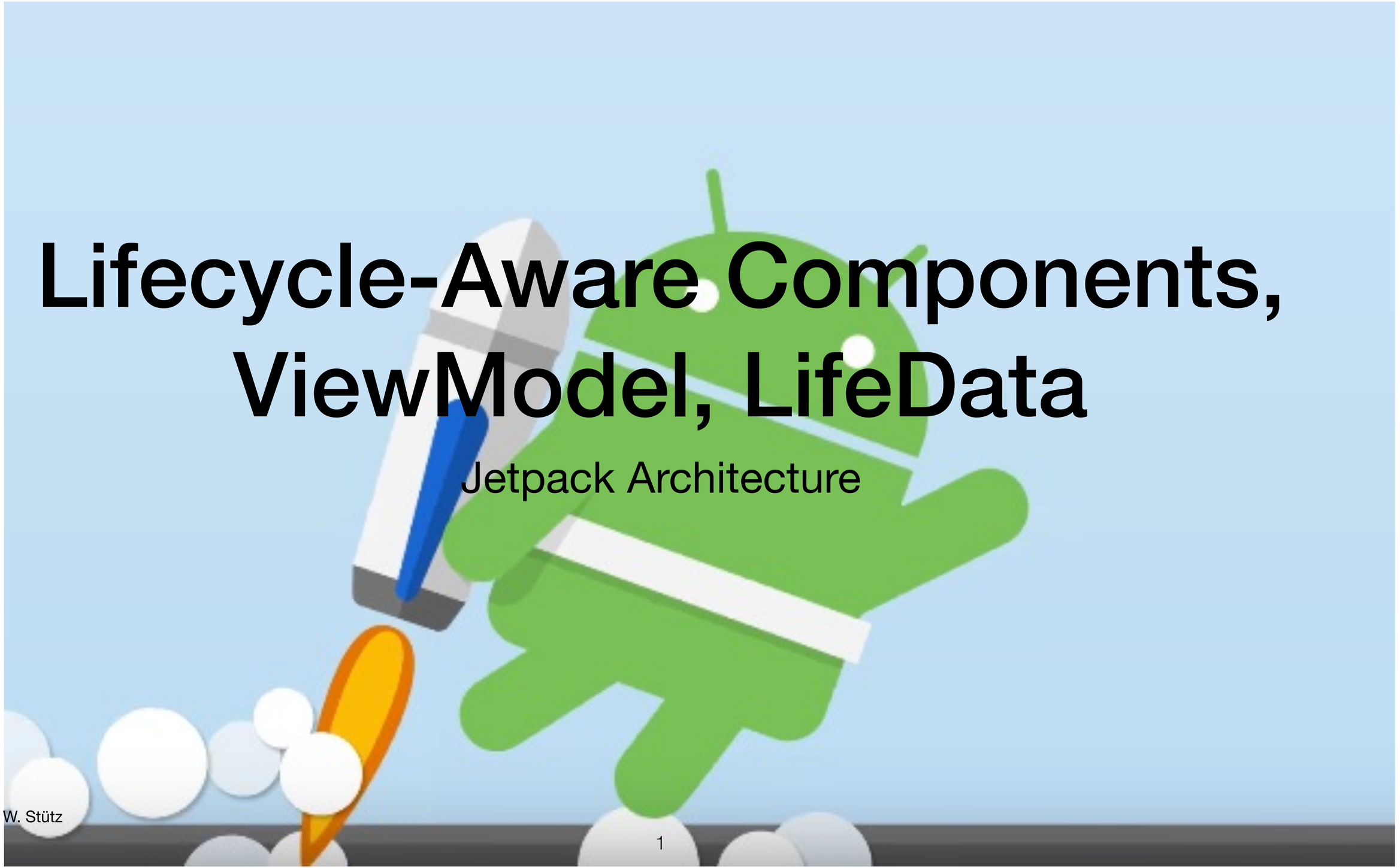
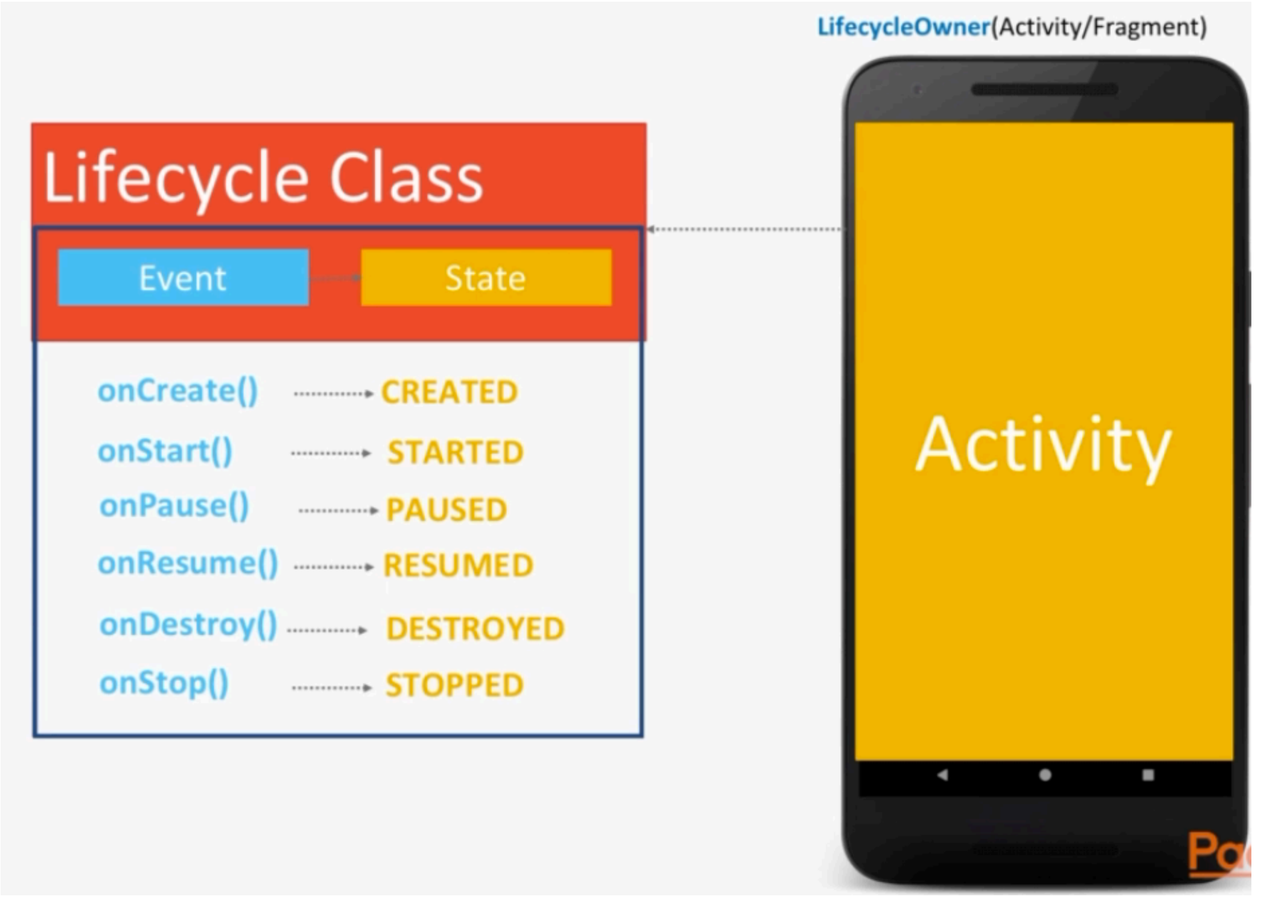
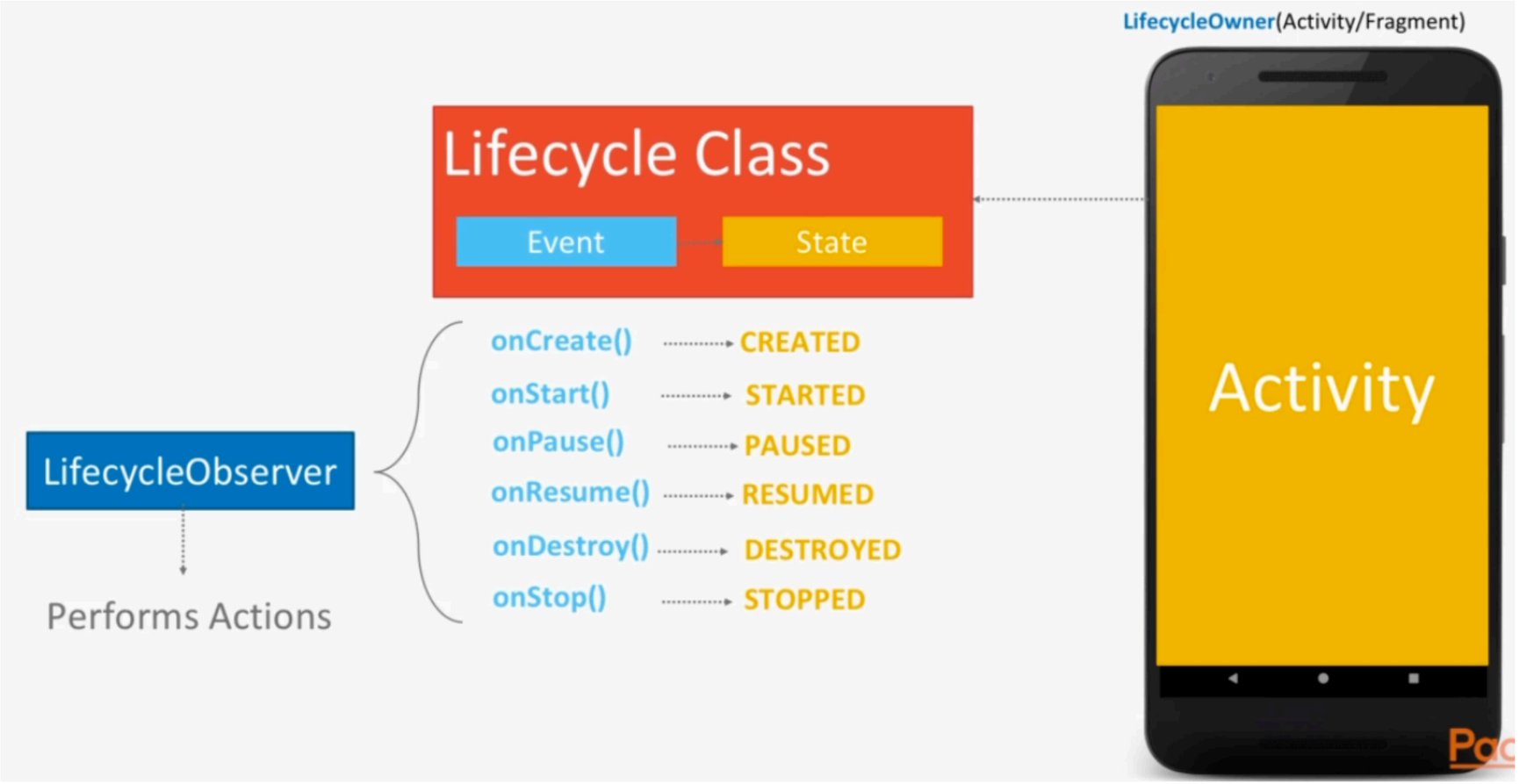
Lifecycle-Aware Components, ViewModel, LifeData

Jetpack Architecture



Thomas W. Stütz


LifecycleOwner(Activity/Fragment)
Lifecycle Class
Event
State
onCreate()
CREATED
onStart()
STARTED
onPause()
PAUSED
onResume()
RESUMED
onDestroy()
DESTROYED
onStop()
STOPPED
Activity


LifecycleOwner(Activity/Fragment)
Lifecycle Class
Event
State
LifecycleObserver
Performs Actions
onCreate()
CREATED
onStart()
STARTED
onPause()
PAUSED
onResume()
RESUMED
onDestroy()
DESTROYED
onStop()
STOPPED
Activity

Handling Lifecycles with Lifecycle-Aware Components

Part of Android Jetpack.

Lifecycle-aware components perform actions in response to a change in the lifecycle status of another component, such as activities and fragments. These components help you produce better-organized, and often lighter-weight code, that is easier to maintain.

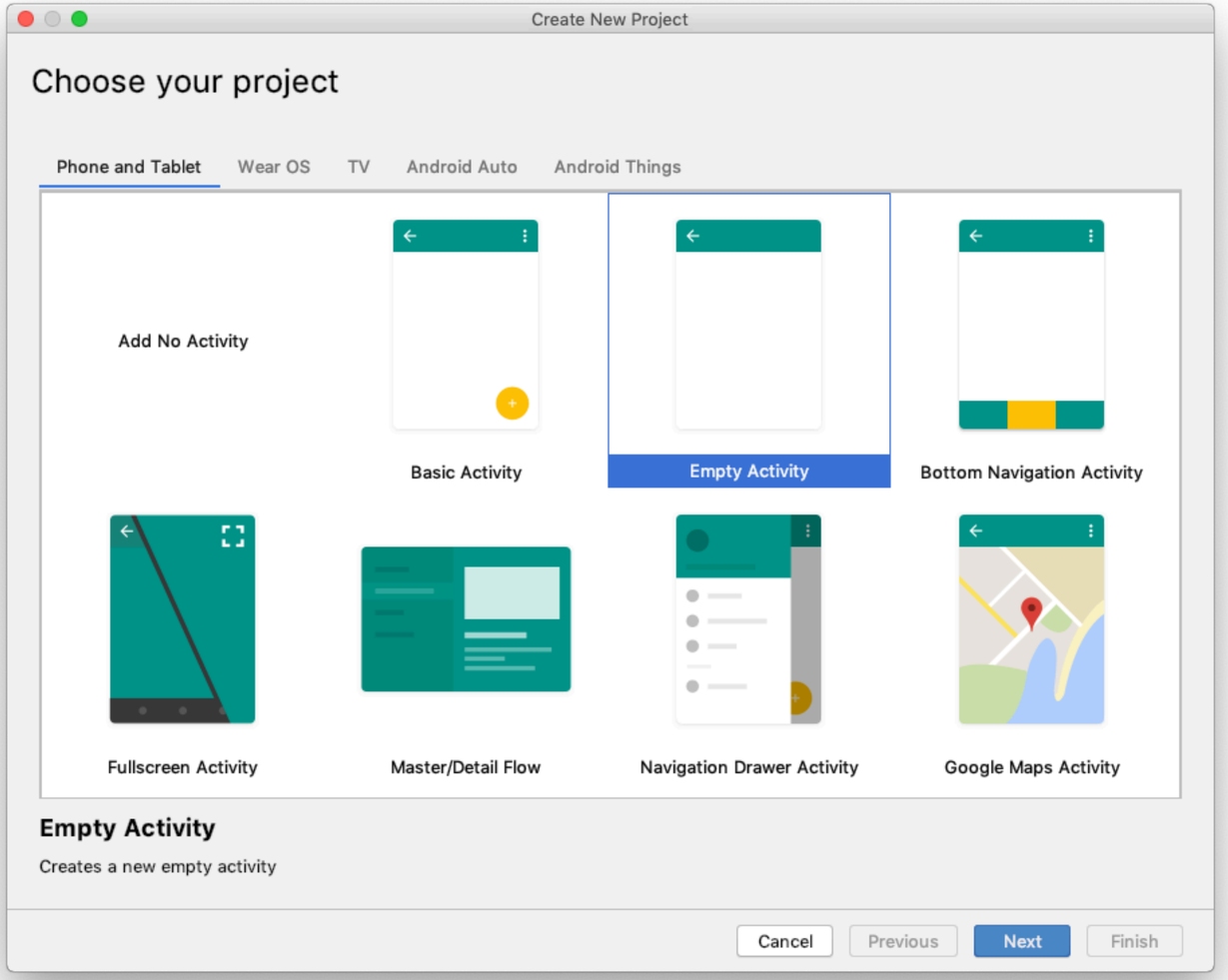
A common pattern is to implement the actions of the dependent components in the lifecycle methods of activities and fragments. However, this pattern leads to a poor organization of the code and to the proliferation of errors. By using lifecycle-aware components, you can move the code of dependent components out of the lifecycle methods and into the components themselves.

