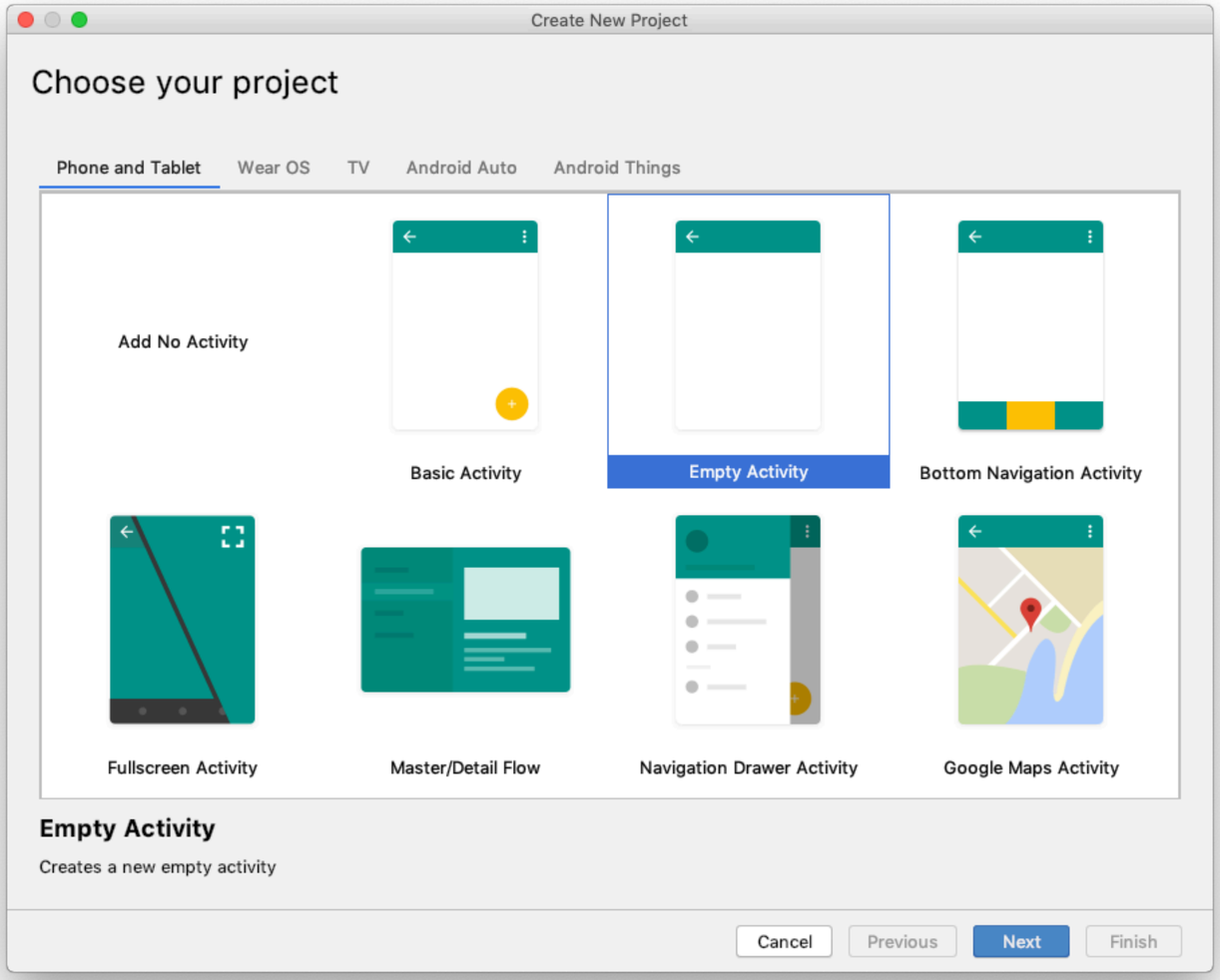
With Data Binding

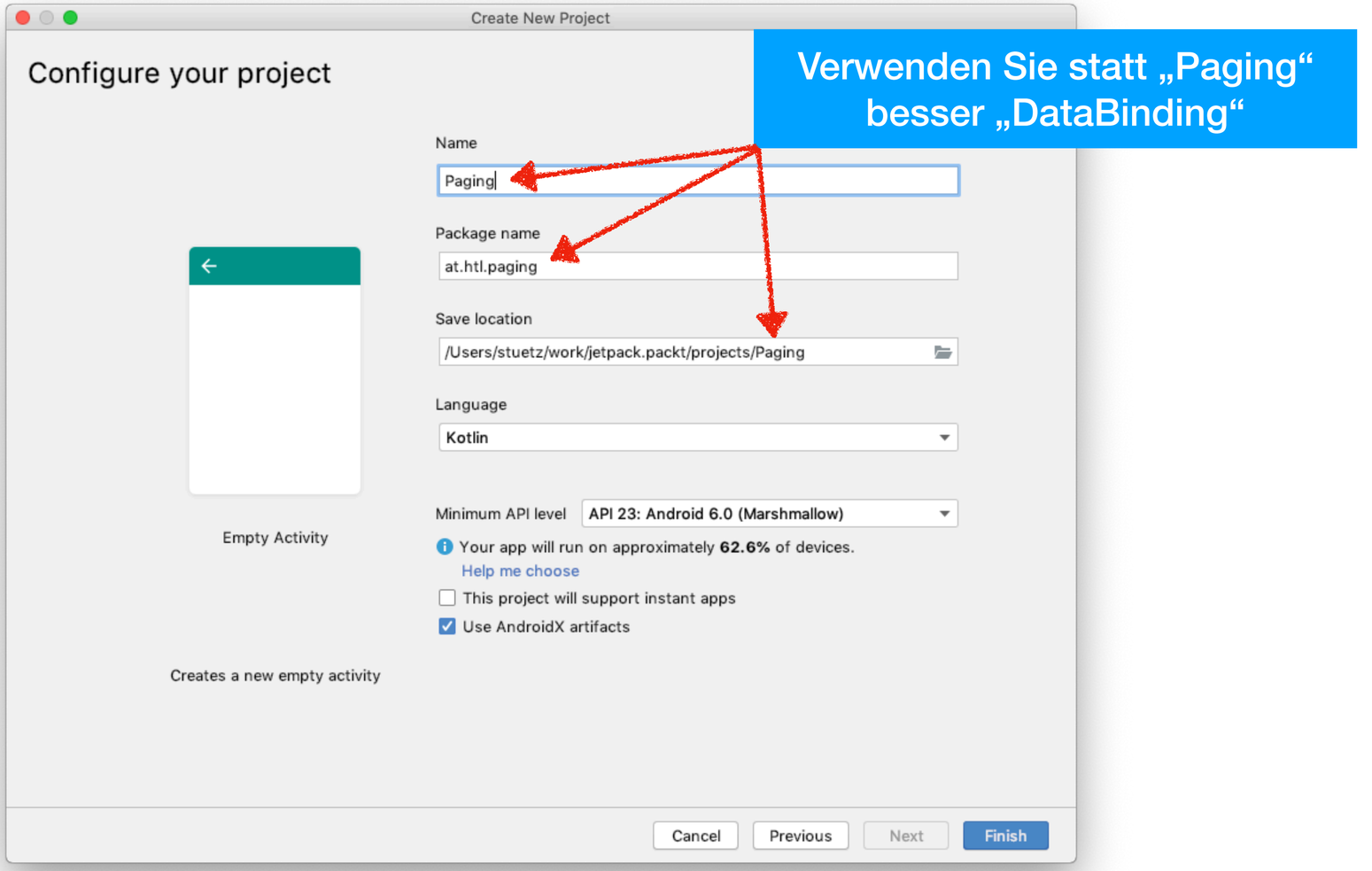


https://codelabs.developers.google.com/codelabs/kotlin-android-training-data-binding-basics/#0