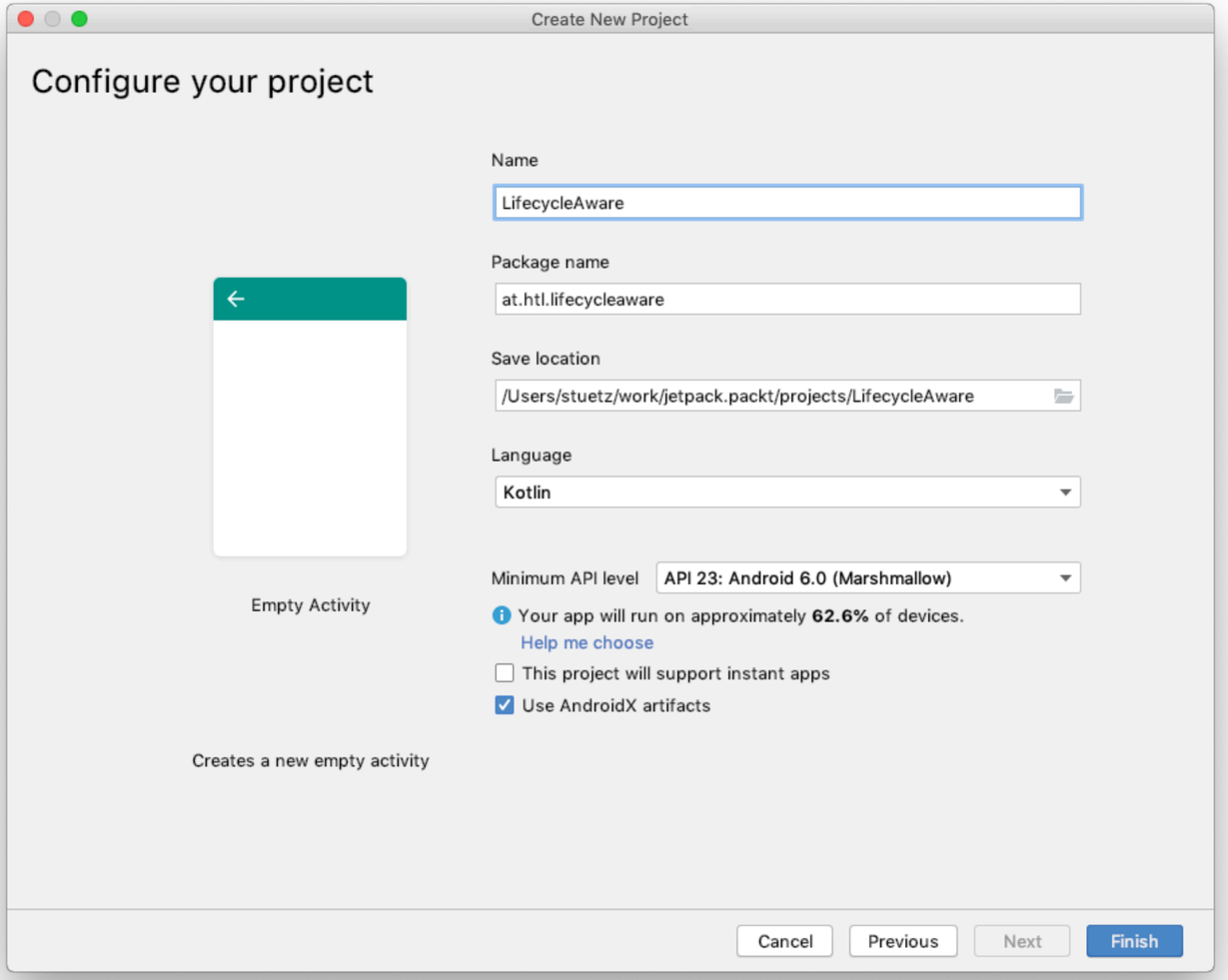
The `android.arch.lifecycle` package provides classes and interfaces that let you build *lifecycle-aware* components—which are components that can automatically adjust their behavior based on the current lifecycle state of an activity or fragment.

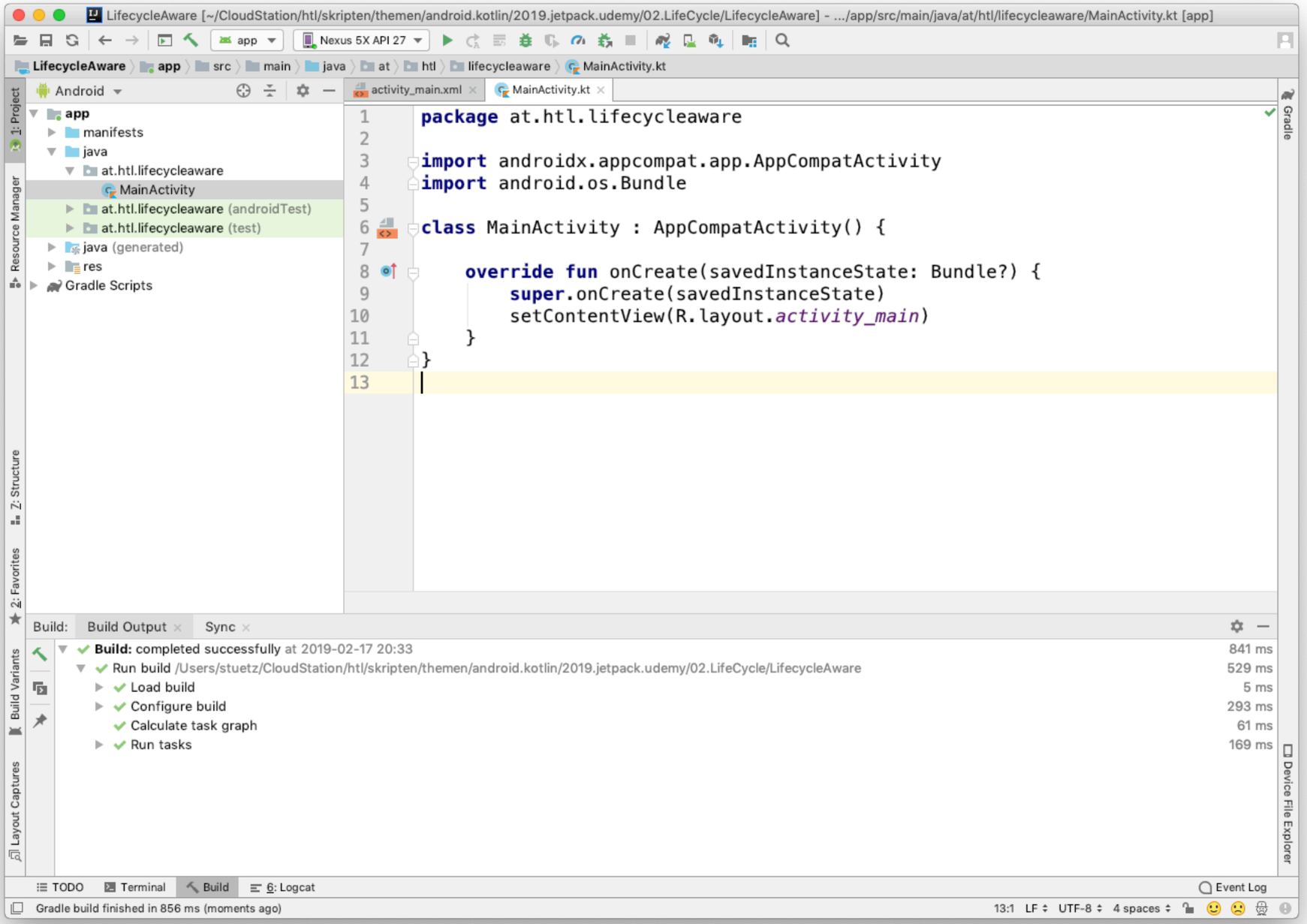
Note: To import `android.arch.lifecycle` into your Android project, see the instructions for declaring dependencies in the Lifecycle release notes.

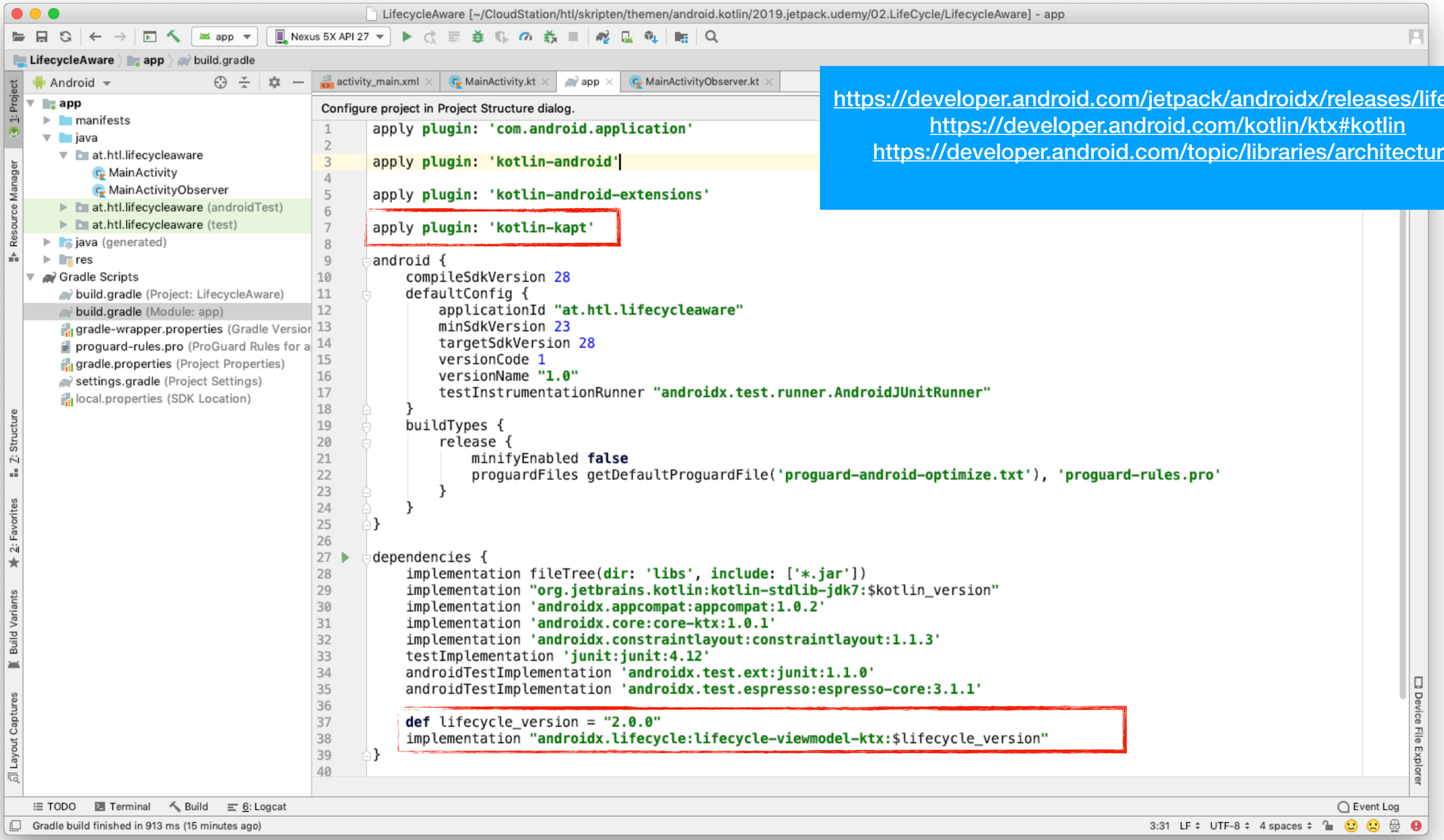
https://developer.android.com/topic/libraries/architecture/lifecycle

Lifecycle-Aware Components

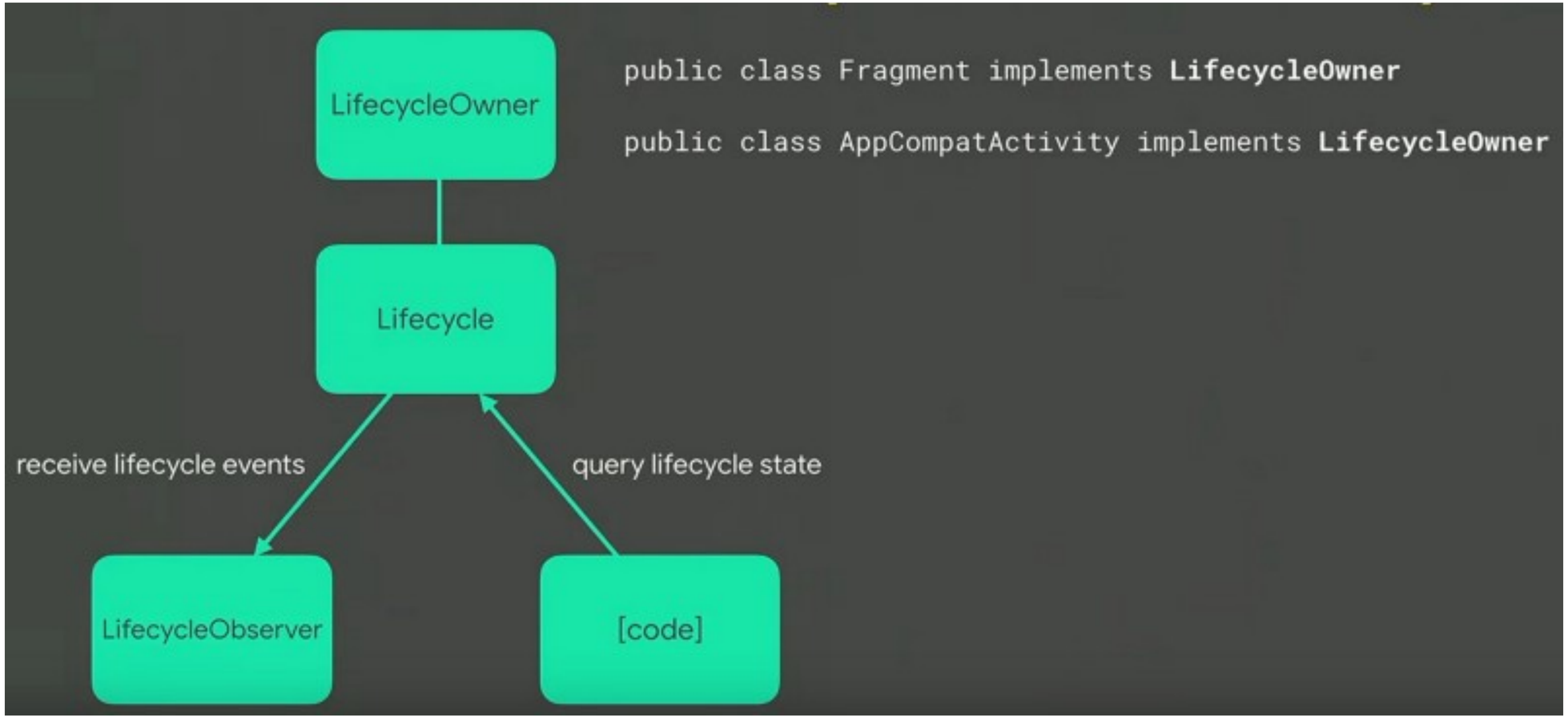

Create New Project
Choose your project
Phone and Tablet
Wear OS
TV
Android Auto
Android Things
Add No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Fullscreen Activity
Master/Detail Flow
Navigation Drawer Activity
Google Maps Activity
Empty Activity
Creates a new empty activity
Cancel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


Create New Project
Configure your project
Name
LifecycleAware
Package name
at.htl.lifecycleaware
Save location
/Users/stuetz/work/jetpack.packt/projects/LifecycleAware
Language
Kotlin
Minimum API level
API 23: Android 6.0 (Marshmallow)
Your app will run on approximately 62.6% of devices.
Help me choose
This project will support instant apps
Use AndroidX artifacts
Empty Activity
Creates a new empty activity
Cancel
Previous
Next
Finish


LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivity.kt [app]
app
Nexus 5X API 27
LifecycleAware
app
src
main
java
at
htl
lifecycleaware
MainActivity.kt
Android
app
manifests
java
at.htl.lifecycleaware
MainActivity
at.htl.lifecycleaware (androidTest)
at.htl.lifecycleaware (test)
java (generated)
res
Gradle Scripts
activity_main.xml
MainActivity.kt
1 package at.htl.lifecycleaware
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7
8 override fun onCreate(savedInstanceState: Bundle?) {
9 super.onCreate(savedInstanceState)
10 setContentView(R.layout.activity_main)
11 }
12 }
13
Build:
Build Output
Sync
Build: completed successfully at 2019-02-17 20:33 841 ms
Run build /Users/stuetz/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware 529 ms
Load build 5 ms
Configure build 293 ms
Calculate task graph 61 ms
Run tasks 169 ms
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants
Layout Captures
Gradle
Device File Explorer
TODO
Terminal
Build
6: Logcat
Event Log
Gradle build finished in 856 ms (moments ago)
13:1 LF UTF-8 4 spaces