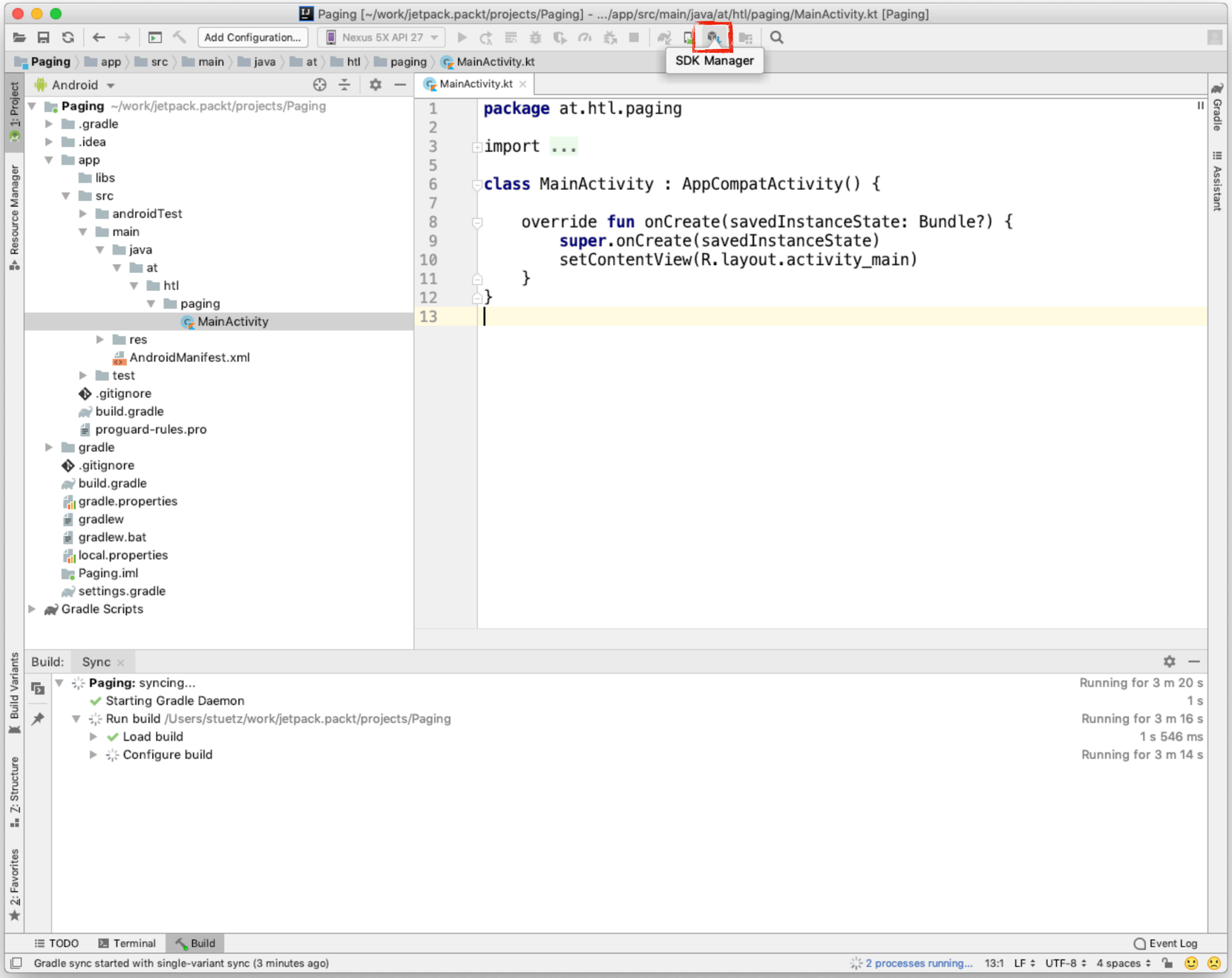
DataBinding konfigurieren

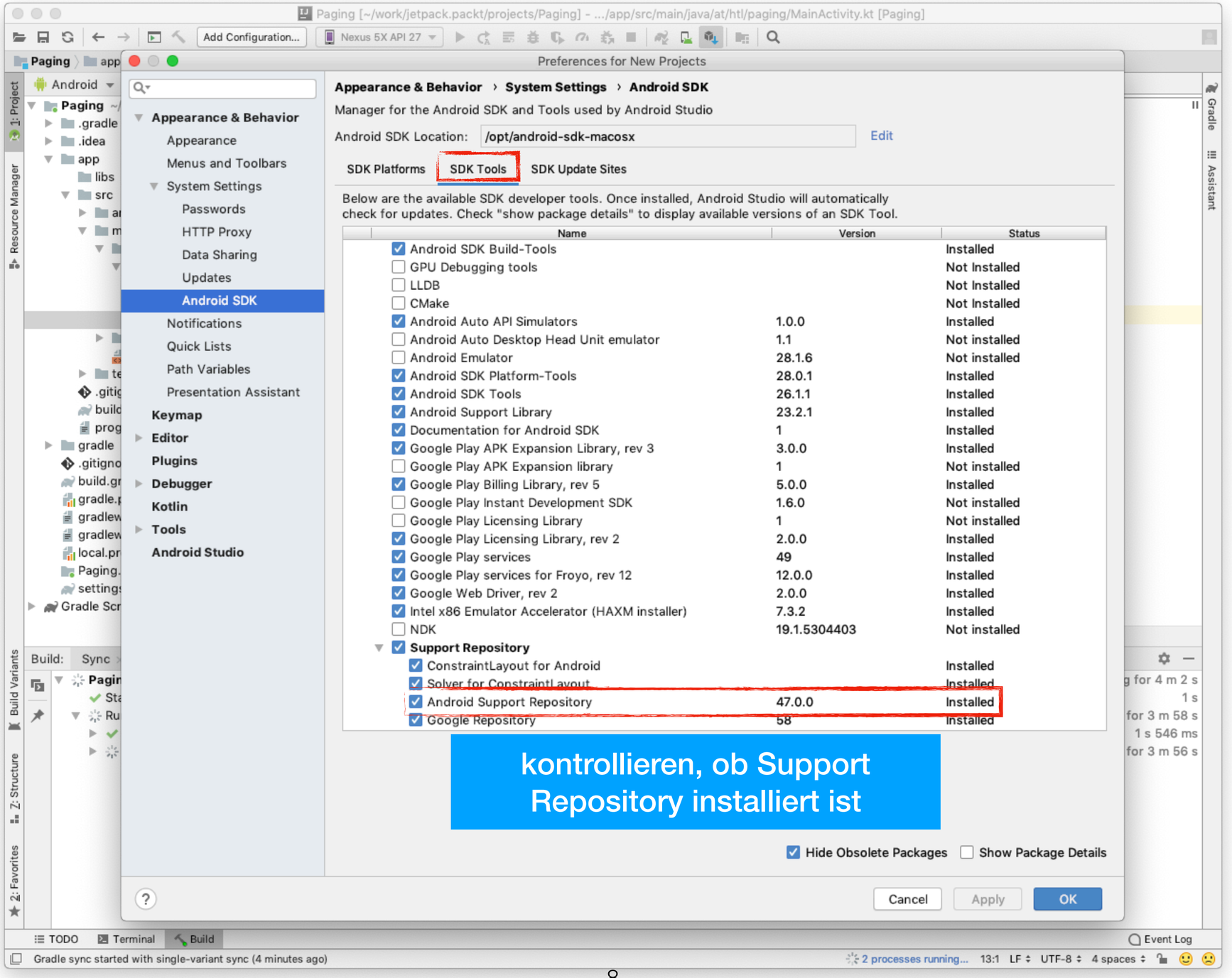
Neues Projekt erstellen



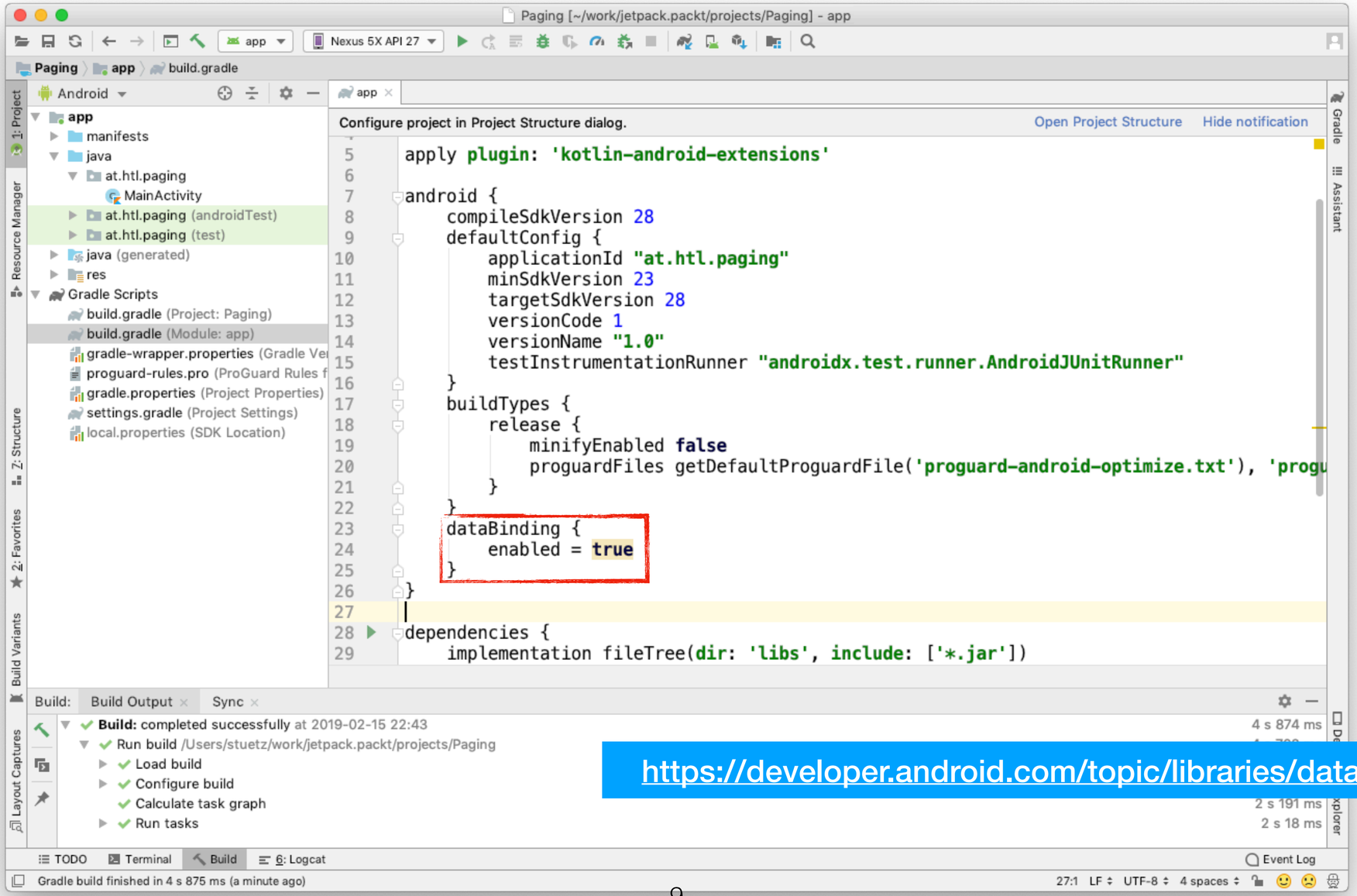


Verwenden Sie statt „Paging“ besser „DataBinding“


Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [Paging]
Add Configuration...
Nexus 5X API 27
SDK Manager
Paging app src main java at htl paging MainActivity.kt
Android
Paging ~/work/jetpack.packt/projects/Paging
.gradle
.idea
app
libs
src
androidTest
main
java
at
htl
paging
MainActivity
res
AndroidManifest.xml
test
.gitignore
build.gradle
proguard-rules.pro
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
Paging.iml
settings.gradle
Gradle Scripts
MainActivity.kt
1 package at.htl.paging
2
3 import ...
5
6 class MainActivity : AppCompatActivity() {
7
8 override fun onCreate(savedInstanceState: Bundle?) {
9 super.onCreate(savedInstanceState)
10 setContentView(R.layout.activity_main)
11 }
12 }
13
Build: Sync
Paging: syncing... Running for 3 m 20 s
Starting Gradle Daemon 1 s
Run build /Users/stuetz/work/jetpack.packt/projects/Paging Running for 3 m 16 s
Load build 1 s 546 ms
Configure build Running for 3 m 14 s
TODO
Terminal
Build
Event Log
Gradle sync started with single-variant sync (3 minutes ago)
2 processes running...
13:1
LF
UTF-8
4 spaces
1: Project
Resource Manager
Build Variants
Z: Structure
2: Favorites
Gradle
Assistant

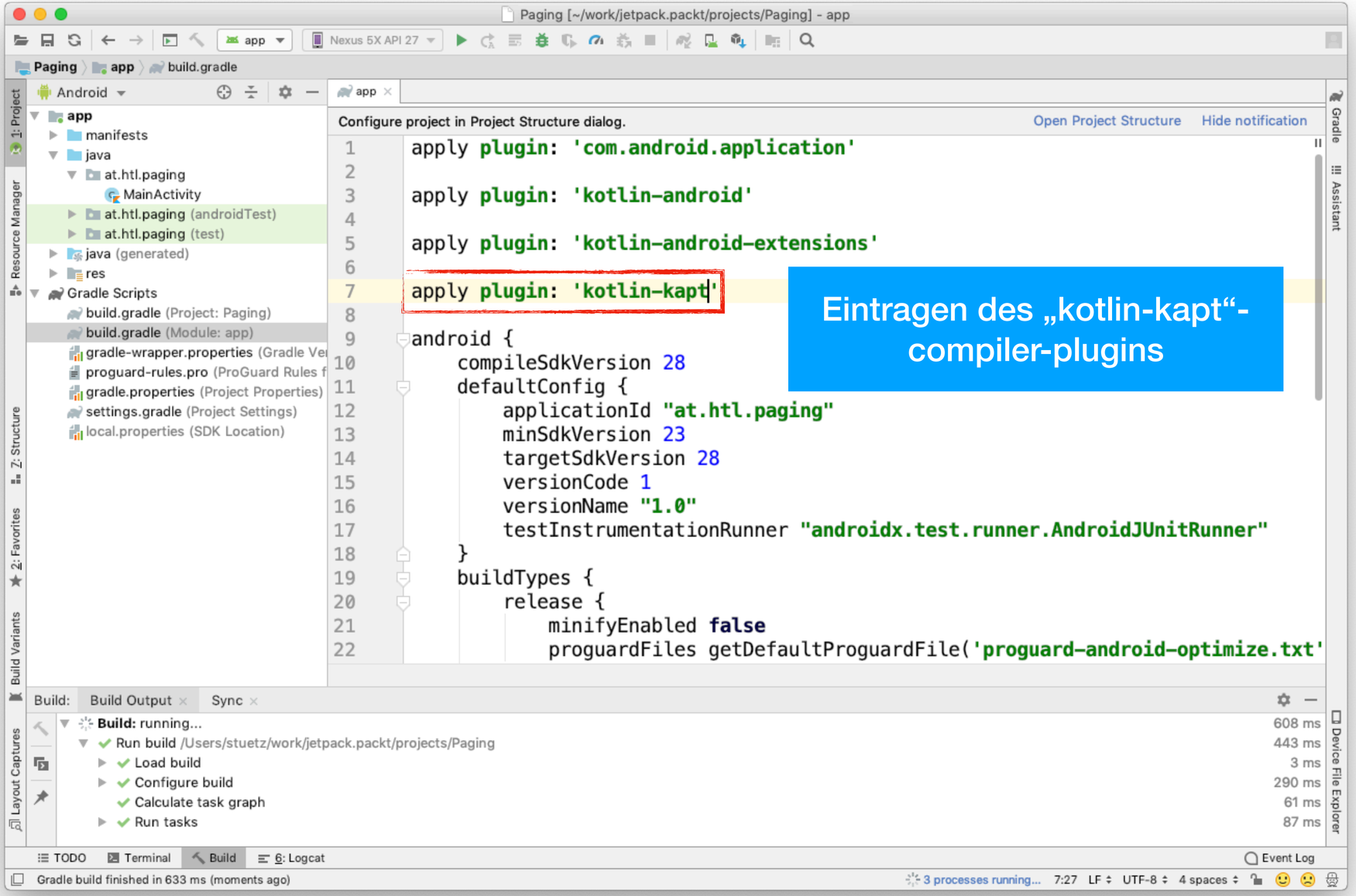

Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [Paging]
Preferences for New Projects
Appearance & Behavior › System Settings › Android SDK
Manager for the Android SDK and Tools used by Android Studio
Android SDK Location: /opt/android-sdk-macosx
Edit
SDK Platforms
SDK Tools
SDK Update Sites
Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.
Name
Version
Status
Android SDK Build-Tools Installed
GPU Debugging tools Not Installed
LLDB Not Installed
CMake Not Installed
Android Auto API Simulators 1.0.0 Installed
Android Auto Desktop Head Unit emulator 1.1 Not installed
Android Emulator 28.1.6 Not installed
Android SDK Platform-Tools 28.0.1 Installed
Android SDK Tools 26.1.1 Installed
Android Support Library 23.2.1 Installed
Documentation for Android SDK 1 Installed
Google Play APK Expansion Library, rev 3 3.0.0 Installed
Google Play APK Expansion library 1 Not installed
Google Play Billing Library, rev 5 5.0.0 Installed
Google Play Instant Development SDK 1.6.0 Not installed
Google Play Licensing Library 1 Not installed
Google Play Licensing Library, rev 2 2.0.0 Installed
Google Play services 49 Installed
Google Play services for Froyo, rev 12 12.0.0 Installed
Google Web Driver, rev 2 2.0.0 Installed
Intel x86 Emulator Accelerator (HAXM installer) 7.3.2 Installed
NDK 19.1.5304403 Not installed
Support Repository
ConstraintLayout for Android Installed
Solver for ConstraintLayout Installed
Android Support Repository 47.0.0 Installed
Google Repository 58 Installed
kontrollieren, ob Support Repository installiert ist
Hide Obsolete Packages
Show Package Details
Cancel
Apply
OK
Appearance & Behavior
Appearance
Menus and Toolbars
System Settings
Passwords
HTTP Proxy
Data Sharing
Updates
Android SDK
Notifications
Quick Lists
Path Variables
Presentation Assistant
Keymap
Editor
Plugins
Debugger
Kotlin
Tools
Android Studio
Gradle sync started with single-variant sync (4 minutes ago)

dataBinding in build.xml eintragen

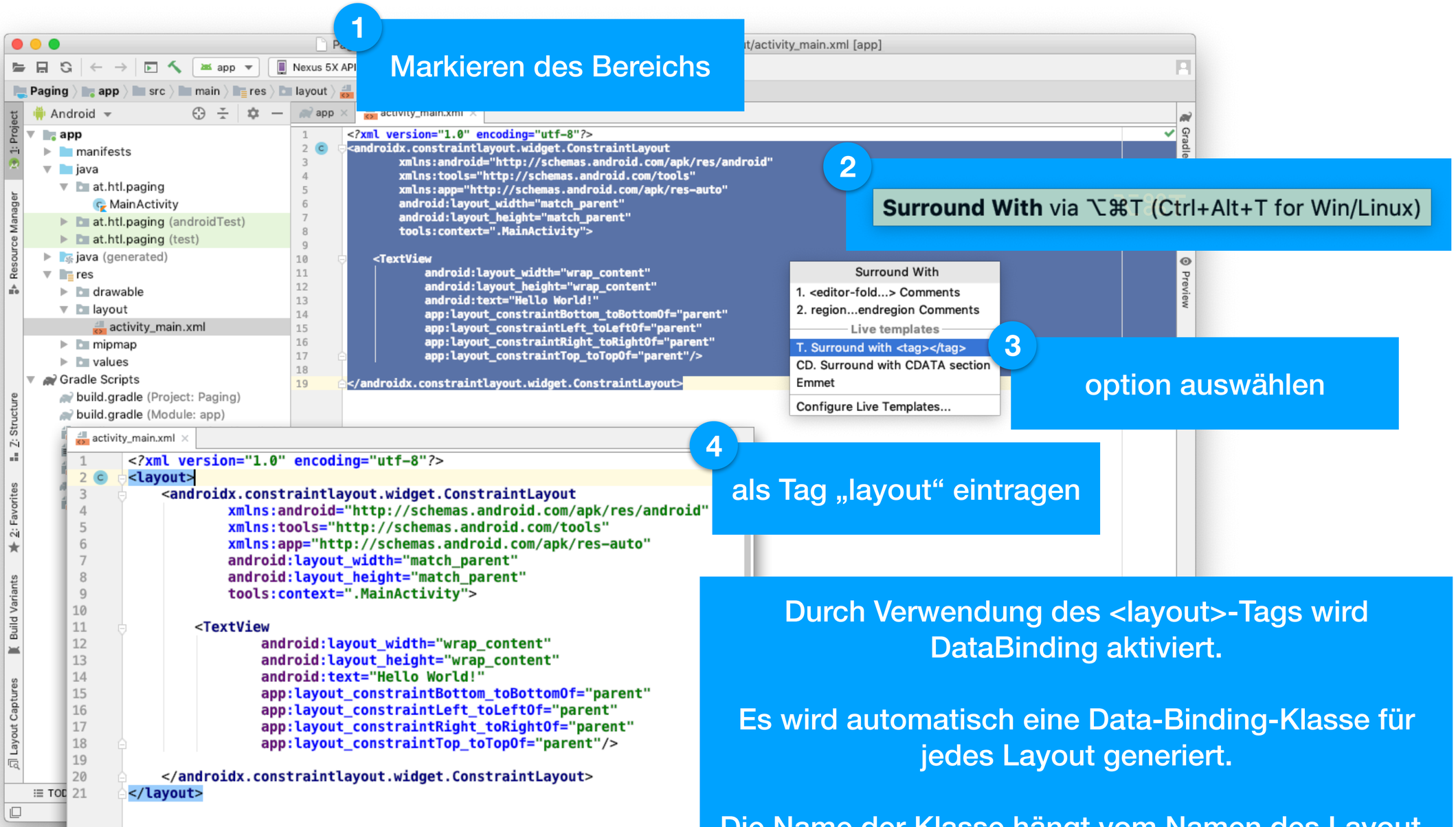


https://developer.android.com/topic/libraries/data-binding/start

Layouts and Binding Expressions

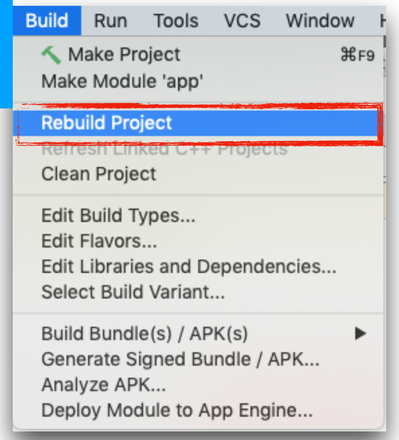


https://kotlinlang.org/docs/reference/kapt.html

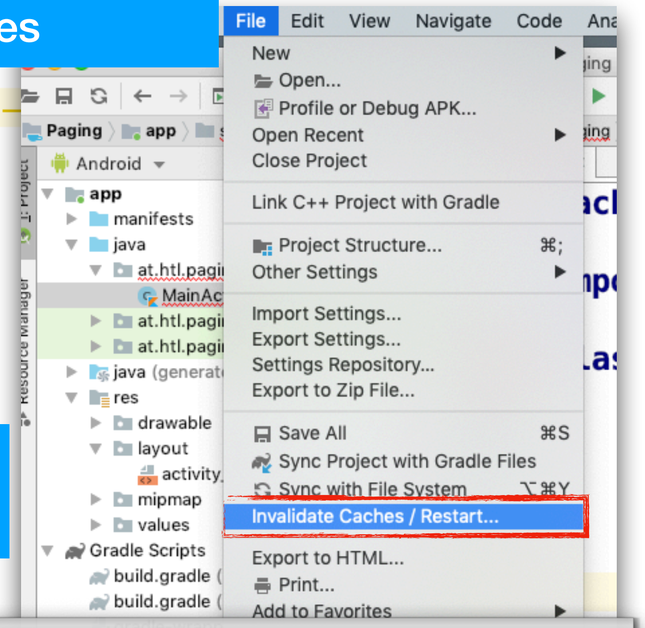

1
Markieren des Bereichs
2
Surround With via ⌥⌘T (Ctrl+Alt+T for Win/Linux)
Surround With
1. <editor-fold...> Comments
2. region...endregion Comments
Live templates
T. Surround with <tag></tag>
CD. Surround with CDATA section
Emmet
Configure Live Templates...
3
option auswählen
4
als Tag „layout“ eintragen
Durch Verwendung des <layout>-Tags wird DataBinding aktiviert.
Es wird automatisch eine Data-Binding-Klasse für jedes Layout generiert.
Die Name der Klasse hängt vom Namen des Layout-Files ab. In unserem Fall wird die DataBinding-Klasse „ActivityMainBinding“

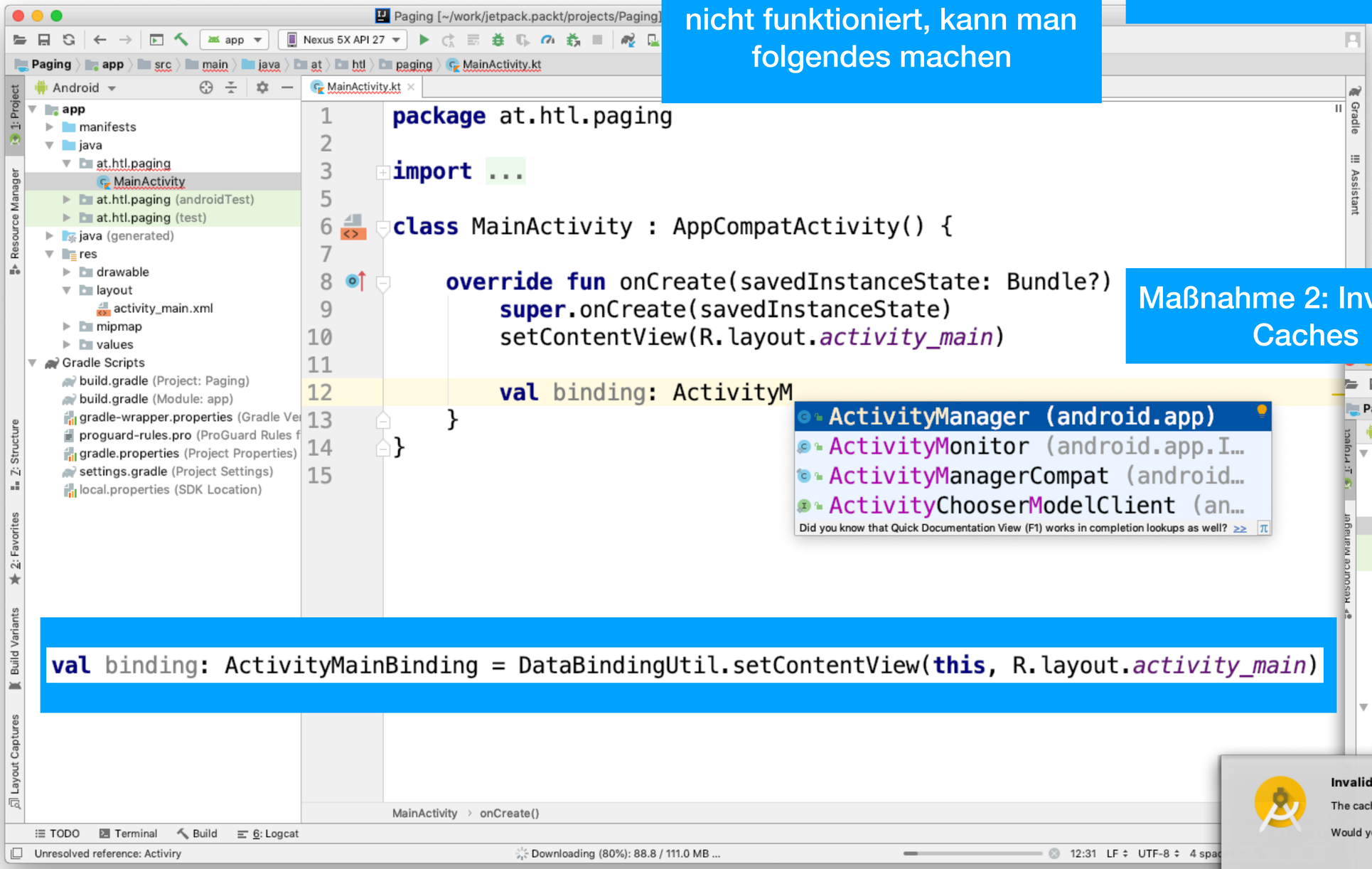
Falls das AutoComplete noch nicht funktioniert, kann man folgendes machen