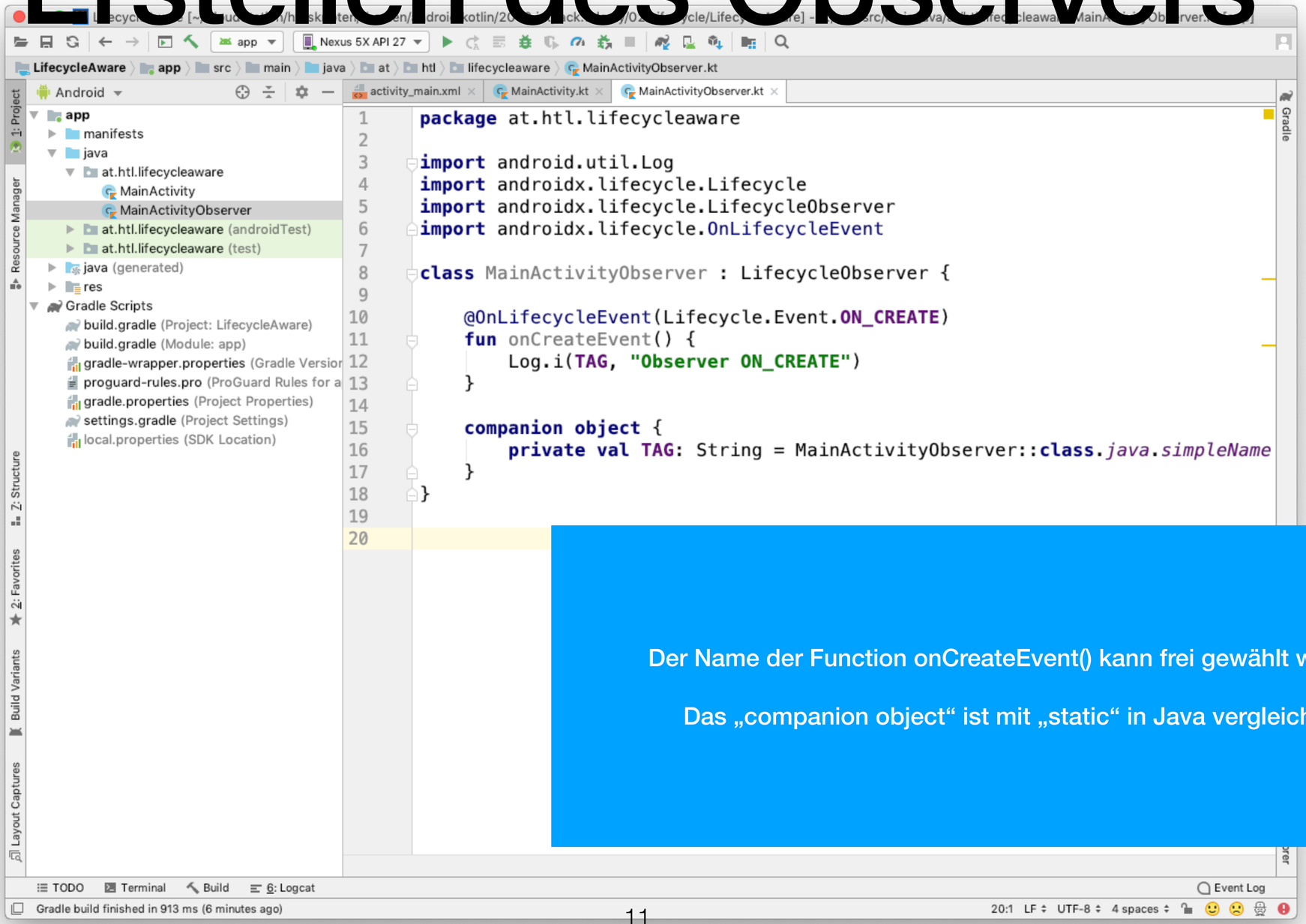


https://developer.android.com/jetpack/androidx/releases/lifecycle



https://android.jlelse.eu/develop-your-android-app-in-a-modern-way-google-i-o-18-tips-8e064bbf0dd9

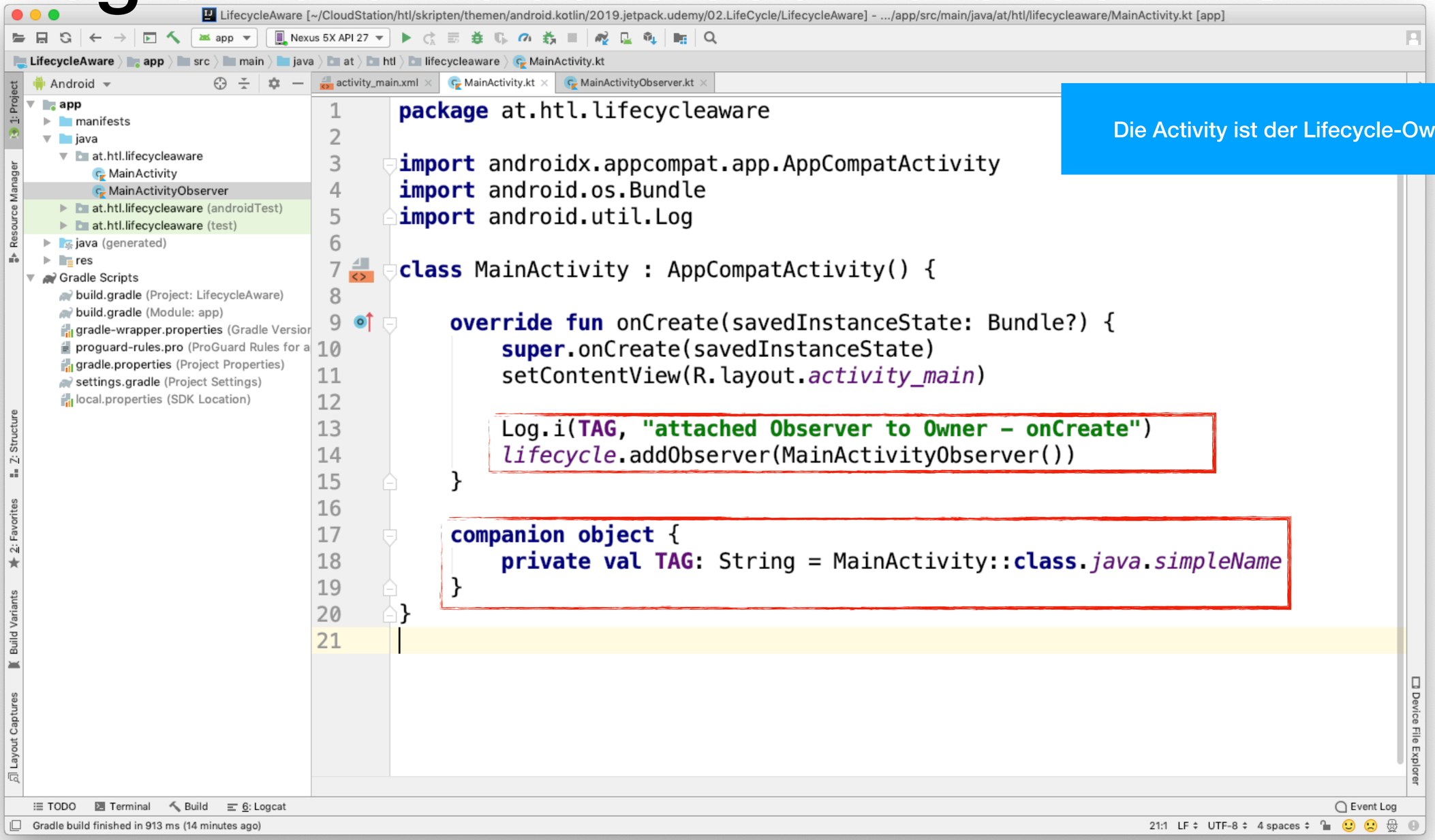
Erstellen des Observers

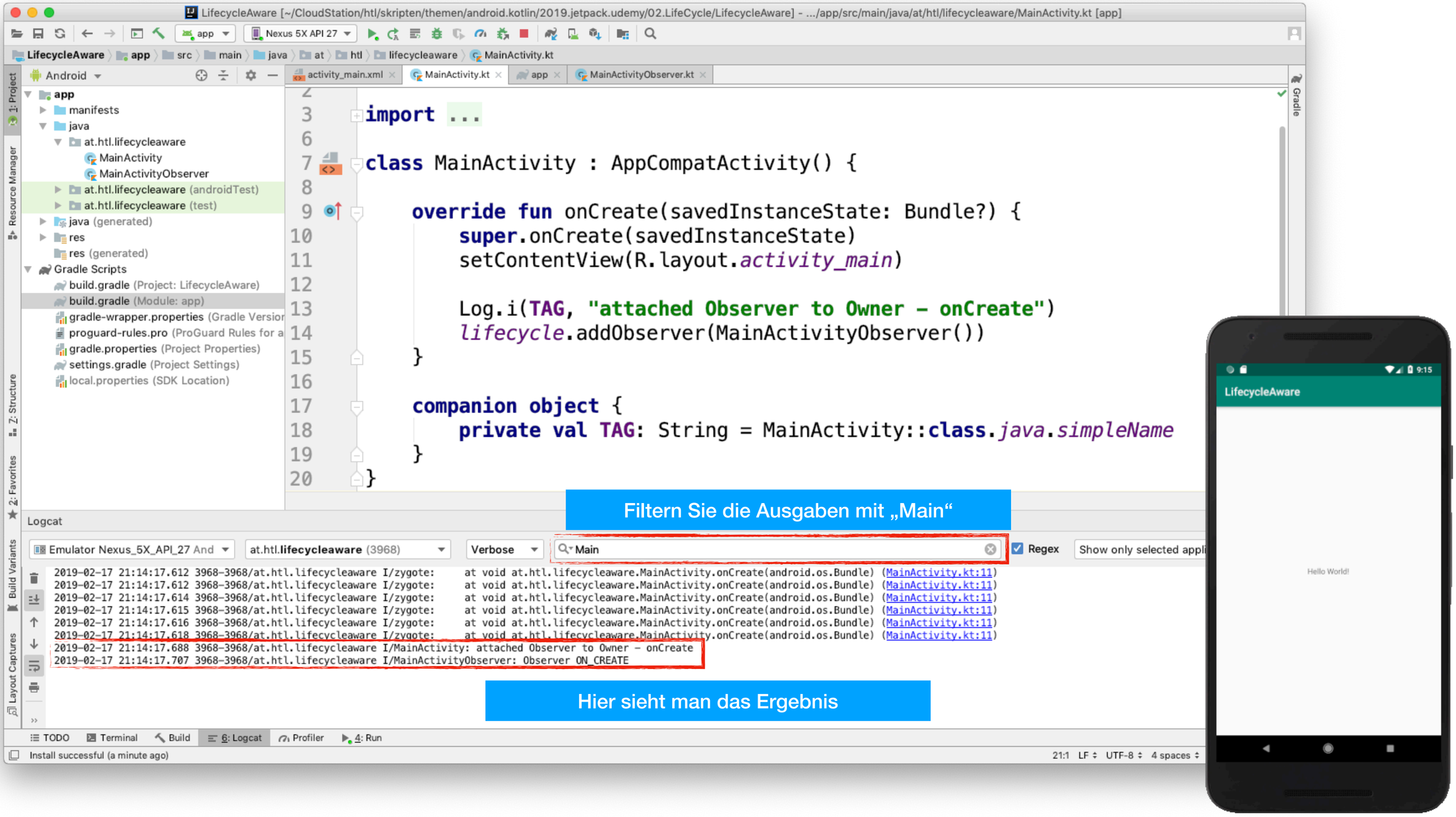


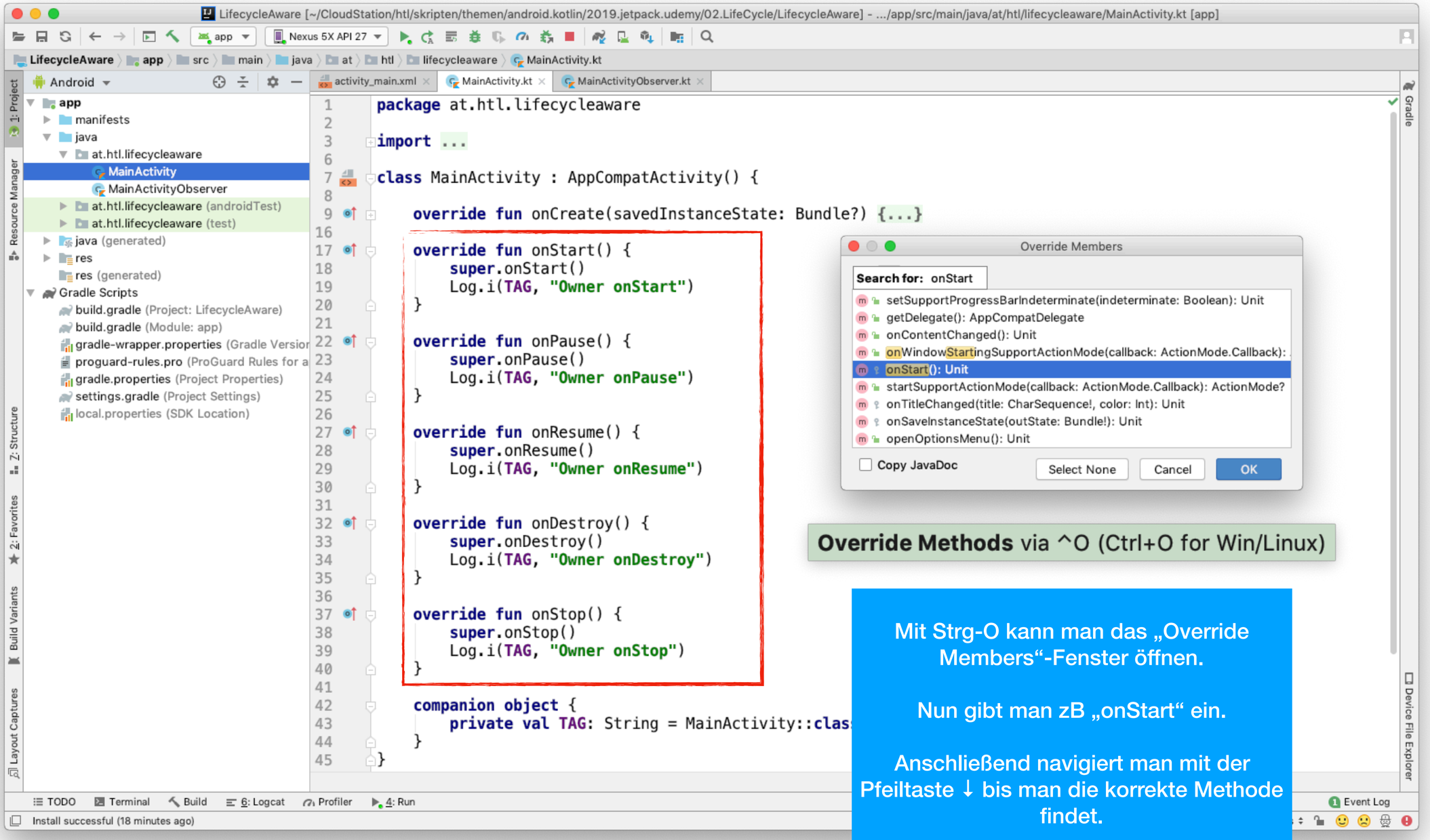
Der Name der Function onCreateEvent() kann frei gewählt werden.