Maßnahme 1: Rebuild

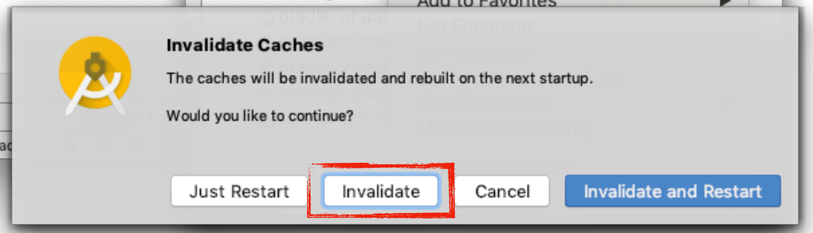


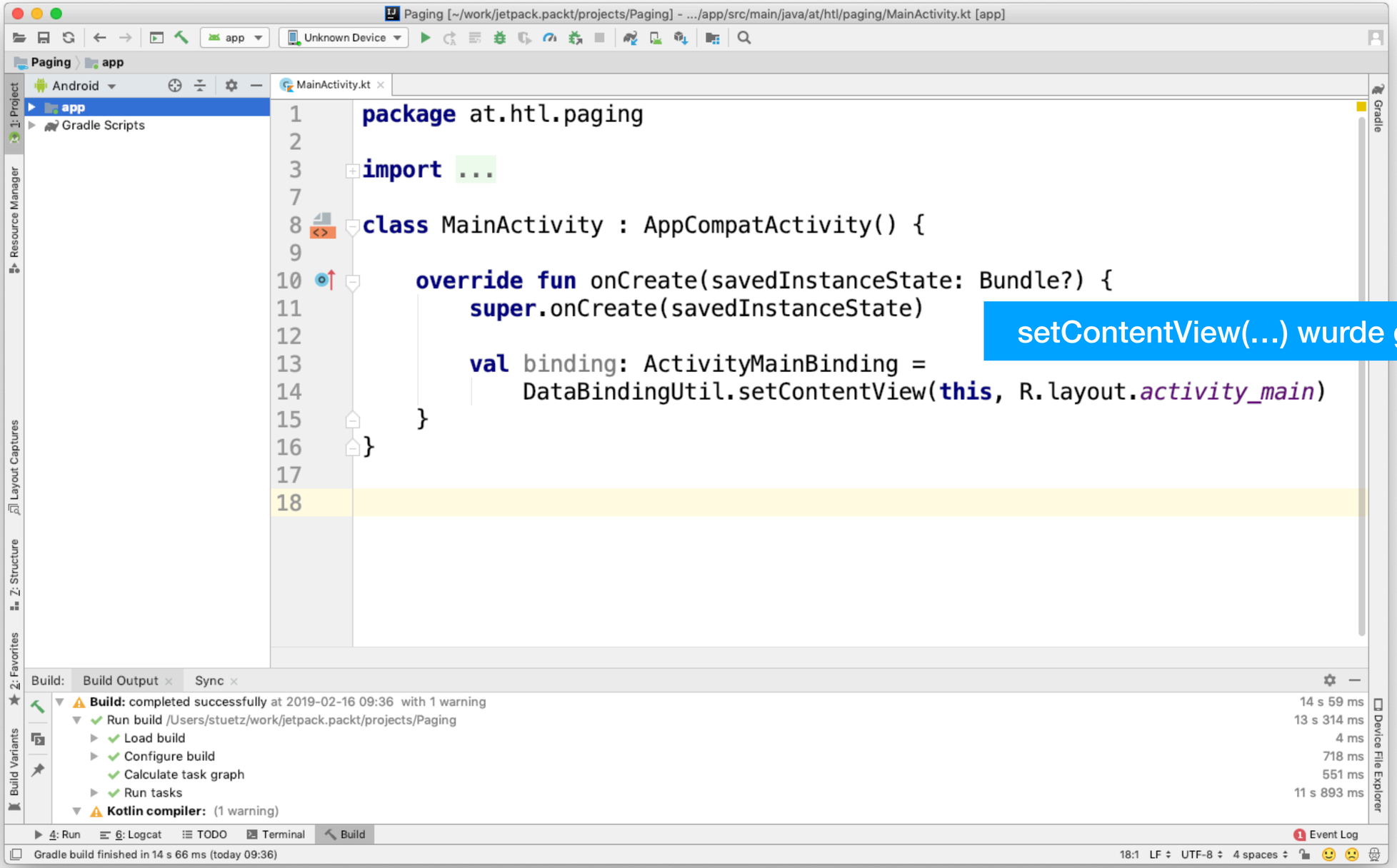
Maßnahme 2: Invalidate Caches



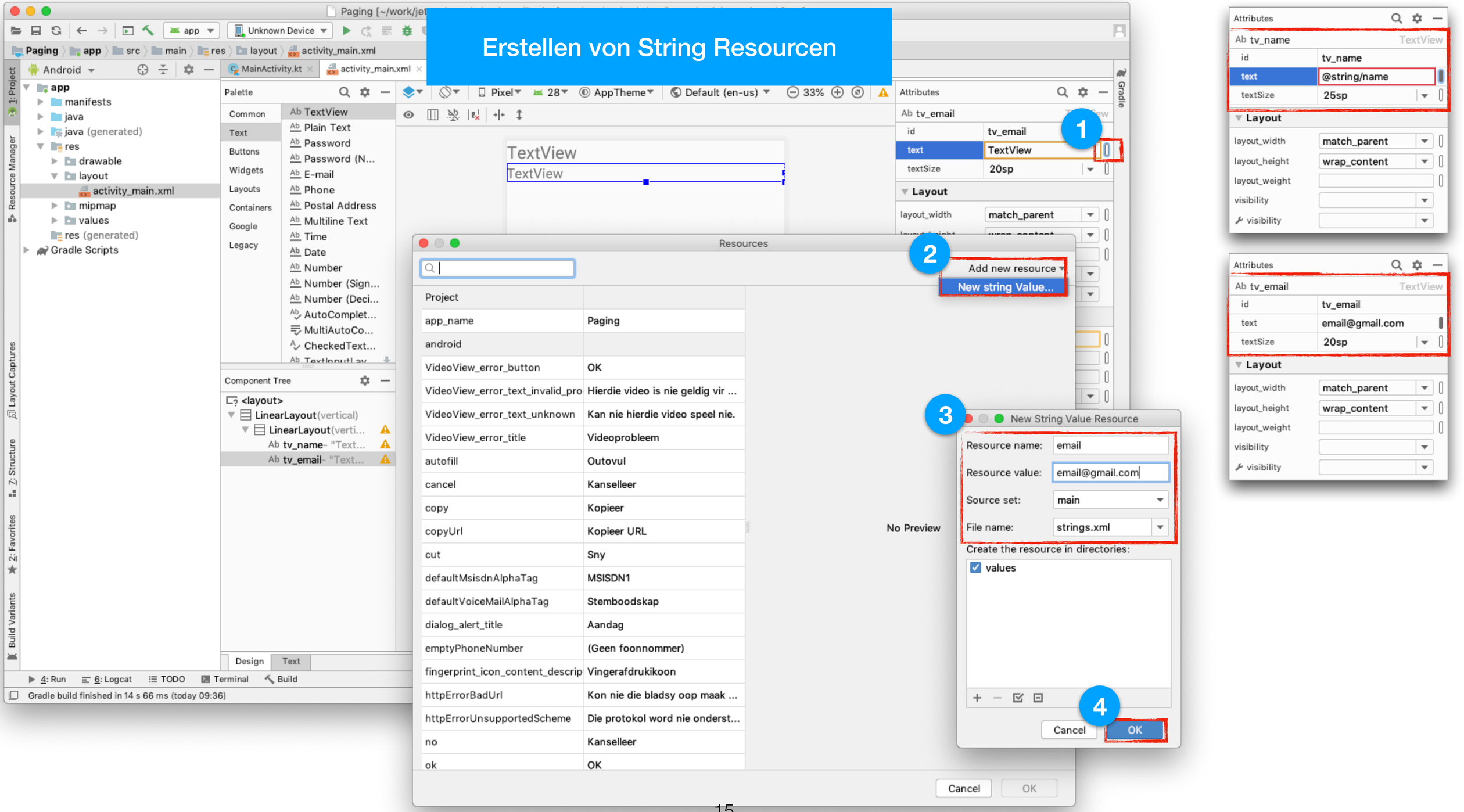


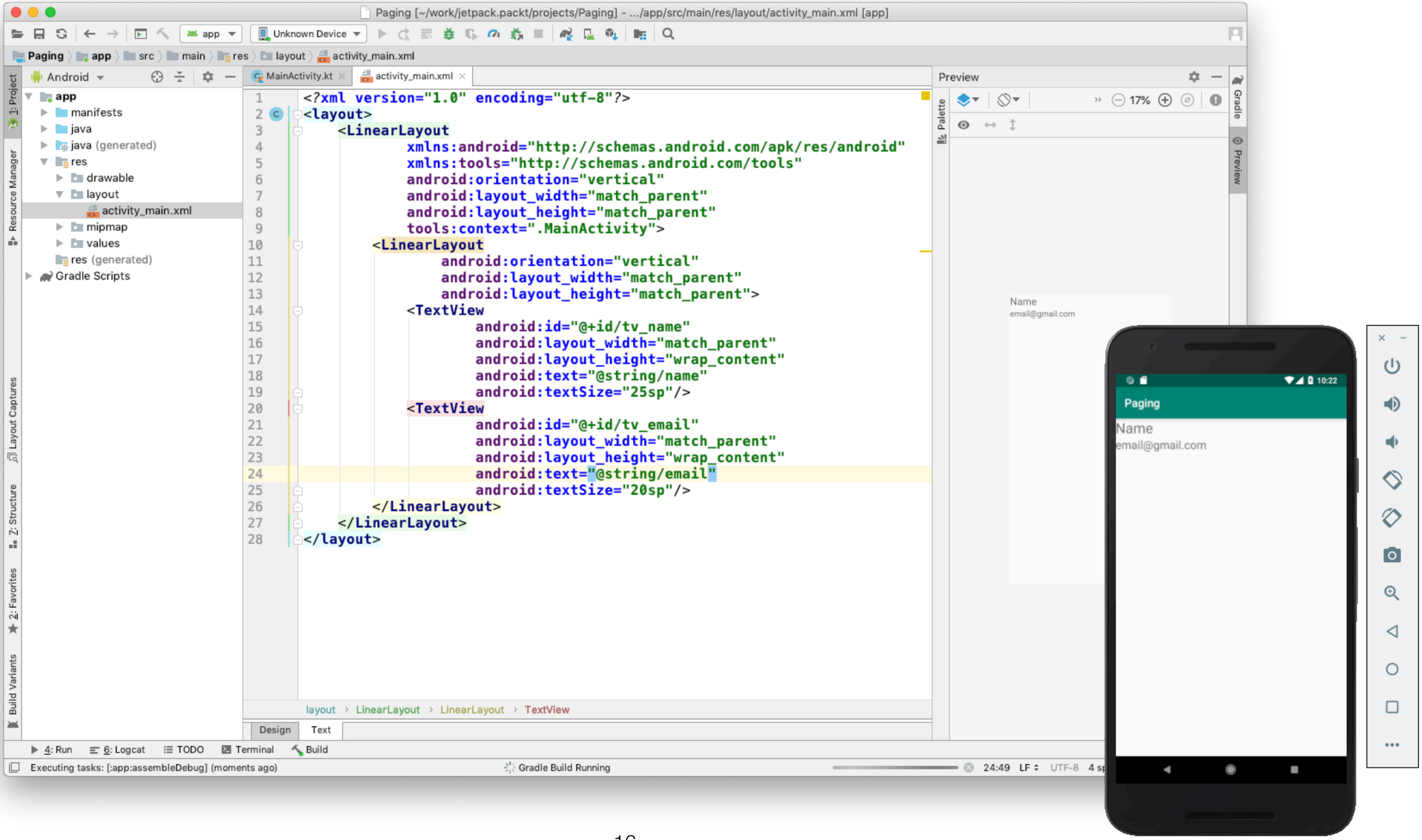

```
val binding: ActivityMainBinding = DataBindingUtil.setContentView(this, R.layout.activity_main)
```

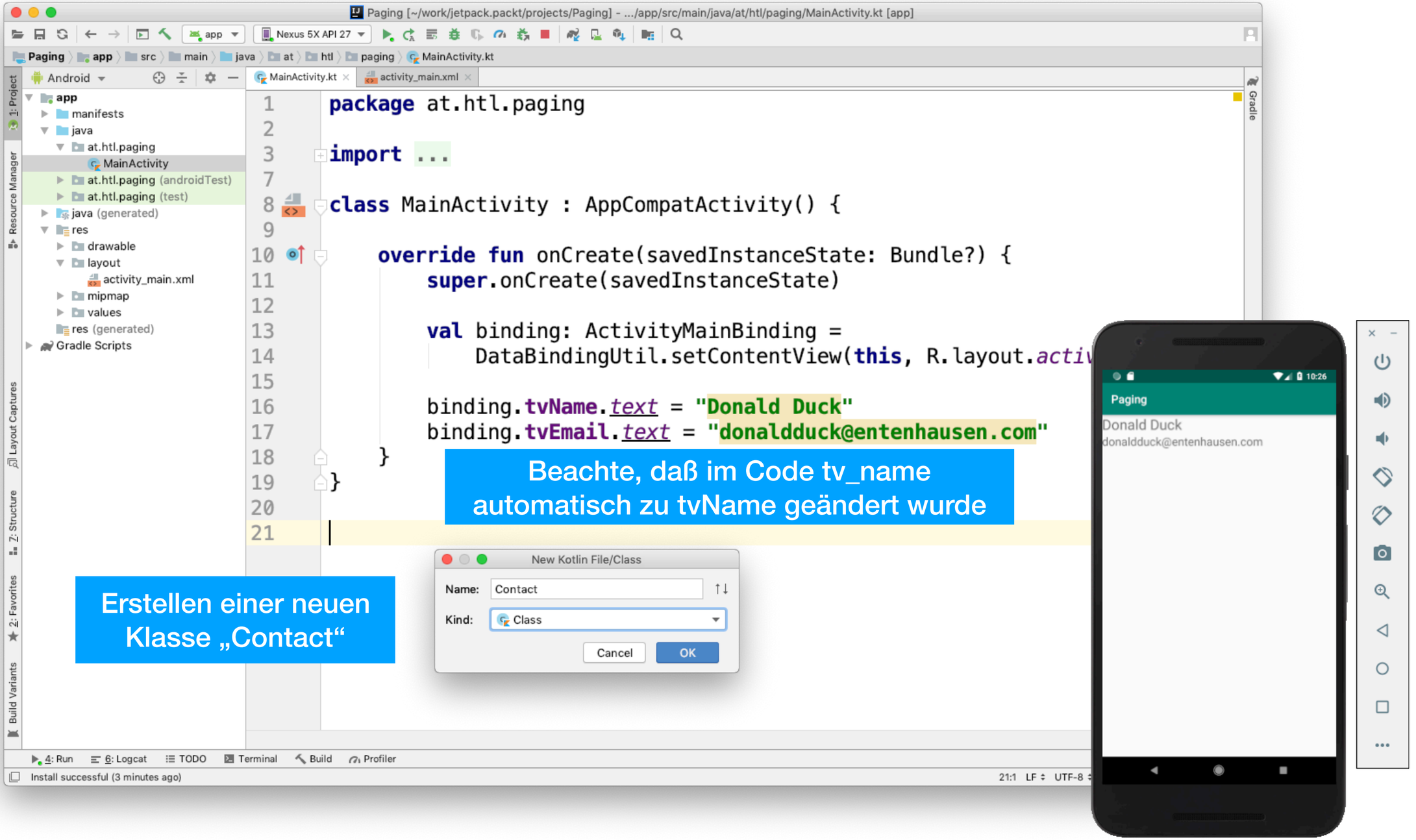


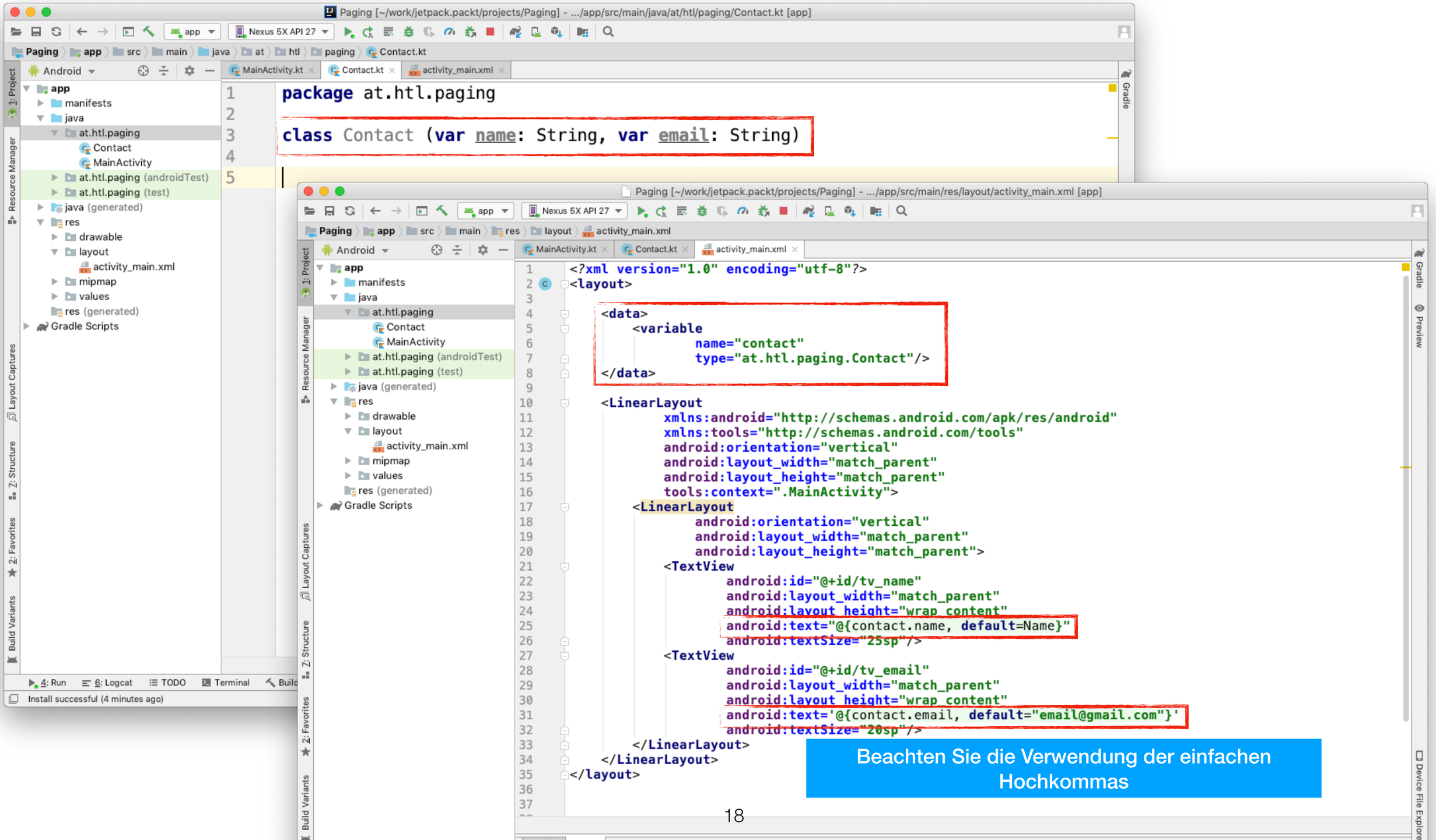

Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [app]
app
Unknown Device
Paging
app
Android
app
Gradle Scripts
MainActivity.kt
1 package at.htl.paging
2
3 import ...
7
8 class MainActivity : AppCompatActivity() {
9
10 override fun onCreate(savedInstanceState: Bundle?) {
11 super.onCreate(savedInstanceState)
12
13 val binding: ActivityMainBinding =
14 DataBindingUtil.setContentView(this, R.layout.activity_main)
15 }
16 }
17
18
setContentView(...) wurde gelöscht
Build:
Build Output
Sync
Build: completed successfully at 2019-02-16 09:36 with 1 warning 14 s 59 ms
Run build /Users/stuetz/work/jetpack.packt/projects/Paging 13 s 314 ms
Load build 4 ms
Configure build 718 ms
Calculate task graph 551 ms
Run tasks 11 s 893 ms
Kotlin compiler: (1 warning)
4: Run
6: Logcat
TODO
Terminal
Build
Event Log
Gradle build finished in 14 s 66 ms (today 09:36)
18:1 LF UTF-8 4 spaces
1: Project
Resource Manager
Layout Captures
Z: Structure
2: Favorites
Build Variants
Gradle
Device File Explorer

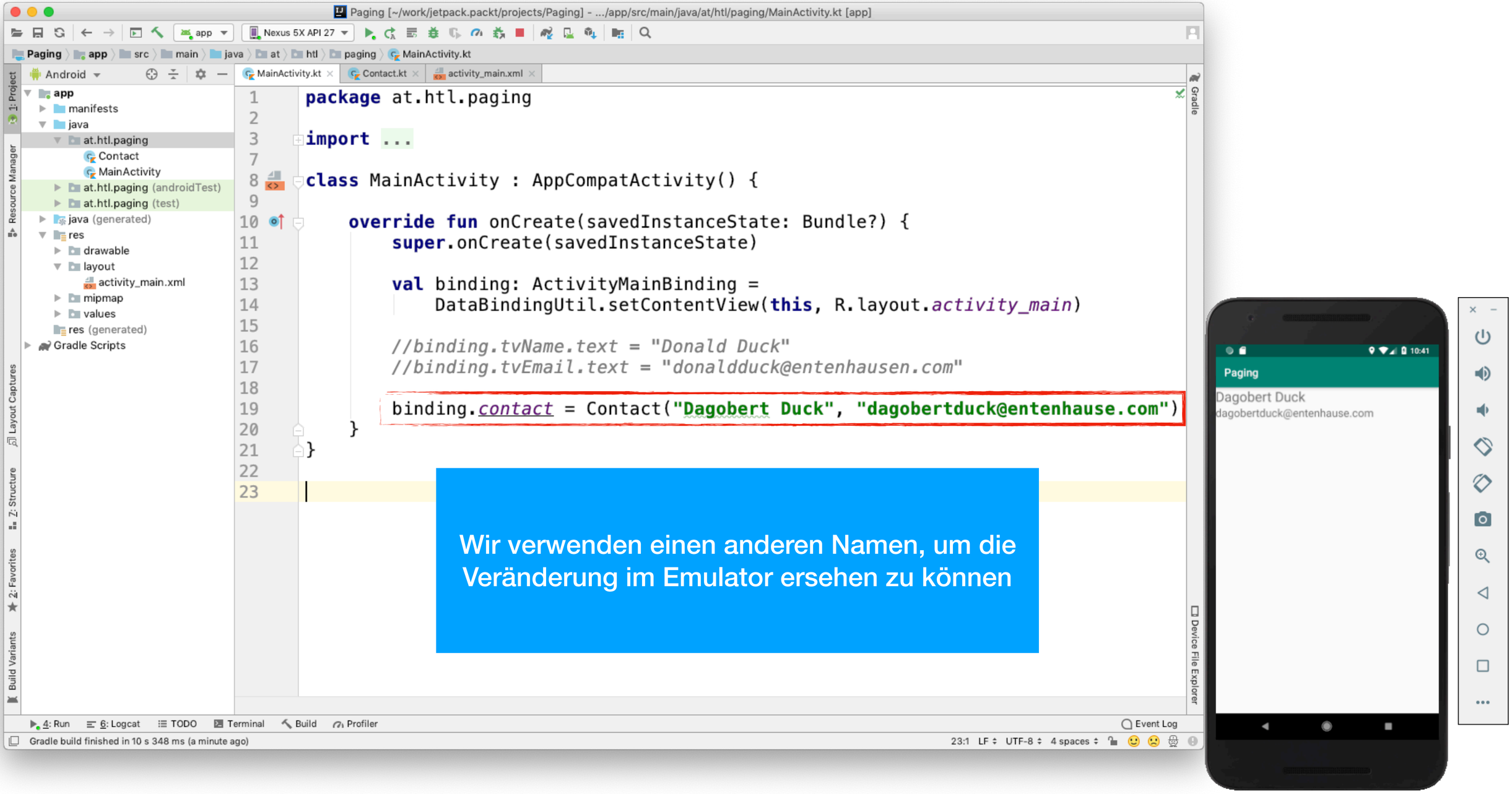
Erstellen von String Resourcen

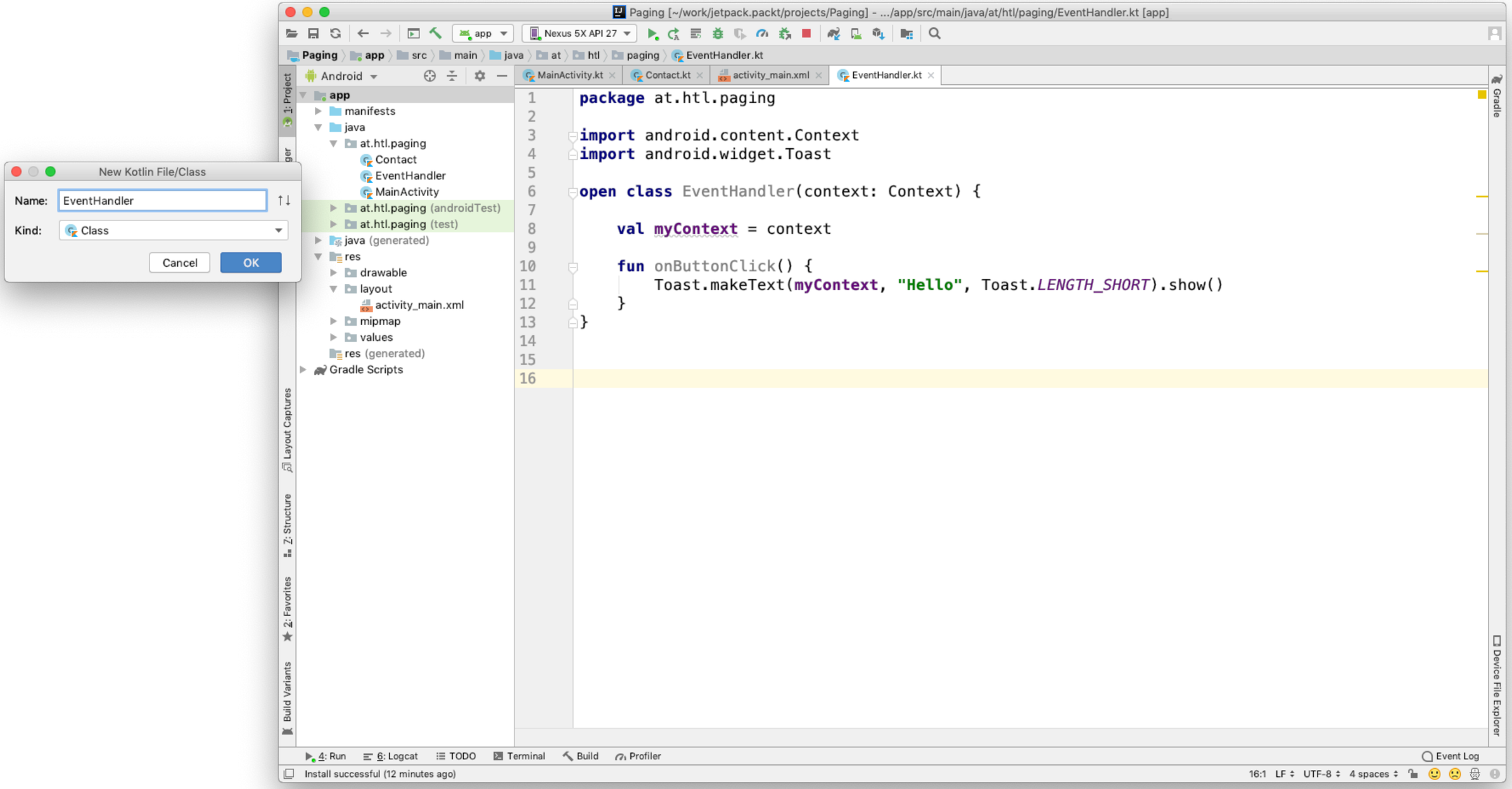



Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/res/layout/activity_main.xml [app]
Paging app src main res layout activity_main.xml
Android
app
manifests
java
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
res (generated)
Gradle Scripts
MainActivity.kt
activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<layout>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<LinearLayout
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:id="@+id/tv_name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/name"
android:textSize="25sp"/>
<TextView
android:id="@+id/tv_email"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/email"
android:textSize="20sp"/>
</LinearLayout>
</LinearLayout>
</layout>
layout › LinearLayout › LinearLayout › TextView
Design
Text
Preview
Name
email@gmail.com
Paging
Name
email@gmail.com
4: Run
6: Logcat
TODO
Terminal
Build
Executing tasks: [:app:assembleDebug] (moments ago)
Gradle Build Running
24:49
LF
UTF-8


Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [app]
app
Nexus 5X API 27
Paging app src main java at htl paging MainActivity.kt
Android
MainActivity.kt
activity_main.xml
app
manifests
java
at.htl.paging
MainActivity
at.htl.paging (androidTest)
at.htl.paging (test)
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
res (generated)
Gradle Scripts
package at.htl.paging
import ...
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
val binding: ActivityMainBinding =
DataBindingUtil.setContentView(this, R.layout.activ
binding.tvName.text = "Donald Duck"
binding.tvEmail.text = "donaldduck@entenhausen.com"
}
}
Beachte, daß im Code tv_name automatisch zu tvName geändert wurde
Erstellen einer neuen Klasse „Contact“
New Kotlin File/Class
Name: Contact
Kind: Class
Cancel
OK
Paging
Donald Duck
donaldduck@entenhausen.com
4: Run
6: Logcat
TODO
Terminal
Build
Profiler
Install successful (3 minutes ago)
21:1
LF
UTF-8

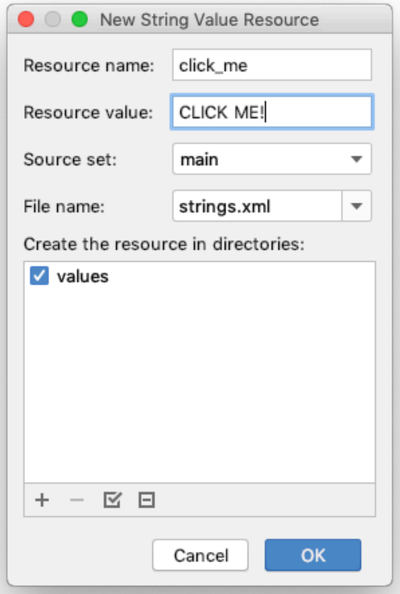

Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/Contact.kt [app]
Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/res/layout/activity_main.xml [app]
package at.htl.paging
class Contact (var name: String, var email: String)
<?xml version="1.0" encoding="utf-8"?>
<layout>
<data>
<variable
name="contact"
type="at.htl.paging.Contact"/>
</data>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<LinearLayout
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:id="@+id/tv_name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@{contact.name, default=Name}"
android:textSize="25sp"/>
<TextView
android:id="@+id/tv_email"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text='@{contact.email, default="email@gmail.com"}'
android:textSize="20sp"/>
</LinearLayout>
</LinearLayout>
</layout>
Beachten Sie die Verwendung der einfachen Hochkommas


Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [app]
app
Nexus 5X API 27
Paging app src main java at htl paging MainActivity.kt
Android
app
manifests
java
at.htl.paging
Contact
MainActivity
at.htl.paging (androidTest)
at.htl.paging (test)
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
res (generated)
Gradle Scripts
MainActivity.kt
Contact.kt
activity_main.xml
package at.htl.paging
import ...
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
val binding: ActivityMainBinding =
DataBindingUtil.setContentView(this, R.layout.activity_main)
//binding.tvName.text = "Donald Duck"
//binding.tvEmail.text = "donaldduck@entenhausen.com"
binding.contact = Contact("Dagobert Duck", "dagobertduck@entenhause.com")
}
}
Wir verwenden einen anderen Namen, um die Veränderung im Emulator ersehen zu können
Paging
Dagobert Duck
dagobertduck@entenhause.com
4: Run
6: Logcat
TODO
Terminal
Build
Profiler
Event Log
Gradle build finished in 10 s 348 ms (a minute ago)

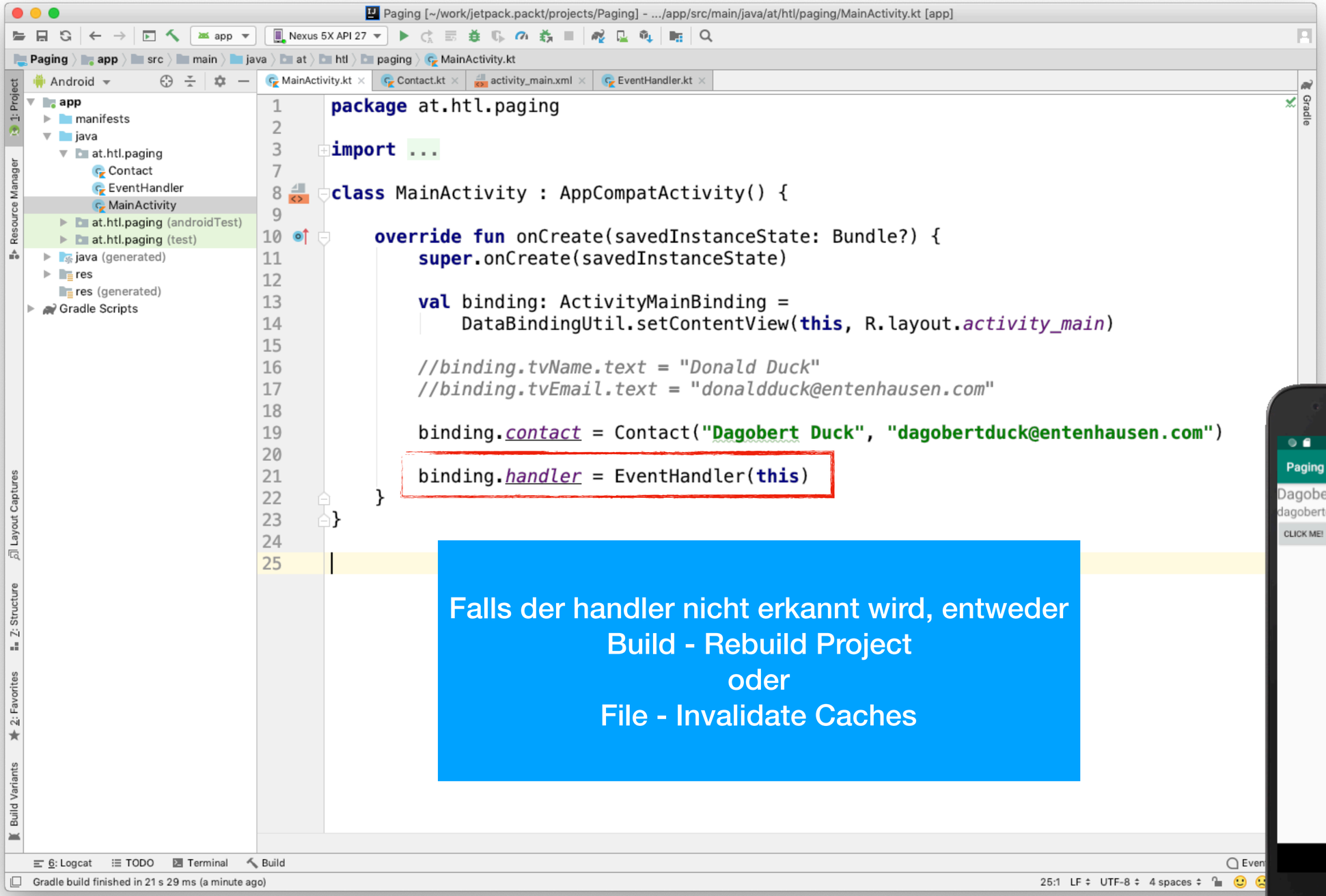

New Kotlin File/Class
Name: EventHandler
Kind: Class
Cancel
OK
Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/EventHandler.kt [app]
app
Nexus 5X API 27
Paging app src main java at htl paging EventHandler.kt
Android
app
manifests
java
at.htl.paging
Contact
EventHandler
MainActivity
at.htl.paging (androidTest)
at.htl.paging (test)
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
res (generated)
Gradle Scripts
MainActivity.kt
Contact.kt
activity_main.xml
EventHandler.kt
package at.htl.paging
import android.content.Context
import android.widget.Toast
open class EventHandler(context: Context) {
val myContext = context
fun onButtonClick() {
Toast.makeText(myContext, "Hello", Toast.LENGTH_SHORT).show()
}
}
4: Run
6: Logcat
TODO
Terminal
Build
Profiler
Event Log
Install successful (12 minutes ago)
16:1
LF
UTF-8
4 spaces