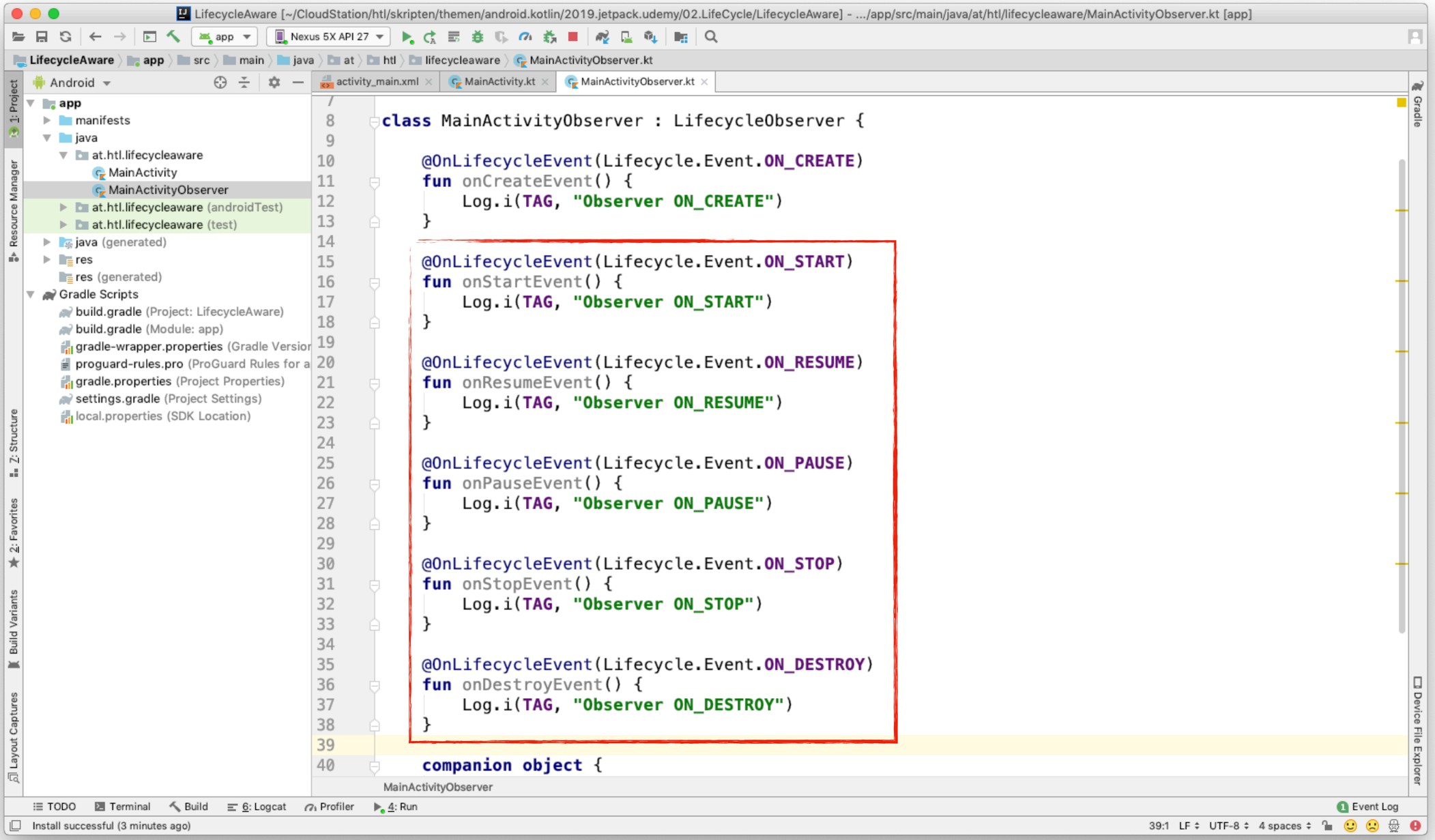
Das „companion object“ ist mit „static“ in Java vergleichbar

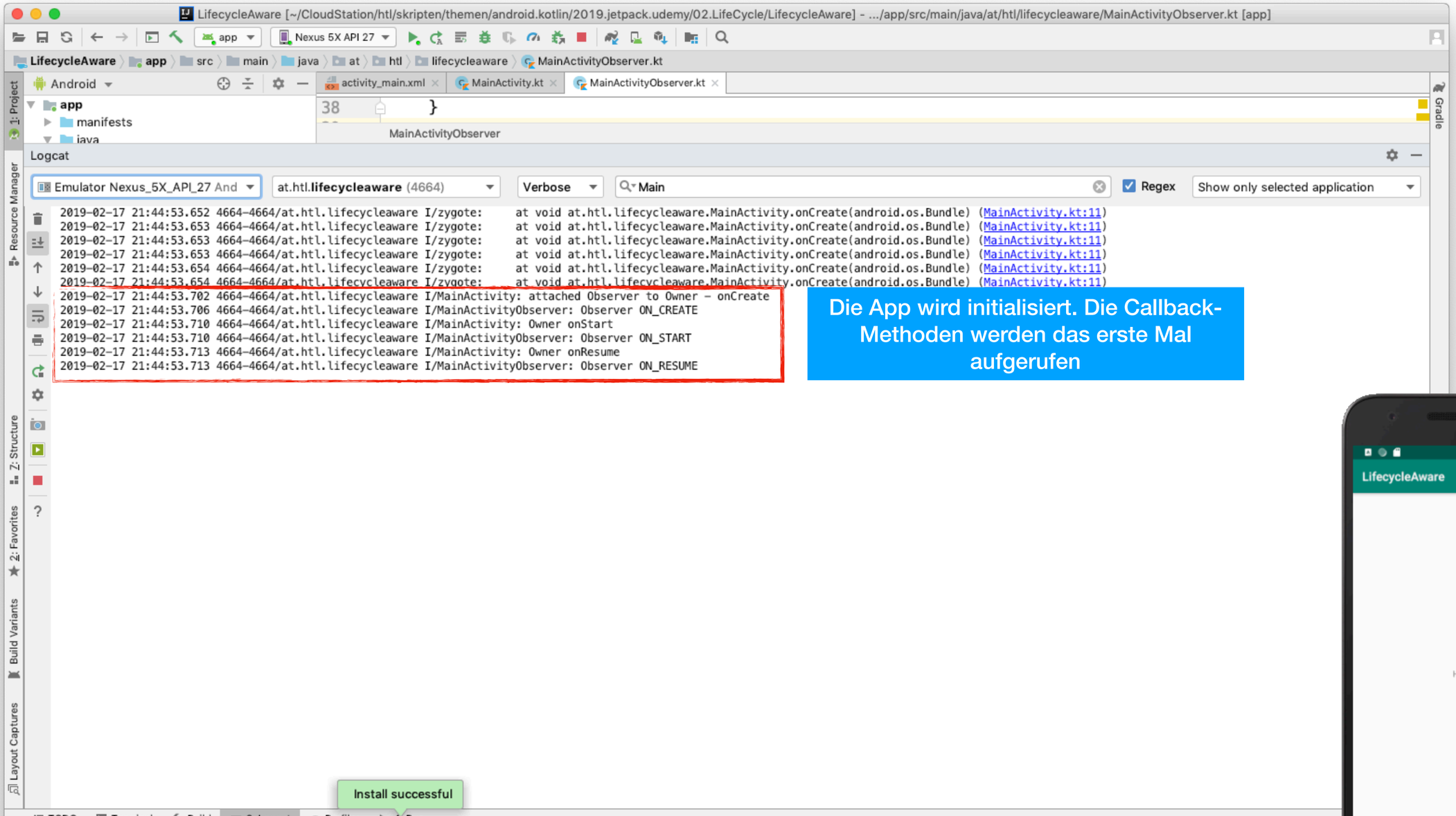
Registrieren des Observers beim Owner

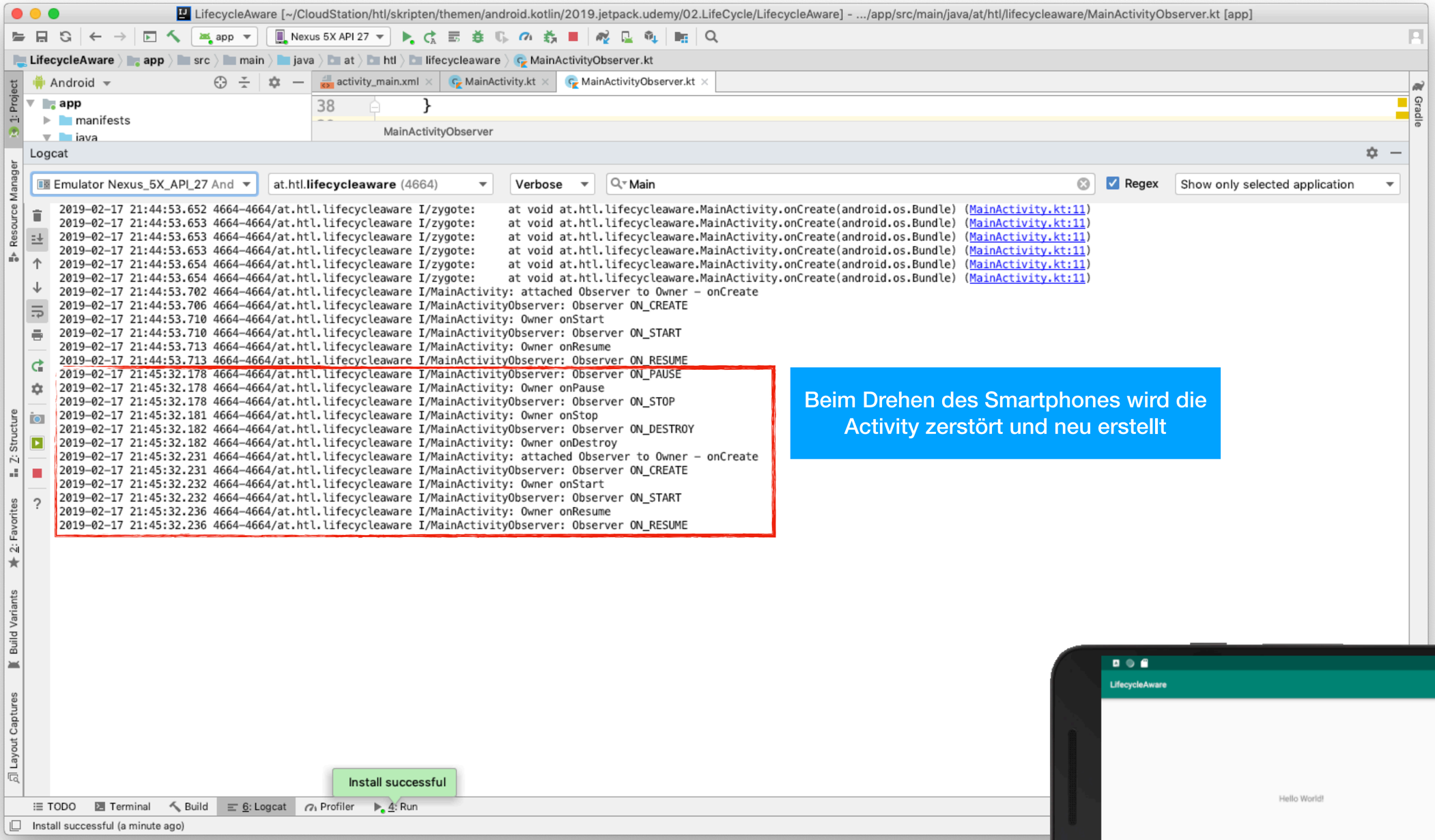


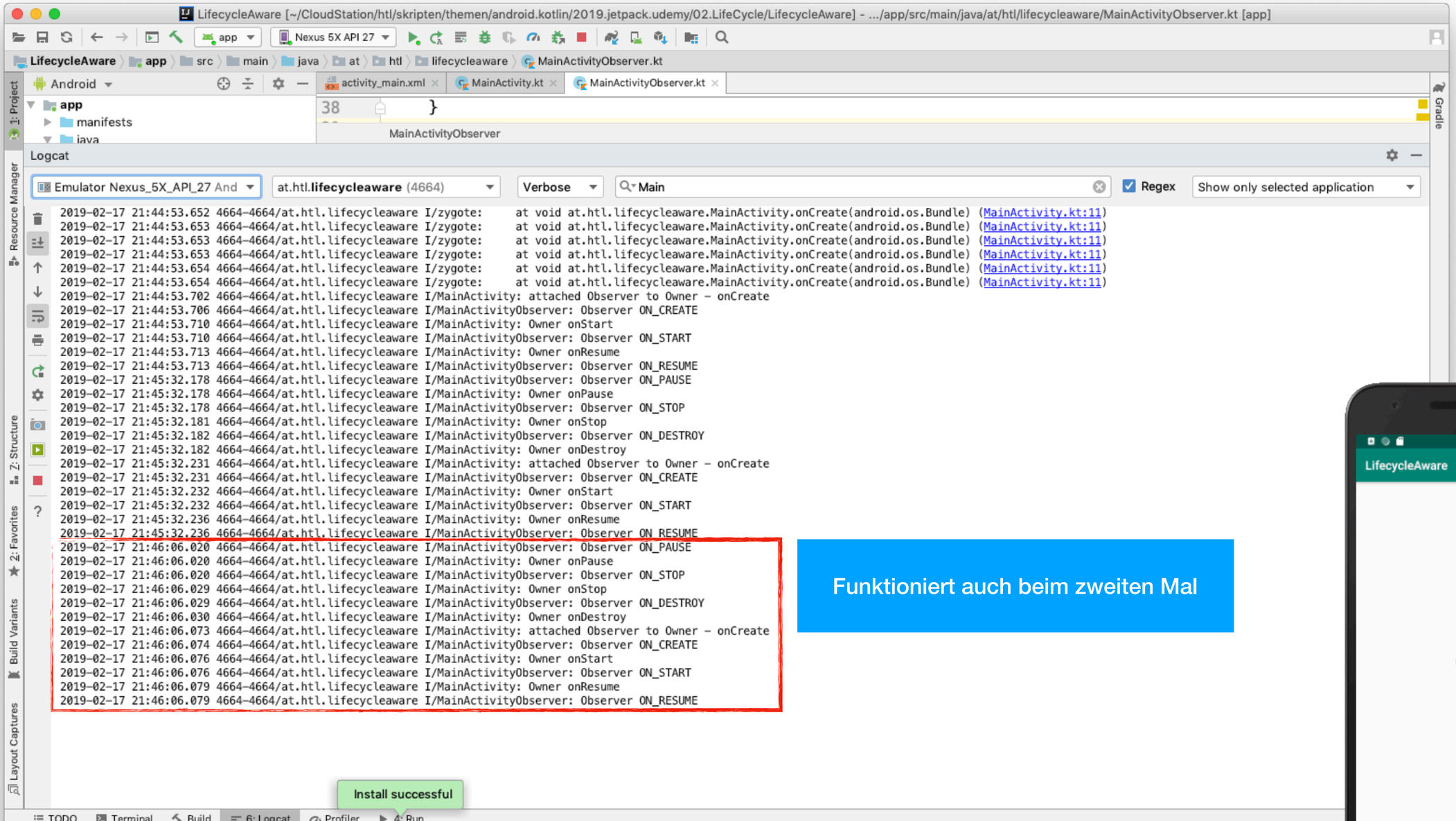

LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivity.kt [app]
app
Nexus 5X API 27
LifecycleAware app src main java at htl lifecycleaware MainActivity.kt
Android
app
manifests
java
at.htl.lifecycleaware
MainActivity
MainActivityObserver
at.htl.lifecycleaware (androidTest)
at.htl.lifecycleaware (test)
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: LifecycleAware)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version
proguard-rules.pro (ProGuard Rules for a
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.kt
app
MainActivityObserver.kt
import ...
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)
Log.i(TAG, "attached Observer to Owner - onCreate")
lifecycle.addObserver(MainActivityObserver())
}
companion object {
private val TAG: String = MainActivity::class.java.simpleName
}
}
Logcat
Filtern Sie die Ausgaben mit „Main“
Emulator Nexus_5X_API_27 And
at.htl.lifecycleaware (3968)
Verbose
Main
Regex
Show only selected appli
2019-02-17 21:14:17.612 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.612 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.614 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.615 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.616 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.618 3968-3968/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:14:17.688 3968-3968/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:14:17.707 3968-3968/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
Hier sieht man das Ergebnis
TODO Terminal Build 6: Logcat Profiler 4: Run
Install successful (a minute ago)
21:1 LF UTF-8 4 spaces
LifecycleAware
Hello World!


LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivity.kt [app]
app
Nexus 5X API 27
LifecycleAware › app › src › main › java › at › htl › lifecycleaware › MainActivity.kt
Android
app
manifests
java
at.htl.lifecycleaware
MainActivity
MainActivityObserver
at.htl.lifecycleaware (androidTest)
at.htl.lifecycleaware (test)
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: LifecycleAware)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version
proguard-rules.pro (ProGuard Rules for a
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.kt
MainActivityObserver.kt
1 package at.htl.lifecycleaware
3 import ...
7 class MainActivity : AppCompatActivity() {
9 override fun onCreate(savedInstanceState: Bundle?) {...}
17 override fun onStart() {
18 super.onStart()
19 Log.i(TAG, "Owner onStart")
20 }
22 override fun onPause() {
23 super.onPause()
24 Log.i(TAG, "Owner onPause")
25 }
27 override fun onResume() {
28 super.onResume()
29 Log.i(TAG, "Owner onResume")
30 }
32 override fun onDestroy() {
33 super.onDestroy()
34 Log.i(TAG, "Owner onDestroy")
35 }
37 override fun onStop() {
38 super.onStop()
39 Log.i(TAG, "Owner onStop")
40 }
42 companion object {
43 private val TAG: String = MainActivity::clas
44 }
45 }
Override Members
Search for: onStart
setSupportProgressBarIndeterminate(indeterminate: Boolean): Unit
getDelegate(): AppCompatDelegate
onContentChanged(): Unit
onWindowStartingSupportActionMode(callback: ActionMode.Callback):
onStart(): Unit
startSupportActionMode(callback: ActionMode.Callback): ActionMode?
onTitleChanged(title: CharSequence!, color: Int): Unit
onSaveInstanceState(outState: Bundle!): Unit
openOptionsMenu(): Unit
Copy JavaDoc
Select None
Cancel
OK
Override Methods via ^O (Ctrl+O for Win/Linux)
Mit Strg-O kann man das „Override Members“-Fenster öffnen.
Nun gibt man zB „onStart“ ein.
Anschließend navigiert man mit der Pfeiltaste ↓ bis man die korrekte Methode findet.
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Event Log
Install successful (18 minutes ago)


LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivityObserver.kt [app]
app
Nexus 5X API 27
LifecycleAware
app
src
main
java
at
htl
lifecycleaware
MainActivityObserver.kt
Android
activity_main.xml
MainActivity.kt
MainActivityObserver.kt
app
manifests
java
at.htl.lifecycleaware
MainActivity
MainActivityObserver
at.htl.lifecycleaware (androidTest)
at.htl.lifecycleaware (test)
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: LifecycleAware)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version
proguard-rules.pro (ProGuard Rules for a
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
class MainActivityObserver : LifecycleObserver {
@OnLifecycleEvent(Lifecycle.Event.ON_CREATE)
fun onCreateEvent() {
Log.i(TAG, "Observer ON_CREATE")
}
@OnLifecycleEvent(Lifecycle.Event.ON_START)
fun onStartEvent() {
Log.i(TAG, "Observer ON_START")
}
@OnLifecycleEvent(Lifecycle.Event.ON_RESUME)
fun onResumeEvent() {
Log.i(TAG, "Observer ON_RESUME")
}
@OnLifecycleEvent(Lifecycle.Event.ON_PAUSE)
fun onPauseEvent() {
Log.i(TAG, "Observer ON_PAUSE")
}
@OnLifecycleEvent(Lifecycle.Event.ON_STOP)
fun onStopEvent() {
Log.i(TAG, "Observer ON_STOP")
}
@OnLifecycleEvent(Lifecycle.Event.ON_DESTROY)
fun onDestroyEvent() {
Log.i(TAG, "Observer ON_DESTROY")
}
companion object {
MainActivityObserver
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Event Log
Install successful (3 minutes ago)
39:1
LF
UTF-8
4 spaces


LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivityObserver.kt [app]
app
Nexus 5X API 27
LifecycleAware app src main java at htl lifecycleaware MainActivityObserver.kt
Android
activity_main.xml
MainActivity.kt
MainActivityObserver.kt
app
manifests
java
38 }
MainActivityObserver
Logcat
Emulator Nexus_5X_API_27 And
at.htl.lifecycleaware (4664)
Verbose
Main
Regex
Show only selected application
2019-02-17 21:44:53.652 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.702 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:44:53.706 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
Die App wird initialisiert. Die Callback-Methoden werden das erste Mal aufgerufen
Install successful
TODO Terminal Build 6: Logcat Profiler 4: Run
Install successful (moments ago)
39:1 LF UTF-8 4 spaces
LifecycleAware
Hello World!

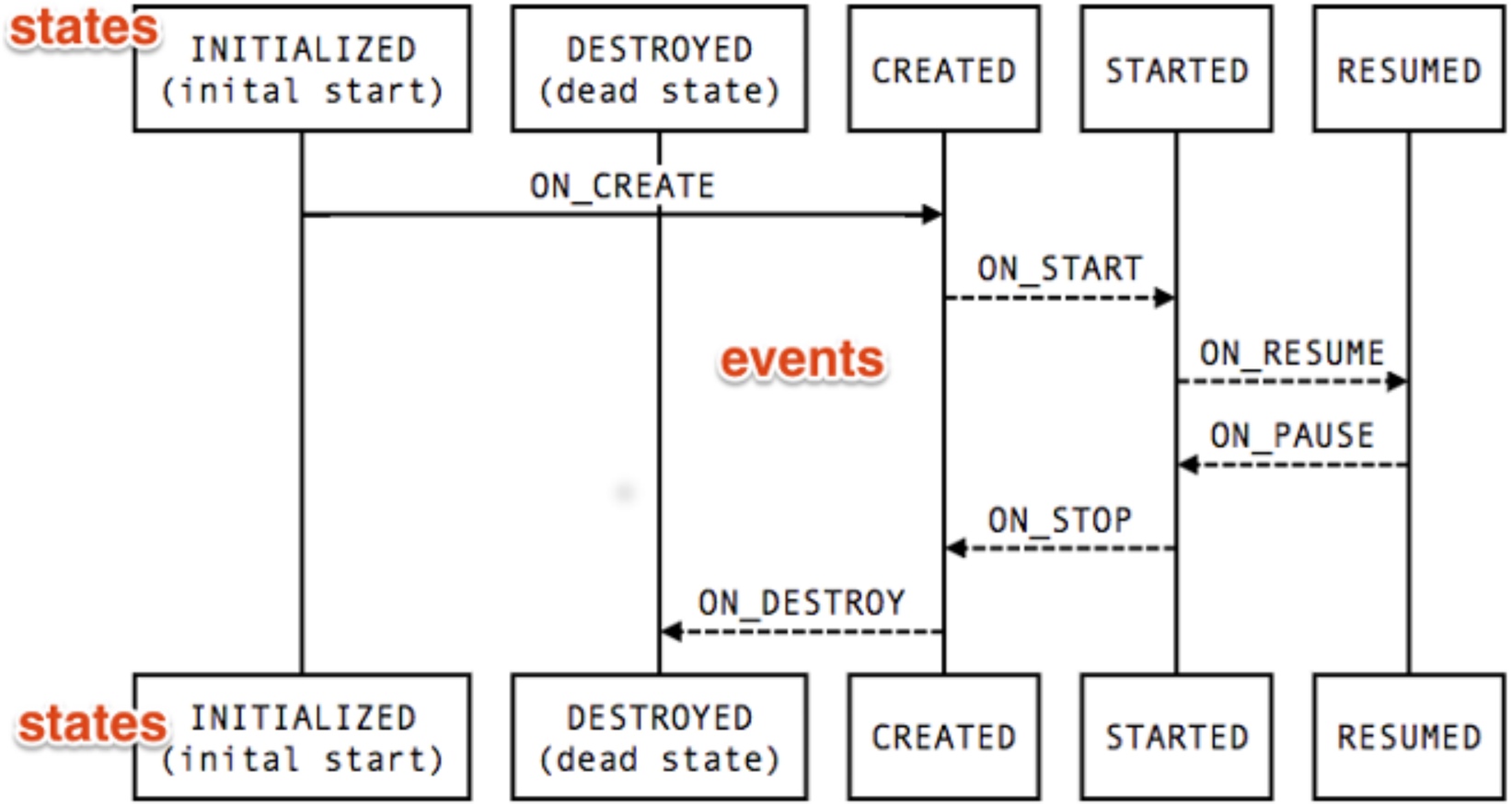

LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivityObserver.kt [app]
app
Nexus 5X API 27
LifecycleAware
app
src
main
java
at
htl
lifecycleaware
MainActivityObserver.kt
Android
activity_main.xml
MainActivity.kt
MainActivityObserver.kt
app
manifests
java
38
}
MainActivityObserver
Logcat
Emulator Nexus_5X_API_27 And
at.htl.lifecycleaware (4664)
Verbose
Main
Regex
Show only selected application
2019-02-17 21:44:53.652 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.702 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:44:53.706 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_PAUSE
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onPause
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_STOP
2019-02-17 21:45:32.181 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStop
2019-02-17 21:45:32.182 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_DESTROY
2019-02-17 21:45:32.182 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onDestroy
2019-02-17 21:45:32.231 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:45:32.231 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:45:32.232 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:45:32.232 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:45:32.236 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:45:32.236 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
Beim Drehen des Smartphones wird die Activity zerstört und neu erstellt
Install successful
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Install successful (a minute ago)
LifecycleAware
Hello World!


LifecycleAware [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/LifecycleAware] - .../app/src/main/java/at/htl/lifecycleaware/MainActivityObserver.kt [app]
Nexus 5X API 27
LifecycleAware app src main java at htl lifecycleaware MainActivityObserver.kt
activity_main.xml
MainActivity.kt
MainActivityObserver.kt
MainActivityObserver
Logcat
Emulator Nexus_5X_API_27 And
at.htl.lifecycleaware (4664)
Verbose
Main
Regex
Show only selected application
2019-02-17 21:44:53.652 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.653 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.654 4664-4664/at.htl.lifecycleaware I/zygote: at void at.htl.lifecycleaware.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:11)
2019-02-17 21:44:53.702 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:44:53.706 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:44:53.710 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:44:53.713 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_PAUSE
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onPause
2019-02-17 21:45:32.178 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_STOP
2019-02-17 21:45:32.181 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStop
2019-02-17 21:45:32.182 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_DESTROY
2019-02-17 21:45:32.182 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onDestroy
2019-02-17 21:45:32.231 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:45:32.231 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:45:32.232 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:45:32.232 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:45:32.236 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:45:32.236 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
2019-02-17 21:46:06.020 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_PAUSE
2019-02-17 21:46:06.020 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onPause
2019-02-17 21:46:06.020 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_STOP
2019-02-17 21:46:06.029 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStop
2019-02-17 21:46:06.029 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_DESTROY
2019-02-17 21:46:06.030 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onDestroy
2019-02-17 21:46:06.073 4664-4664/at.htl.lifecycleaware I/MainActivity: attached Observer to Owner - onCreate
2019-02-17 21:46:06.074 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_CREATE
2019-02-17 21:46:06.076 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onStart
2019-02-17 21:46:06.076 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_START
2019-02-17 21:46:06.079 4664-4664/at.htl.lifecycleaware I/MainActivity: Owner onResume
2019-02-17 21:46:06.079 4664-4664/at.htl.lifecycleaware I/MainActivityObserver: Observer ON_RESUME
Funktioniert auch beim zweiten Mal
Install successful
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Install successful (a minute ago)
LifecycleAware
Hello World!

Zusammenfassung

- Lifecycle-Aware Components reagieren auf Änderungen des Zustands anderer UI-Komponenten
  - Eintragen der Dependency und des kapt-Compilers
  - Erstellen des Observer mit seinen @OnLifecycleEvent-Methoden
  - Registrieren des Observer beim Owner (der Activity oder dem Fragment)
  - Überschreiben der On-Methoden beim Owner

States and Events



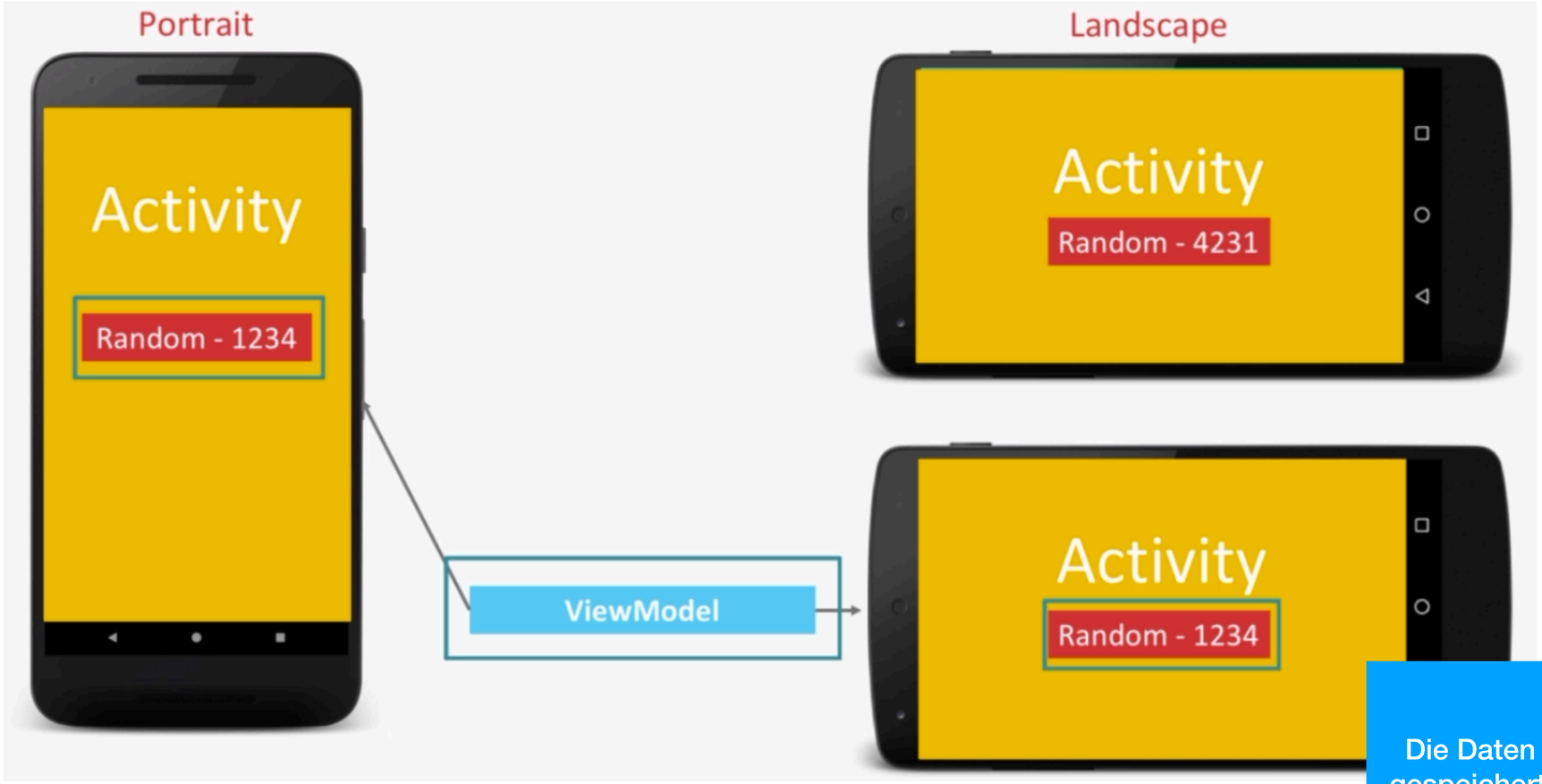
https://developer.android.com/topic/libraries/architecture/lifecycle

ViewModel

Problem

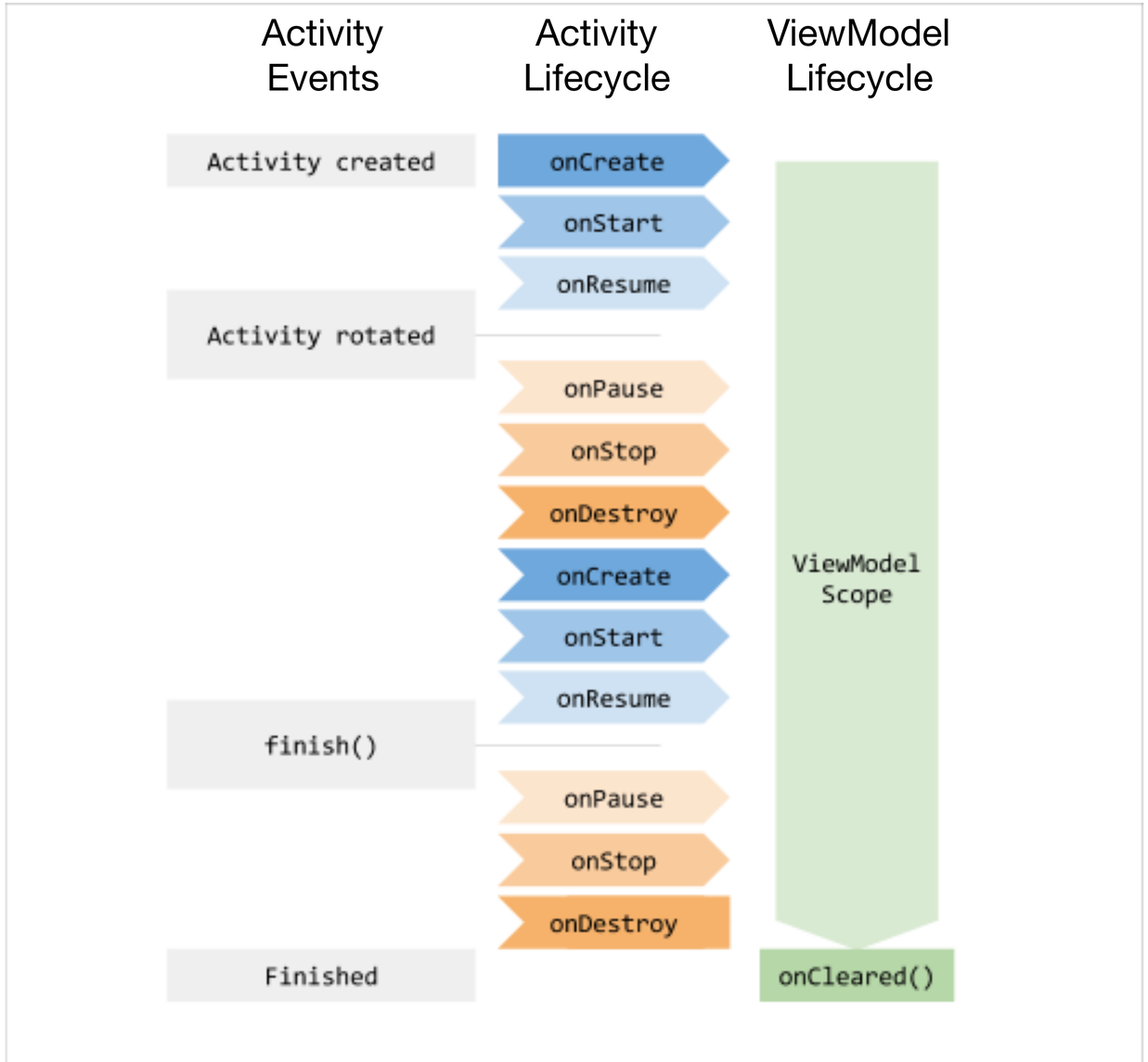
Bsp.: Eine Activity zeigt eine Zufallszahl an. Wird das Gerät nun gedreht, so wird die Activity zerstört und neu erstellt -> eine andere Zufallszahl wird angezeigt -> Die Daten sind verloren gegangen.