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

    <data>
        <variable
                name="contact"
                type="at.htl.paging.Contact"/>
        <variable
                name="handler"
                type="at.htl.paging.EventHandler"/>
    </data>

    <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MainActivity">
        <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            <TextView
                    android:id="@+id/tv_name"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@{contact.name, default=Name}"
                    android:textSize="25sp"/>
            <TextView
                    android:id="@+id/tv_email"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text='@{contact.email, default="email@gmail.com"}'
                    android:textSize="20sp"/>
            <Button
                    android:id="@+id/btn_click"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="CLICK ME!"
                    android:onClick="@{() -> handler.onButtonClick()}"/>
        </LinearLayout>
    </LinearLayout>
</layout>
```

Der EventHandler wird als Variable eingetragen



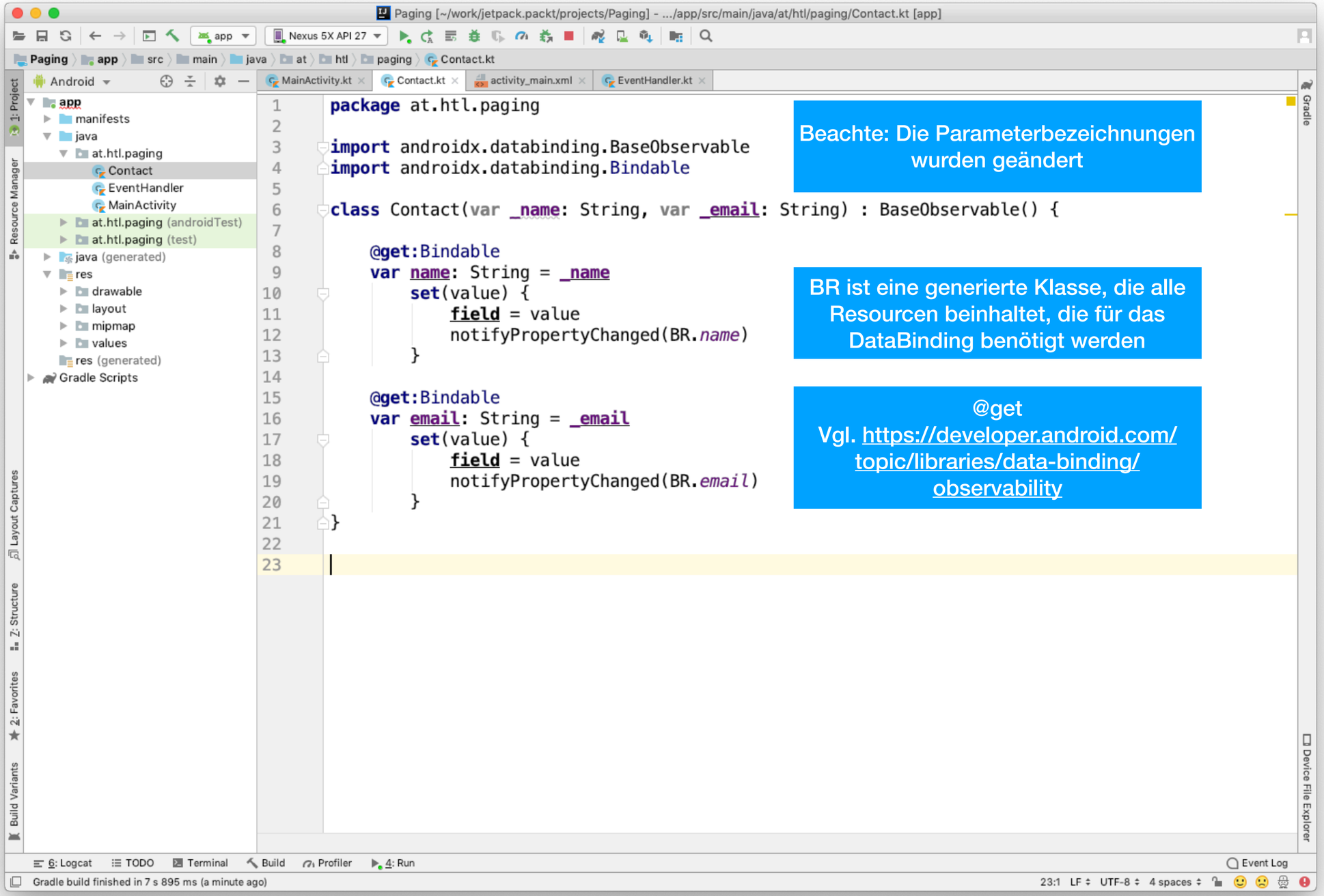
Ebenso die onClick-Methode


Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/MainActivity.kt [app]
Paging › app › src › main › java › at › htl › paging › MainActivity.kt
Android
app
manifests
java
at.htl.paging
Contact
EventHandler
MainActivity
at.htl.paging (androidTest)
at.htl.paging (test)
java (generated)
res
res (generated)
Gradle Scripts
MainActivity.kt
Contact.kt
activity_main.xml
EventHandler.kt
package at.htl.paging
import ...
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
val binding: ActivityMainBinding =
DataBindingUtil.setContentView(this, R.layout.activity_main)
//binding.tvName.text = "Donald Duck"
//binding.tvEmail.text = "donaldduck@entenhausen.com"
binding.contact = Contact("Dagobert Duck", "dagobertduck@entenhausen.com")
binding.handler = EventHandler(this)
}
}
Falls der handler nicht erkannt wird, entweder
Build - Rebuild Project
oder
File - Invalidate Caches
Paging
Dagobert Duck
dagobertduck@entenhausen.com
CLICK ME!
Hello
6: Logcat
TODO
Terminal
Build
Gradle build finished in 21 s 29 ms (a minute ago)

Zusammenfassung

- In gradle.build dataBinding-enabled und kotlin-kapt-plugin eintragen
- <layout> - Tag in layout-Resource eintragen
- Datenklassen und EventHandler-Klassen erstellen
- Datenklassen und EventHandler als <variable> in layout-Resource eintragen
- In der layout-Resource den Zugriff auf die Variablen mit #{…} eintragen
- In der zugehörigen Activity werden nun die Variablen (Datenobjekte, Handler, …) mit Werten initialisiert

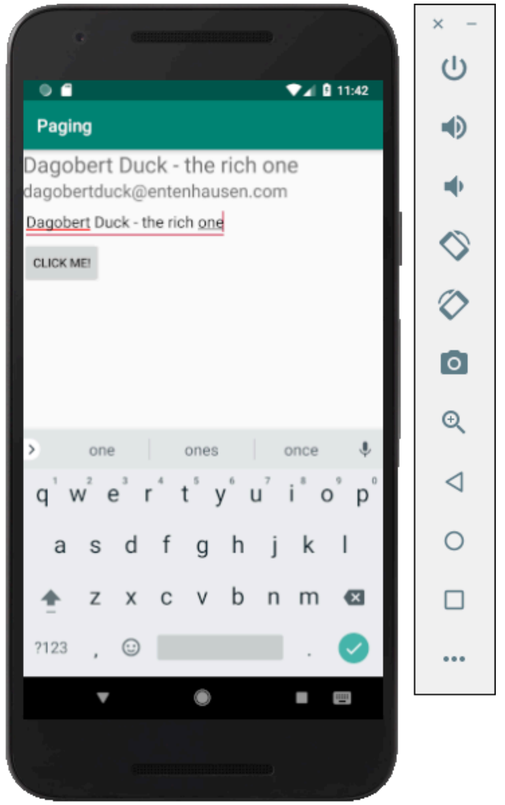
Observable Data Objects


Paging [~/work/jetpack.packt/projects/Paging] - .../app/src/main/java/at/htl/paging/Contact.kt [app]
Paging app src main java at htl paging Contact.kt
Android
app
manifests
java
at.htl.paging
Contact
EventHandler
MainActivity
at.htl.paging (androidTest)
at.htl.paging (test)
java (generated)
res
drawable
layout
mipmap
values
res (generated)
Gradle Scripts
MainActivity.kt
Contact.kt
activity_main.xml
EventHandler.kt
1 package at.htl.paging
2
3 import androidx.databinding.BaseObservable
4 import androidx.databinding.Bindable
5
6 class Contact(var _name: String, var _email: String) : BaseObservable() {
7
8 @get:Bindable
9 var name: String = _name
10 set(value) {
11 field = value
12 notifyPropertyChanged(BR.name)
13 }
14
15 @get:Bindable
16 var email: String = _email
17 set(value) {
18 field = value
19 notifyPropertyChanged(BR.email)
20 }
21 }
22
23
Beachte: Die Parameterbezeichnungen wurden geändert
BR ist eine generierte Klasse, die alle Resourcen beinhaltet, die für das DataBinding benötigt werden
@get
Vgl. https://developer.android.com/topic/libraries/data-binding/observability
6: Logcat
TODO
Terminal
Build
Profiler
4: Run
Event Log
Gradle build finished in 7 s 895 ms (a minute ago)
23:1 LF UTF-8 4 spaces

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

    <data>
        <variable
                name="contact"
                type="at.htl.paging.Contact"/>
        <variable
                name="handler"
                type="at.htl.paging.EventHandler"/>
    </data>

    <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MainActivity">
        <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent">
            <TextView
                    android:id="@+id/tv_name"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@{contact.name, default=Name}"
                    android:textSize="25sp"/>
            <TextView
                    android:id="@+id/tv_email"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text='@{contact.email, default="email@gmail.com"}'
                    android:textSize="20sp"/>
            <EditText
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:id="@+id/et_name"
                    android:inputType="text"
                    android:text="@={contact.name, default=Name}"/>
            <Button
                    android:id="@+id/btn_click"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="CLICK ME!"
                    android:onClick="@{() -> handler.onButtonClick()}"/>
        </LinearLayout>
    </LinearLayout>
</layout>
```

```xml
<EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/et_name"
        android:inputType="text"
        android:text="@={contact.name, default=Name}"/>
```

Das =-Zeichen bedeutet, daß der eingegebene Text akzeptiert und im Datenobjekt gespeichert wird



Ändert man nun den Text im „EditText“, so wird der Text auch in der TextView geändert

Zusammenfassung Observable Data

- Man leitet die Entity- Klasse (Datenklasse) von BaseObservable ab
- In den Getter- und Setter-Methoden wird die Methode notifyPropertyChanged() aufgerufen
-

EventHandler mit Parameter

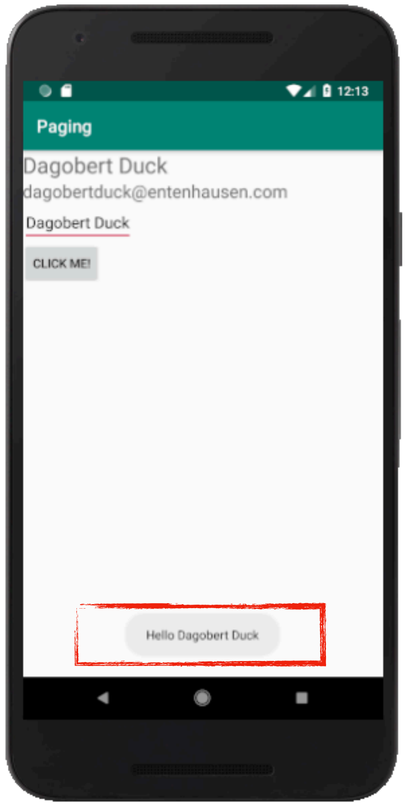
EventHandler mit Parameter

activity_main.xml

```
<Button
        android:id="@+id/btn_click"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CLICK ME!"
        android:onClick="@{() -> handler.onButtonClick(contact.name)}"/>
```

EventHandler.kt

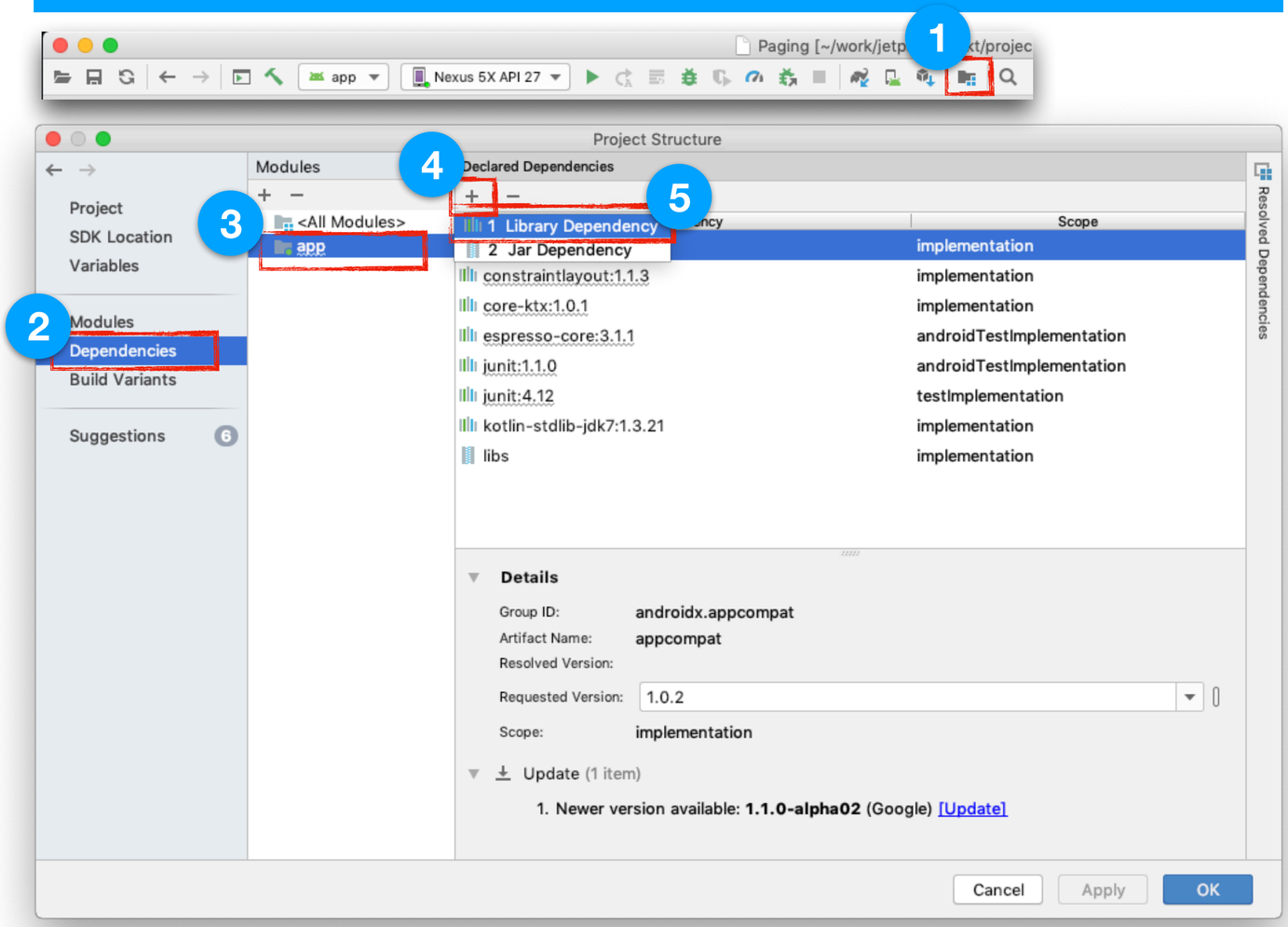
```
open class EventHandler(context: Context) {

    val myContext = context

    fun onButtonClick(name: String) {
        Toast.makeText(myContext, "Hello $name", Toast.LENGTH_SHORT).show()
    }
}
```