Lösung - ViewModel

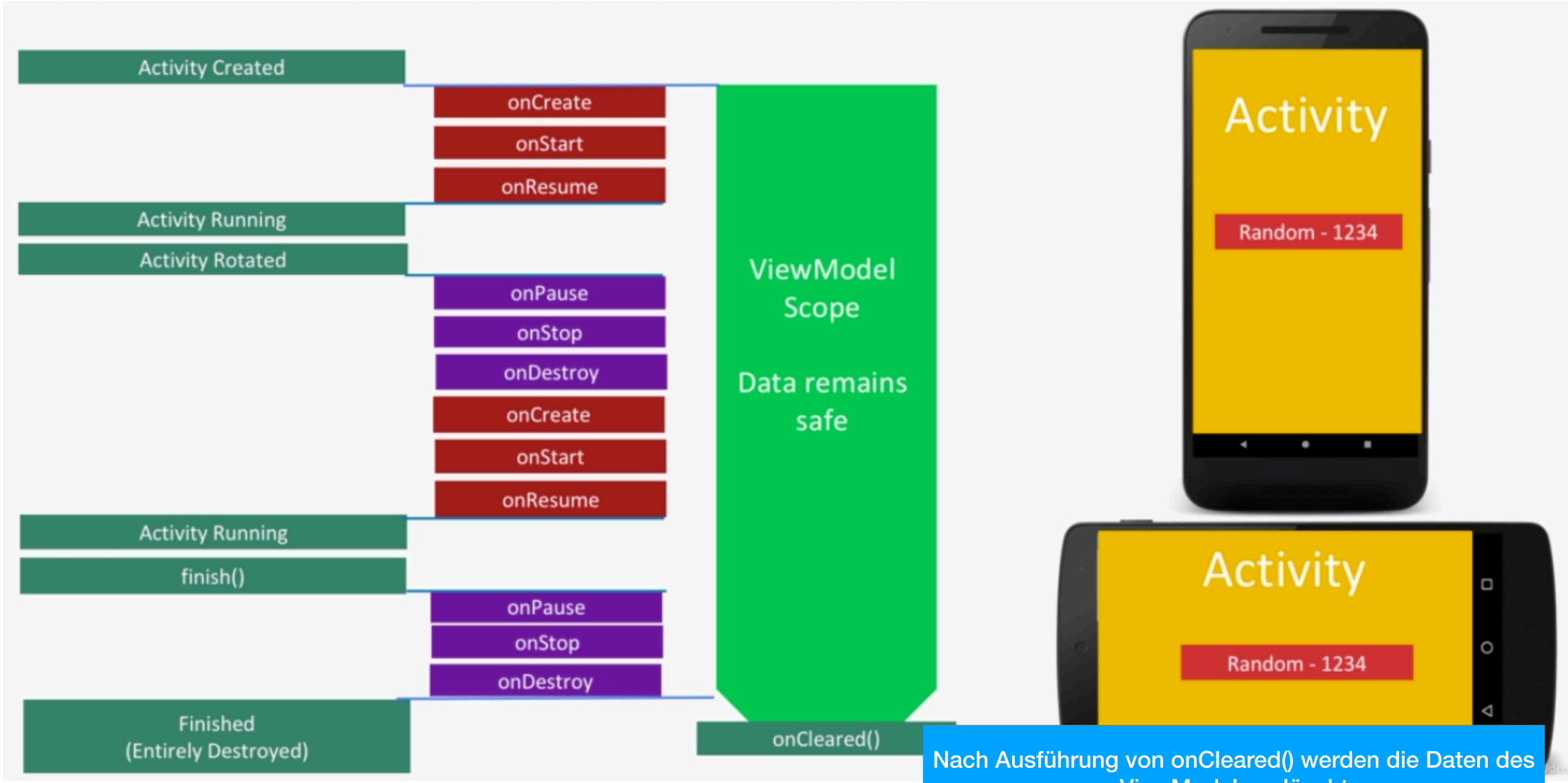


Die Daten werden im ViewModel gespeichert. Wird die neue Activity erstellt, so können diese gespeicherten Daten wiederverwendet werden.