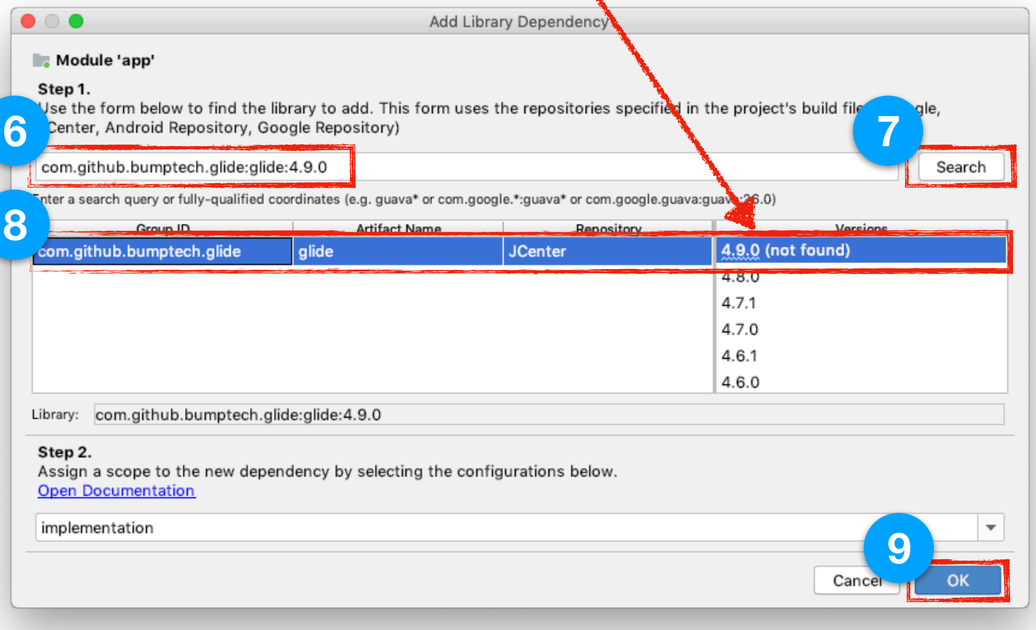
Loading images from an URL with Glide

Voraussetzungen

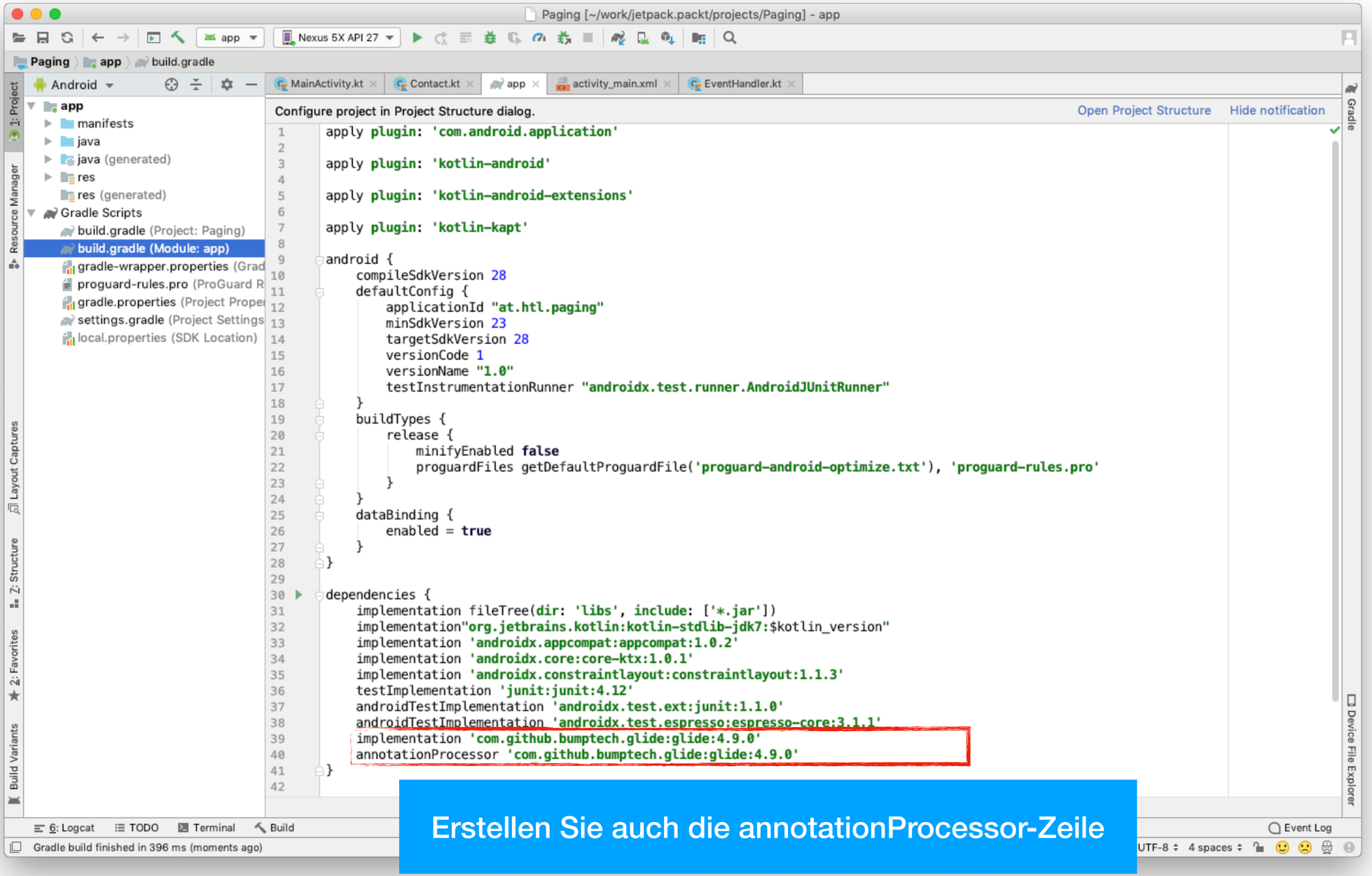
Man kann die Bibliotheken (Libraries) entweder in gradle.build eintragen, oder in der IDE



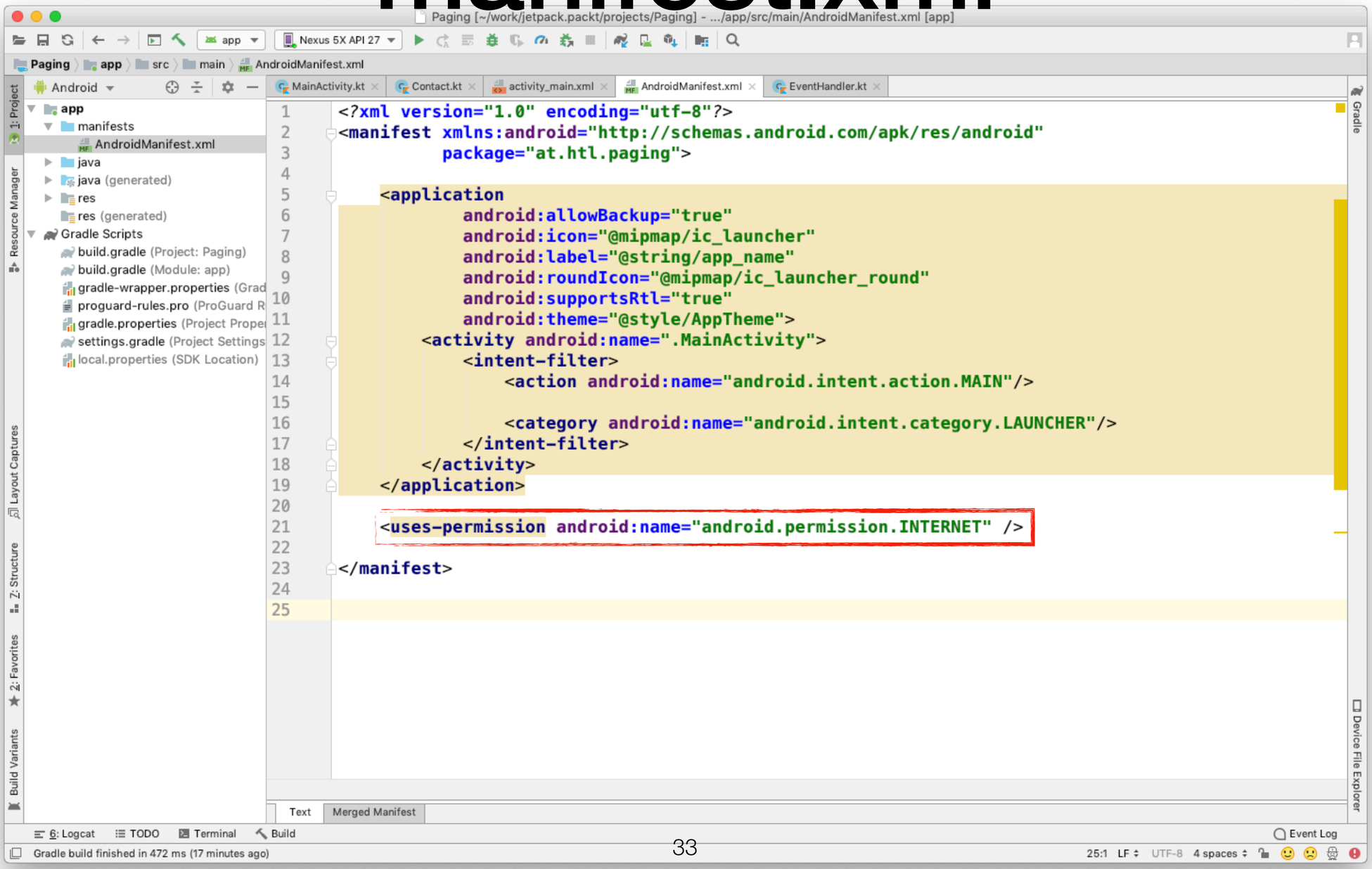
Die Library in der Version 4.9.0 wird nicht gefunden, da sie noch nicht im lokalen maven-repo enthalten ist



https://github.com/bumptech/glide


Paging [~/work/jetpack.packt/projects/Paging] - app
Paging app build.gradle
Android
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: Paging)
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro
gradle.properties
settings.gradle
local.properties
MainActivity.kt
Contact.kt
app
activity_main.xml
EventHandler.kt
Configure project in Project Structure dialog.
Open Project Structure
Hide notification
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'kotlin-kapt'
android {
compileSdkVersion 28
defaultConfig {
applicationId "at.htl.paging"
minSdkVersion 23
targetSdkVersion 28
versionCode 1
versionName "1.0"
testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
}
buildTypes {
release {
minifyEnabled false
proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
}
}
dataBinding {
enabled = true
}
}
dependencies {
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation"org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
implementation 'androidx.appcompat:appcompat:1.0.2'
implementation 'androidx.core:core-ktx:1.0.1'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'androidx.test.ext:junit:1.1.0'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
implementation 'com.github.bumptech.glide:glide:4.9.0'
annotationProcessor 'com.github.bumptech.glide:glide:4.9.0'
}
6: Logcat
TODO
Terminal
Build
Event Log
Gradle build finished in 396 ms (moments ago)
UTF-8
4 spaces
Erstellen Sie auch die annotationProcessor-Zeile