Lebenszyklus

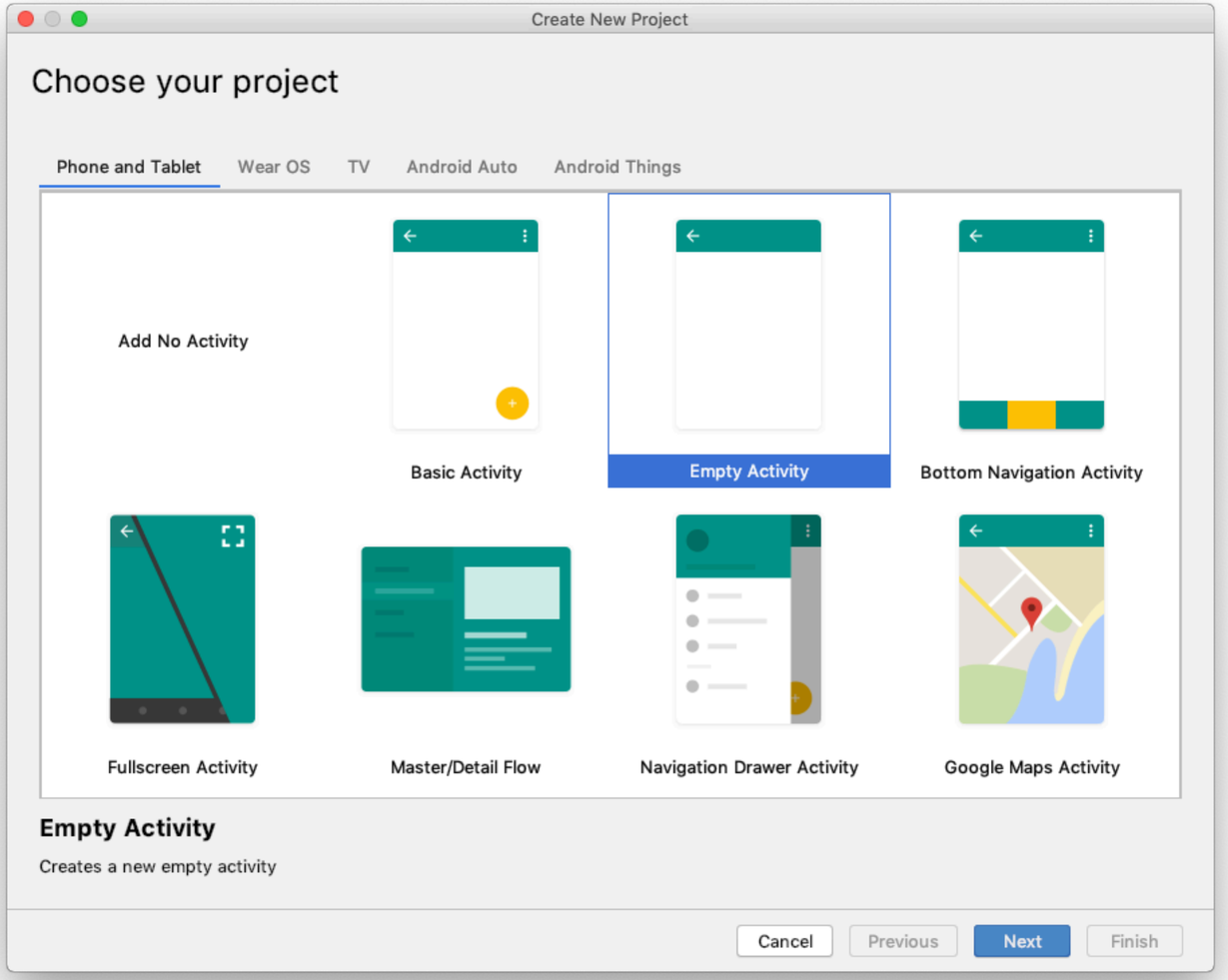


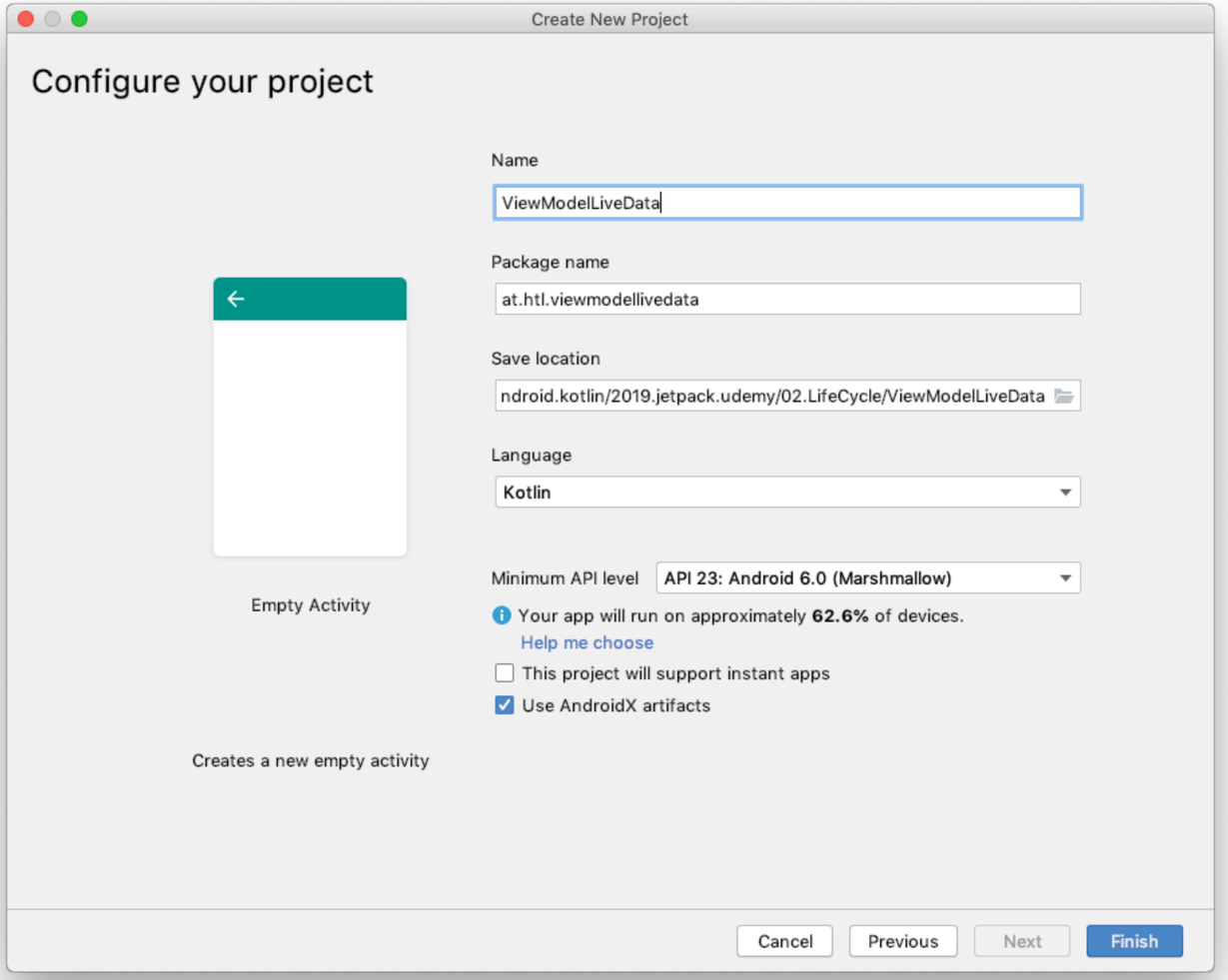
Beispiel

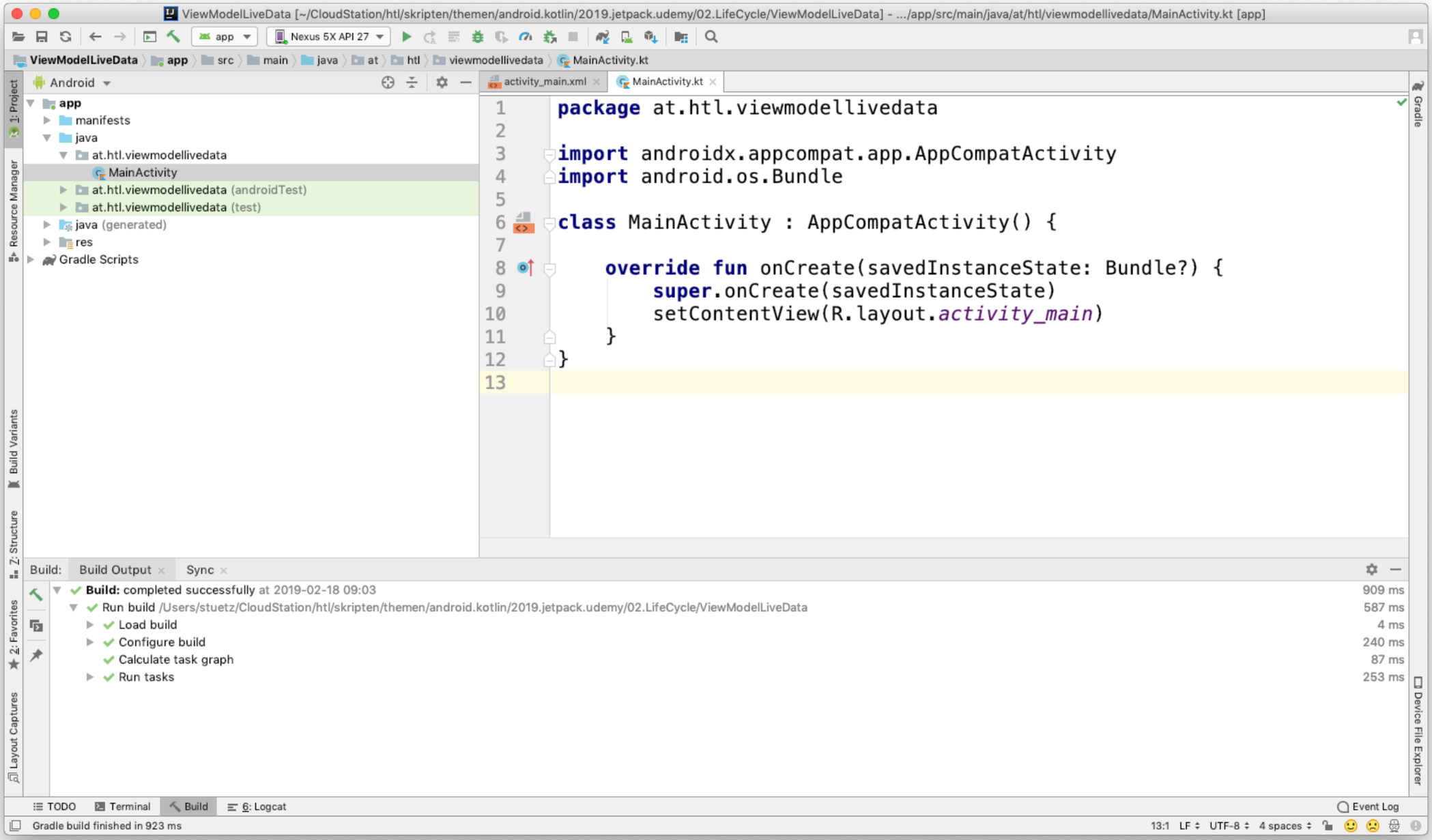


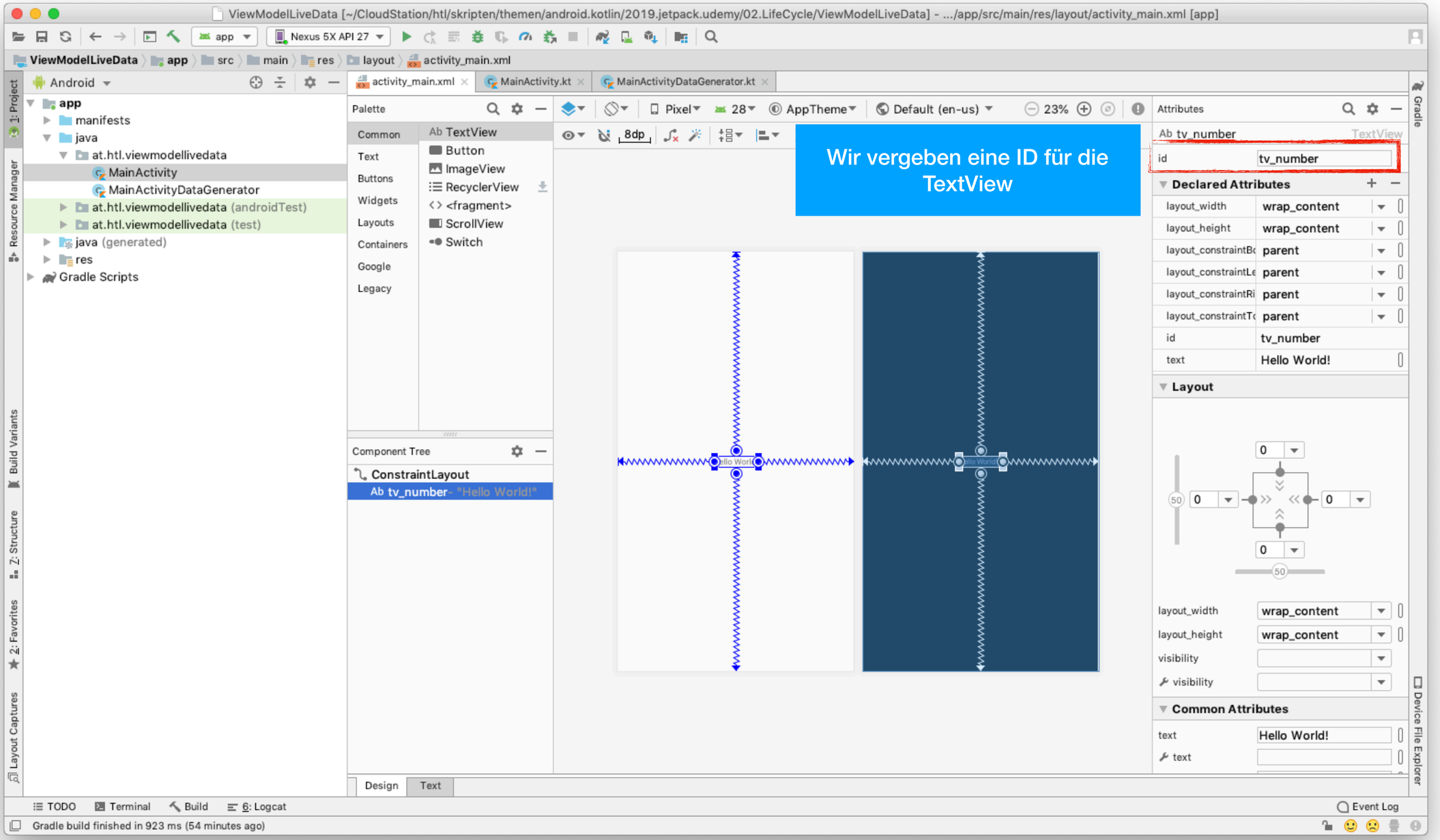
Nach Ausführung von onCleared() werden die Daten des ViewModels gelöscht

Variante 1: Ohne ViewModel

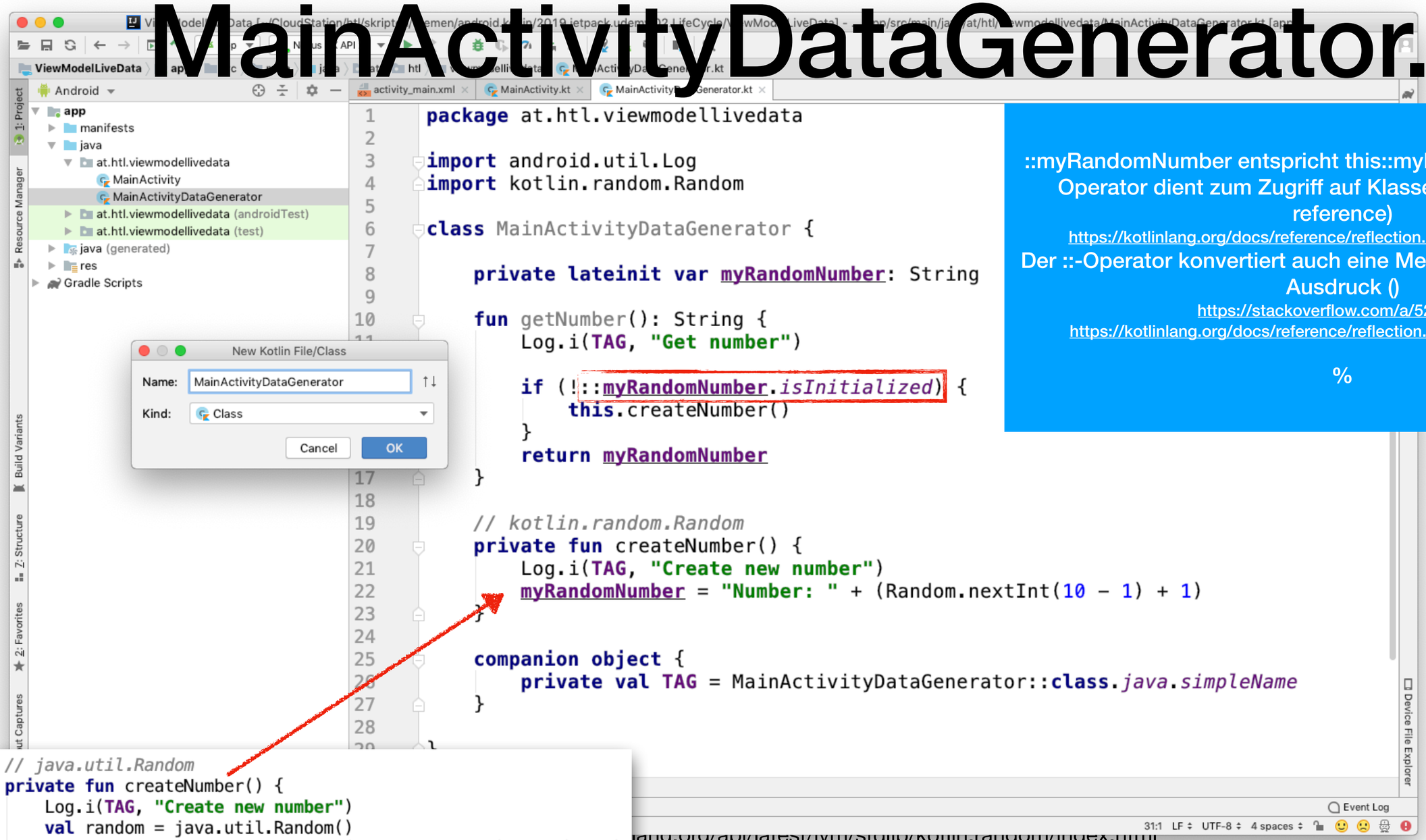

Create New Project
Choose your project
Phone and Tablet
Wear OS
TV
Android Auto
Android Things
Add No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Fullscreen Activity
Master/Detail Flow
Navigation Drawer Activity
Google Maps Activity
Empty Activity
Creates a new empty activity
Cancel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


Create New Project
Configure your project
Name
ViewModelLiveData
Package name
at.htl.viewmodellivedata
Save location
ndroid.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData
Language
Kotlin
Minimum API level
API 23: Android 6.0 (Marshmallow)
Your app will run on approximately 62.6% of devices.
Help me choose
This project will support instant apps
Use AndroidX artifacts
Empty Activity
Creates a new empty activity
Cancel
Previous
Next
Finish


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivity.kt [app]
app
Nexus 5X API 27
ViewModelLiveData
app
src
main
java
at
htl
viewmodellivedata
MainActivity.kt
Android
app
manifests
java
at.htl.viewmodellivedata
MainActivity
at.htl.viewmodellivedata (androidTest)
at.htl.viewmodellivedata (test)
java (generated)
res
Gradle Scripts
activity_main.xml
MainActivity.kt
1 package at.htl.viewmodellivedata
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7
8 override fun onCreate(savedInstanceState: Bundle?) {
9 super.onCreate(savedInstanceState)
10 setContentView(R.layout.activity_main)
11 }
12 }
13
Build:
Build Output
Sync
Build: completed successfully at 2019-02-18 09:03 909 ms
Run build /Users/stuetz/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData 587 ms
Load build 4 ms
Configure build 240 ms
Calculate task graph 87 ms
Run tasks 253 ms
TODO
Terminal
Build
6: Logcat
Event Log
Gradle build finished in 923 ms
13:1 LF UTF-8 4 spaces
1: Project
Resource Manager
Build Variants
7: Structure
2: Favorites
Layout Captures
Gradle
Device File Explorer


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/res/layout/activity_main.xml [app]
Nexus 5X API 27
ViewModelLiveData
app
src
main
res
layout
activity_main.xml
Android
app
manifests
java
at.htl.viewmodellivedata
MainActivity
MainActivityDataGenerator
at.htl.viewmodellivedata (androidTest)
at.htl.viewmodellivedata (test)
java (generated)
res
Gradle Scripts
activity_main.xml
MainActivity.kt
MainActivityDataGenerator.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
ConstraintLayout
Ab tv_number- "Hello World!"
Pixel
28
AppTheme
Default (en-us)
23%
8dp
Wir vergeben eine ID für die TextView
Attributes
Ab tv_number
TextView
id
tv_number
Declared Attributes
layout_width
wrap_content
layout_height
wrap_content
layout_constraintBo
parent
layout_constraintLe
parent
layout_constraintRi
parent
layout_constraintTo
parent
id
tv_number
text
Hello World!
Layout
layout_width
wrap_content
layout_height
wrap_content
visibility
visibility
Common Attributes
text
Hello World!
text
Design
Text
TODO
Terminal
Build
6: Logcat
Event Log
Gradle build finished in 923 ms (54 minutes ago)
1: Project
Resource Manager
Build Variants
7: Structure
2: Favorites
Layout Captures
Gradle
Device File Explorer

MainActivityDataGenerator.kt



::myRandomNumber entspricht this::myRandomNumber. Der ::-Operator dient zum Zugriff auf Klassenattribute (property reference)

https://kotlinlang.org/docs/reference/reflection.html#property-references

Der ::-Operator konvertiert auch eine Methode in einen Lambda-Ausdruck ()

https://stackoverflow.com/a/52463295

https://kotlinlang.org/docs/reference/reflection.html#function-references

%

```
// java.util.Random
private fun createNumber() {
    Log.i(TAG, "Create new number")
    val random = java.util.Random()
    myRandomNumber = "Number: " + (random.nextInt(10 - 1) + 1)
}
```

:: - Operator

Function References

When we have a named function declared like this:

```
fun isOdd(x: Int) = x % 2 != 0
```

We can easily call it directly (`isOdd(5)`), but we can also use it as a function type value, e.g. pass it to another function. To do this, we use the `::` operator:

```
val numbers = listOf(1, 2, 3)
println(numbers.filter(::isOdd))
```

Target platform: JVM Running on kotlin v. 1.3.21

Here `::isOdd` is a value of function type `(Int) -> Boolean` .

https://kotlinlang.org/docs/reference/reflection.html#function-references

Property References

To access properties as first-class objects in Kotlin, we can also use the `::` operator:

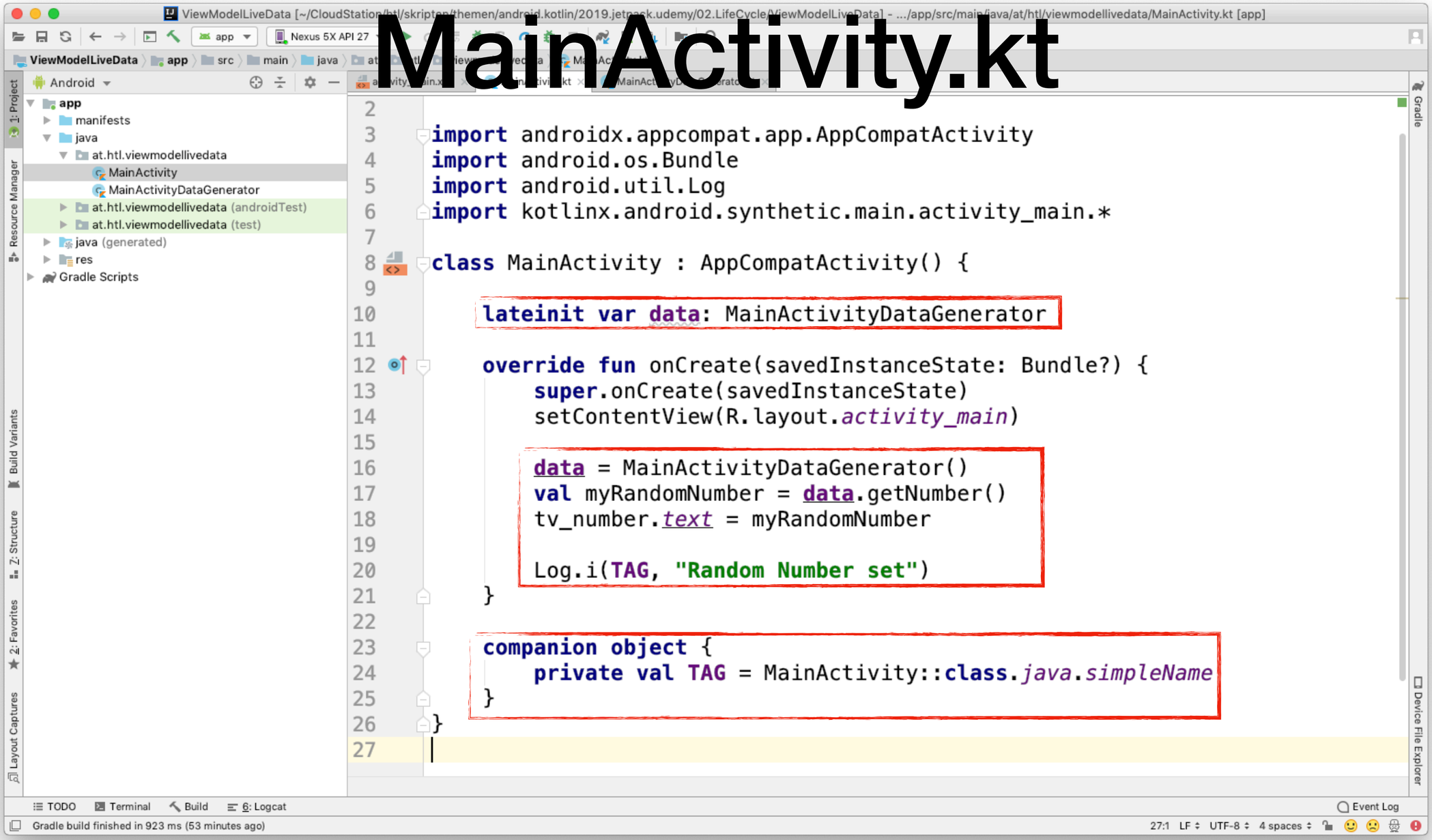
```
val x = 1

fun main() {
    println(::x.get())
    println(::x.name)
}
```