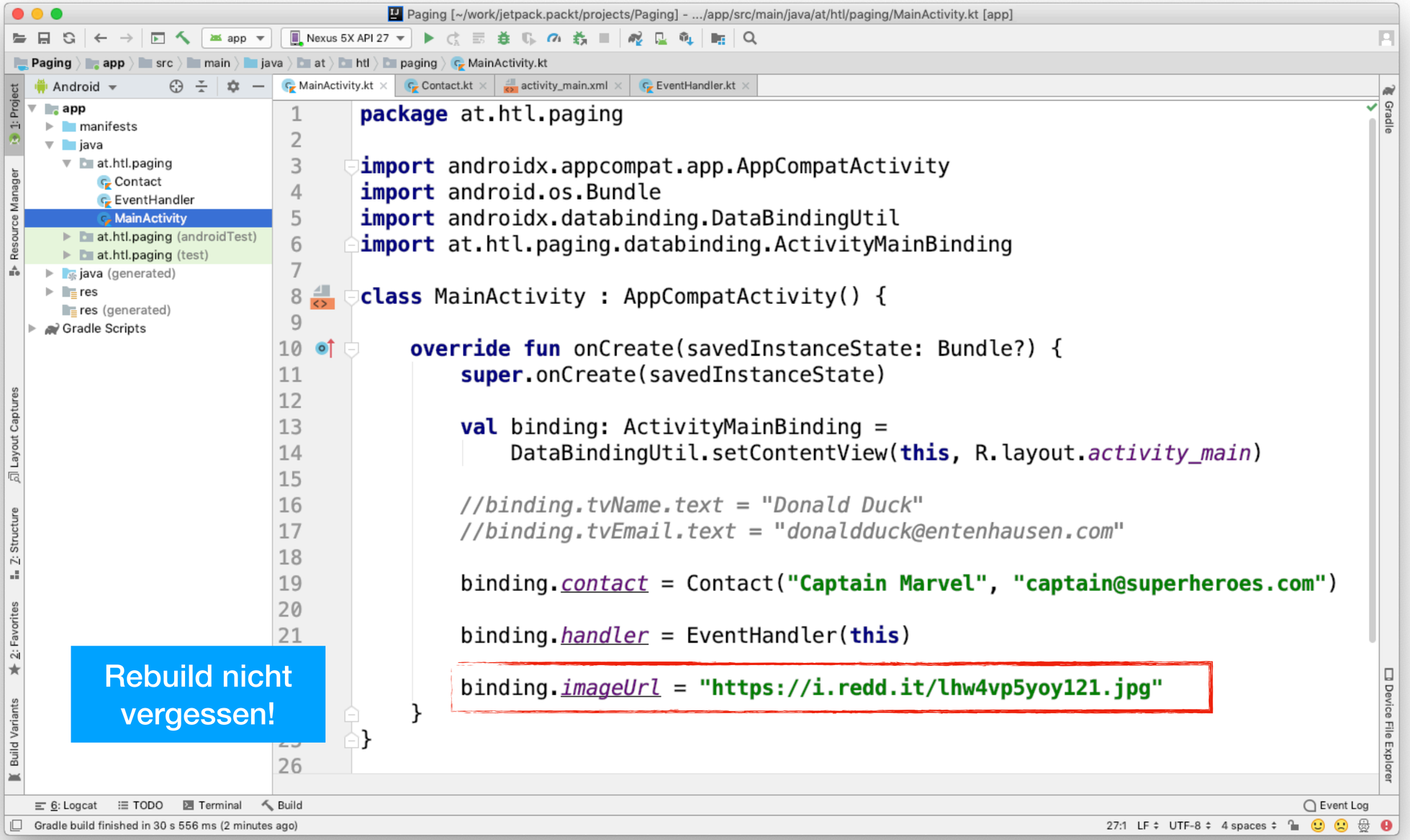
manifest.xml



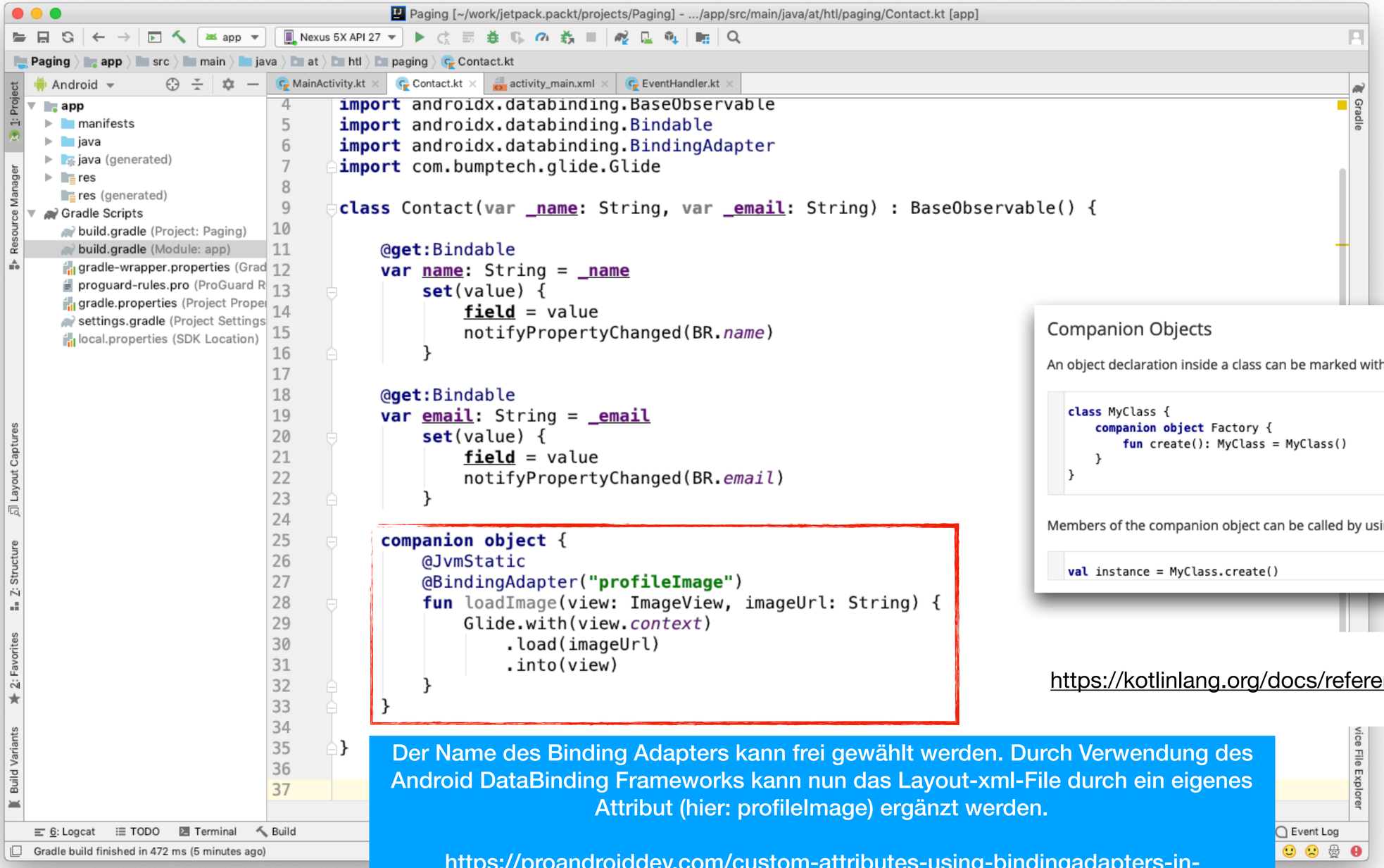
activity_main.xml

Die LinearLayouts werden verschachtelt. Dies ist keine optimale Lösung. Besser: ConstraintLayout

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

    <data>
        <variable
                name="contact"
                type="at.htl.paging.Contact"/>
        <variable
                name="handler"
                type="at.htl.paging.EventHandler"/>
        <variable
                name="imageUrl"
                type="String"/>
    </data>

    <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MainActivity">

        <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:padding="5dp">

            <ImageView
                    android:id="@+id/iv_profile_image"
                    android:layout_width="100dp"
                    android:layout_height="100dp"
                    android:padding="5dp"
                    android:src="@mipmap/ic_launcher"/>
        </LinearLayout>

        <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="5dp">
            <TextView
                    android:id="@+id/tv_name"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@{contact.name, default=Name}"
                    android:textSize="25sp"/>
            <TextView
                    android:id="@+id/tv_email"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text='@{contact.email, default="email@gmail.com"}'
                    android:textSize="20sp"/>
            <EditText
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:id="@+id/et_name"
                    android:inputType="text"
                    android:text="@={contact.name, default=Name}"/>
            <Button
                    android:id="@+id/btn_click"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="CLICK ME!"
                    android:onClick="@{() -> handler.onButtonClick(contact.name)}"/>
        </LinearLayout>
    </LinearLayout>
</layout>
```



https://i.redd.it/lhw4vp5yoy121.jpg



Companion Objects

An object declaration inside a class can be marked with the `companion` keyword:

```
class MyClass {
    companion object Factory {
        fun create(): MyClass = MyClass()
    }
}
```

Members of the companion object can be called by using simply the class name as the qualifier:

```
val instance = MyClass.create()
```

https://kotlinlang.org/docs/reference/object-declarations.html

Der Name des Binding Adapters kann frei gewählt werden. Durch Verwendung des Android DataBinding Frameworks kann nun das Layout-xml-File durch ein eigenes Attribut (hier: profileImage) ergänzt werden.

https://proandroiddev.com/custom-attributes-using-bindingadapters-in-kotlin-971ef8fcc259

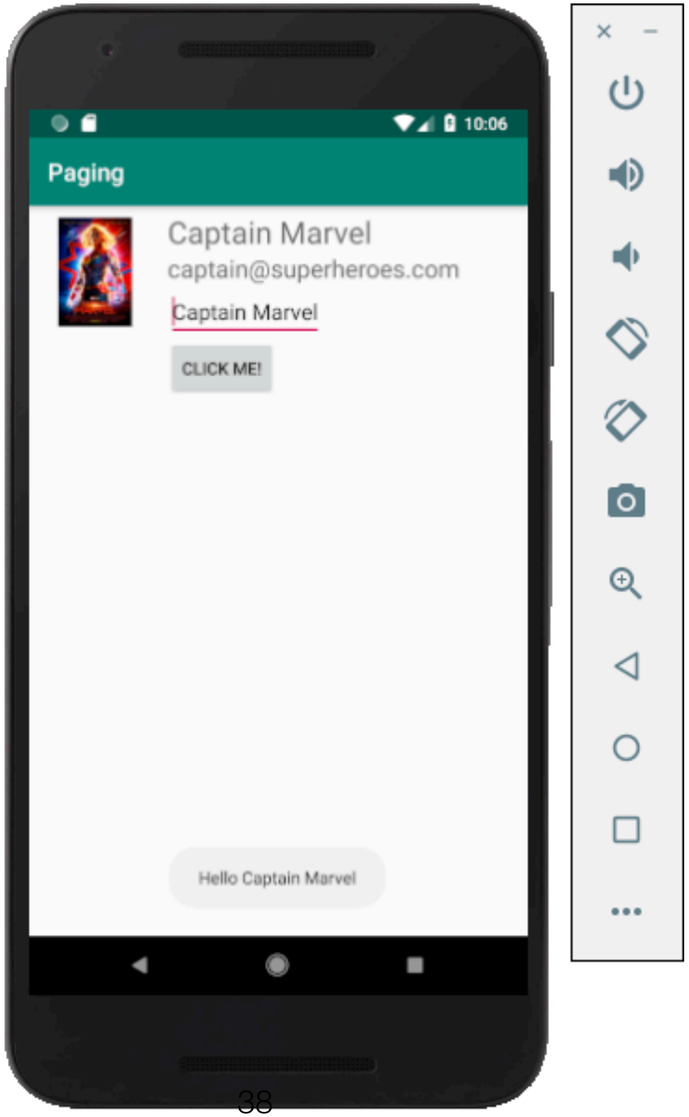
activity_main.xml

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

    <data>
        <variable
                name="contact"
                type="at.htl.paging.Contact"/>
        <variable
                name="handler"
                type="at.htl.paging.EventHandler"/>
        <variable
                name="imageUrl"
                type="String"/>
    </data>

    <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MainActivity">

        <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:padding="5dp">

            <ImageView
                    android:id="@+id/iv_profile_image"
                    android:layout_width="100dp"
                    android:layout_height="100dp"
                    android:padding="5dp"
                    app:profileImage="@{imageUrl}"
                    android:src="@mipmap/ic_launcher"/>
        </LinearLayout>

        <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="5dp">
            <TextView
                    android:id="@+id/tv_name"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@{contact.name, default=Name}"
                    android:textSize="25sp"/>
            <TextView
                    android:id="@+id/tv_email"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text='@{contact.email, default="email@gmail.com"}'
                    android:textSize="20sp"/>
            <EditText
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:id="@+id/et_name"
                    android:inputType="text"
                    android:text="@={contact.name, default=Name}"/>
            <Button
                    android:id="@+id/btn_click"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="CLICK ME!"
                    android:onClick="@{() -> handler.onButtonClick(contact.name)}"/>
        </LinearLayout>
    </LinearLayout>
</layout>
```

Den Namespace durch Alt-Enter halbautomatisch importieren

Testlauf

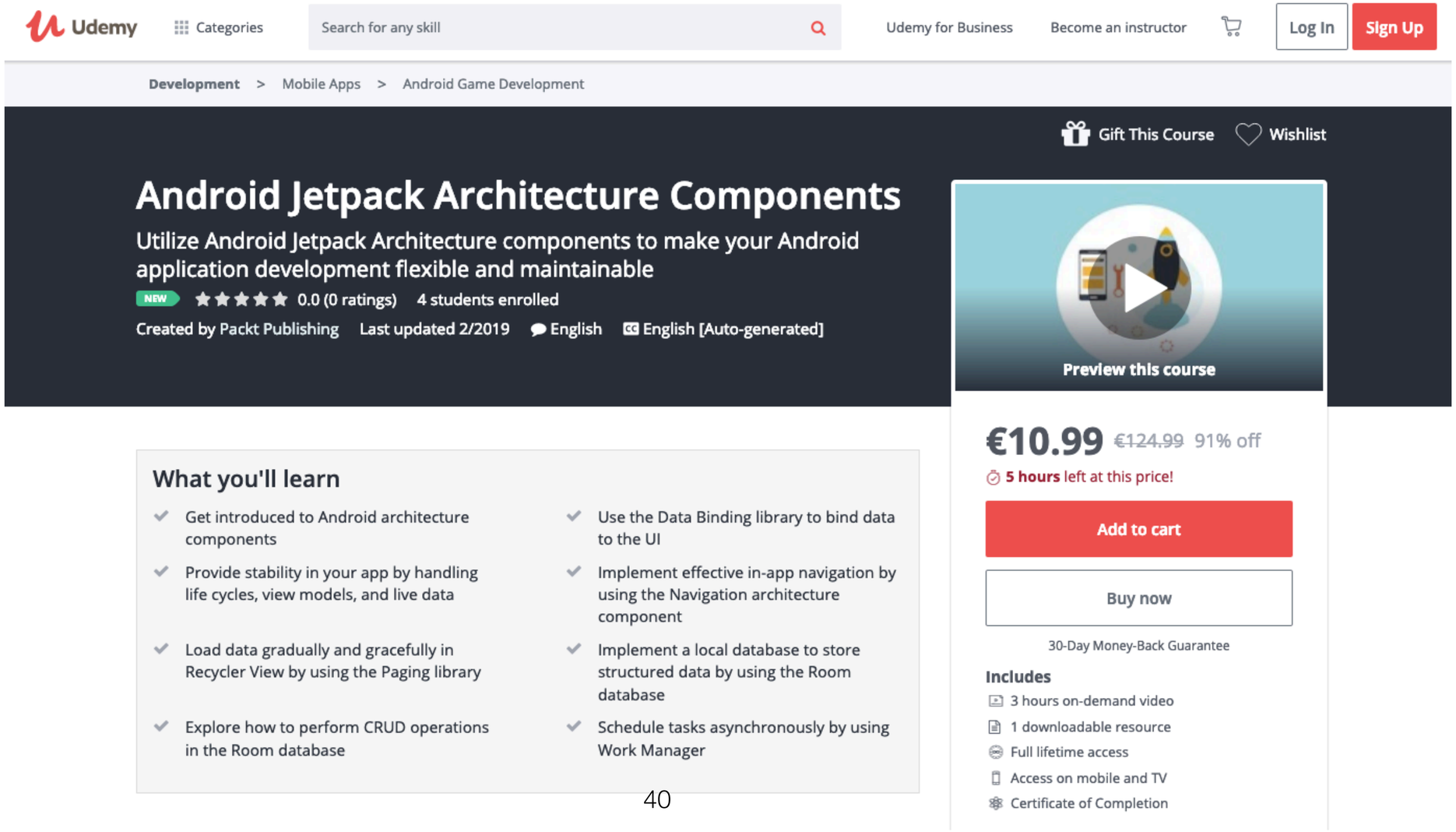

Very
Good


10:06
Paging
Captain Marvel
captain@superheroes.com
Captain Marvel
CLICK ME!
Hello Captain Marvel

Noch Fragen?

Quelle

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



Quellen

- https://codelabs.developers.google.com/codelabs/android-databinding/index.html#0
- https://codelabs.developers.google.com/?cat=Android
- https://www.vogella.com/tutorials/AndroidDatabinding/article.html
- https://proandroiddev.com/exploring-android-data-binding-library-ff467171c756
-