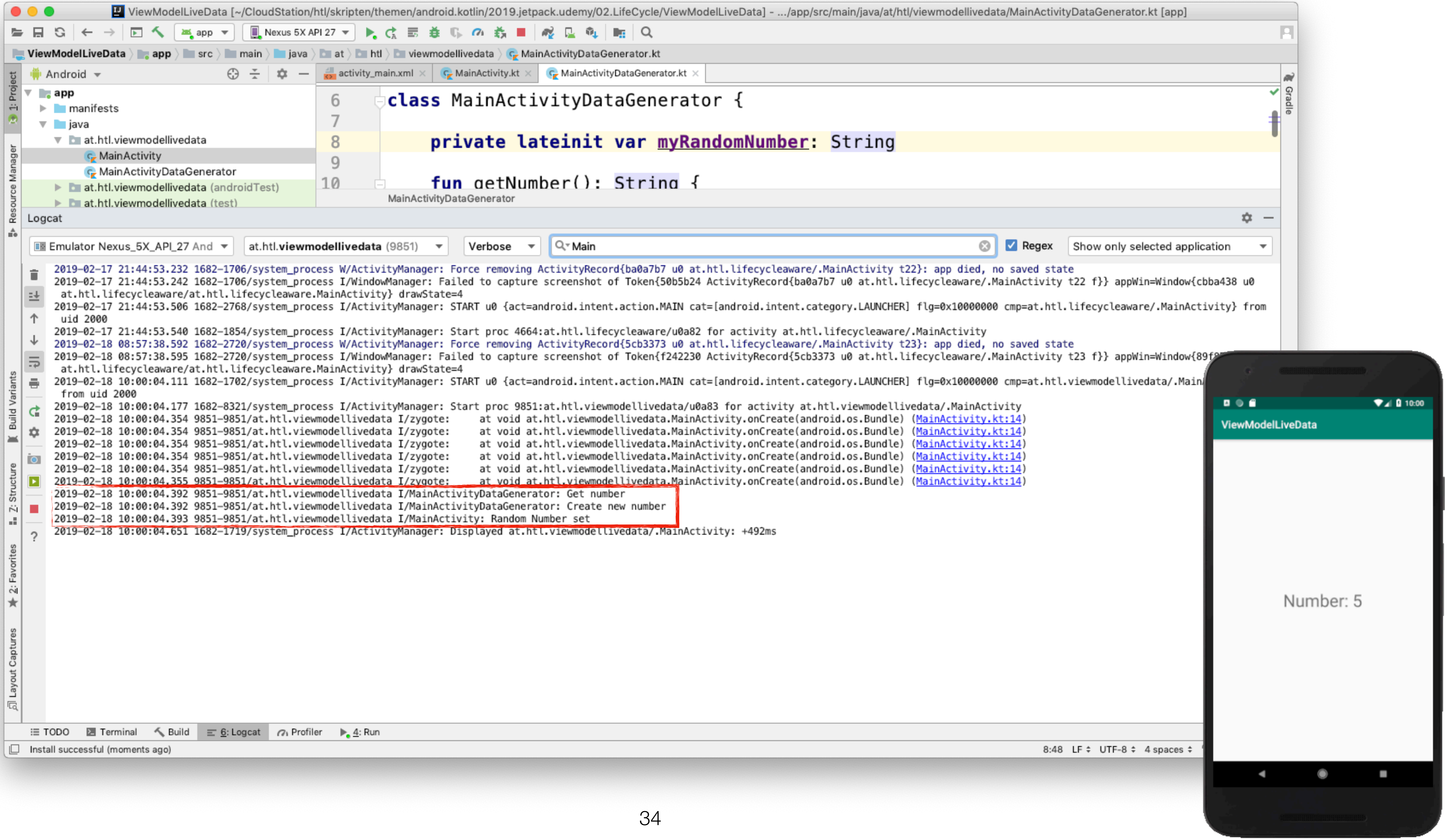
Target platform: JVM Running on kotlin v. 1.3.21

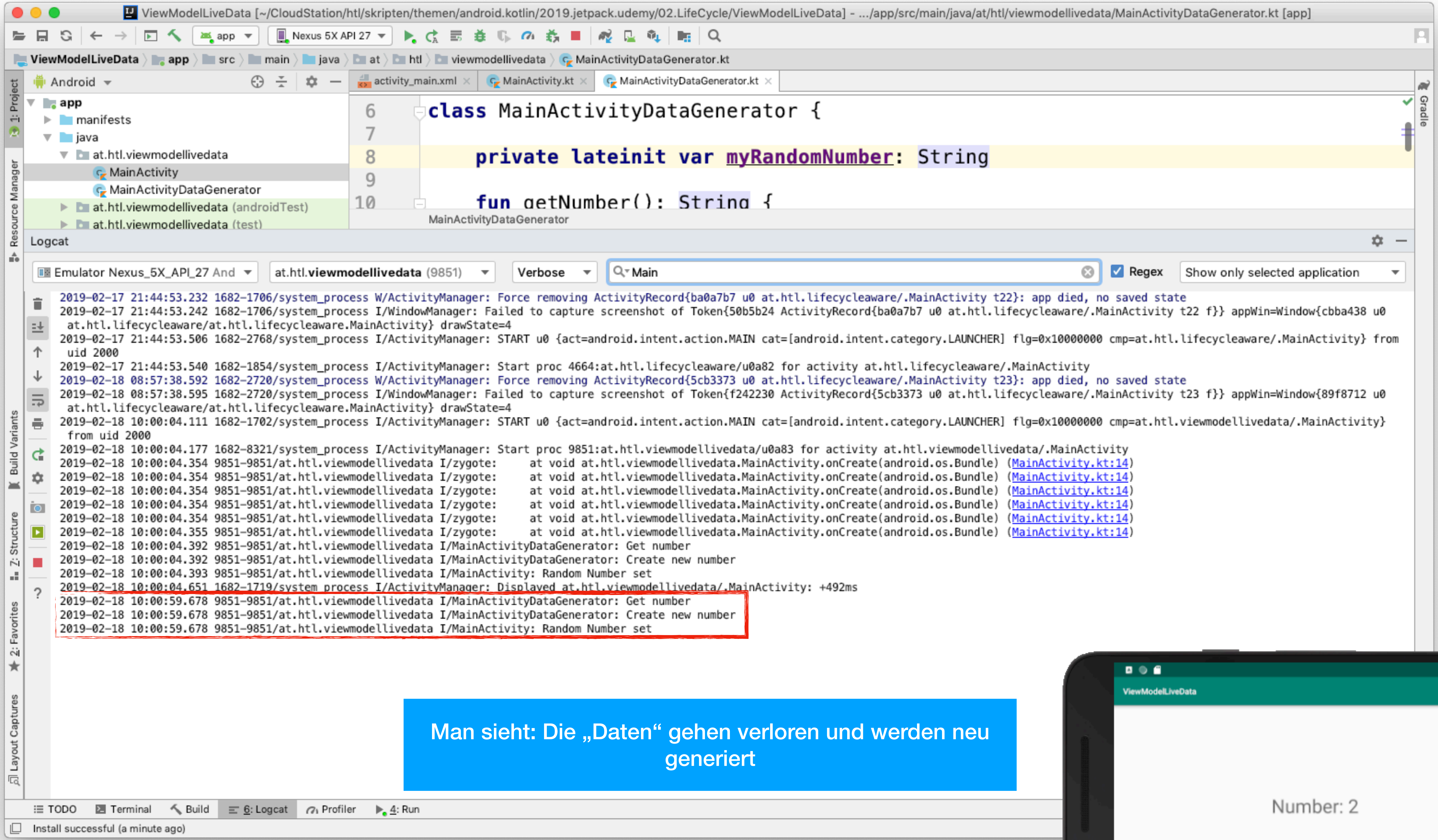
The expression `::x` evaluates to a property object of type `KProperty<Int>` , which allows us to read its value using `get()` or retrieve the property name using the `name` property. For more information, please refer to the docs on the `KProperty` class.

https://kotlinlang.org/docs/reference/reflection.html#property-references

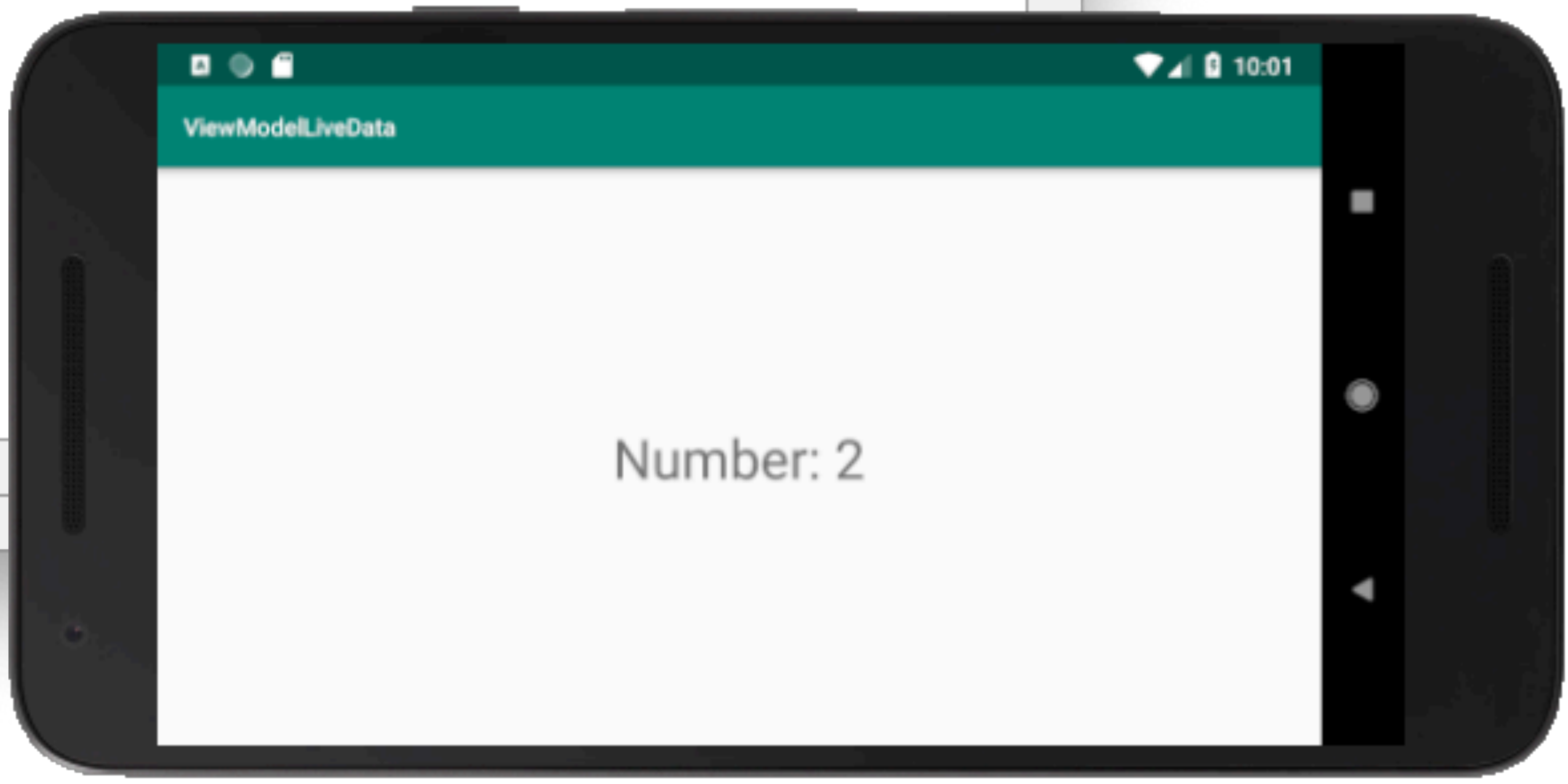
MainActivity.kt



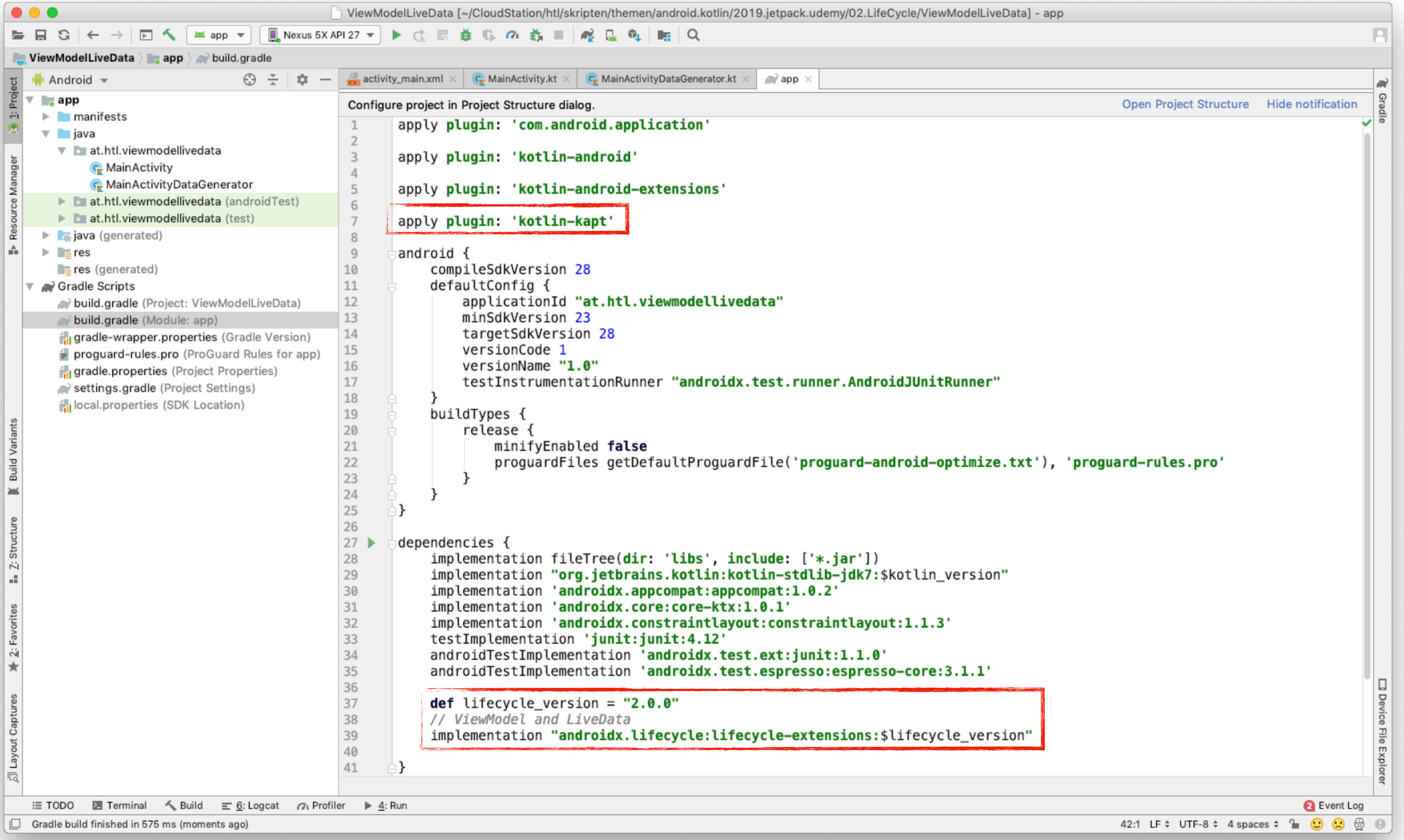

ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivityDataGenerator.kt [app]
ViewModelLiveData app src main java at htl viewmodellivedata MainActivityDataGenerator.kt
Android
app
manifests
java
at.htl.viewmodellivedata
MainActivity
MainActivityDataGenerator
at.htl.viewmodellivedata (androidTest)
at.htl.viewmodellivedata (test)
activity_main.xml
MainActivity.kt
MainActivityDataGenerator.kt
class MainActivityDataGenerator {
private lateinit var myRandomNumber: String
fun getNumber(): String {
Logcat
Emulator Nexus_5X_API_27 And
at.htl.viewmodellivedata (9851)
Verbose
Main
Regex
Show only selected application
2019-02-18 10:00:04.392 9851-9851/at.htl.viewmodellivedata I/MainActivityDataGenerator: Get number
2019-02-18 10:00:04.392 9851-9851/at.htl.viewmodellivedata I/MainActivityDataGenerator: Create new number
2019-02-18 10:00:04.393 9851-9851/at.htl.viewmodellivedata I/MainActivity: Random Number set
2019-02-18 10:00:04.651 1682-1719/system_process I/ActivityManager: Displayed at.htl.viewmodellivedata/.MainActivity: +492ms
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Install successful (moments ago)
ViewModelLiveData
Number: 5



Man sieht: Die „Daten“ gehen verloren und werden neu generiert



Variante 2: mit ViewModel




```
1  apply plugin: 'com.android.application'
2
3  apply plugin: 'kotlin-android'
4
5  apply plugin: 'kotlin-android-extensions'
6
7  apply plugin: 'kotlin-kapt'
8
9  android {
10     compileSdkVersion 28
11     defaultConfig {
12         applicationId "at.htl.viewmodellivedata"
13         minSdkVersion 23
14         targetSdkVersion 28
15         versionCode 1
16         versionName "1.0"
17         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
18     }
19     buildTypes {
20         release {
21             minifyEnabled false
22             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
23         }
24     }
25 }
26
27 dependencies {
28     implementation fileTree(dir: 'libs', include: ['*.jar'])
29     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
30     implementation 'androidx.appcompat:appcompat:1.0.2'
31     implementation 'androidx.core:core-ktx:1.0.1'
32     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
33     testImplementation 'junit:junit:4.12'
34     androidTestImplementation 'androidx.test.ext:junit:1.1.0'
35     androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
36
37     def lifecycle_version = "2.0.0"
38     // ViewModel and LiveData
39     implementation "androidx.lifecycle:lifecycle-extensions:$lifecycle_version"
40
41 }
```

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

Erstellen eines ViewModels

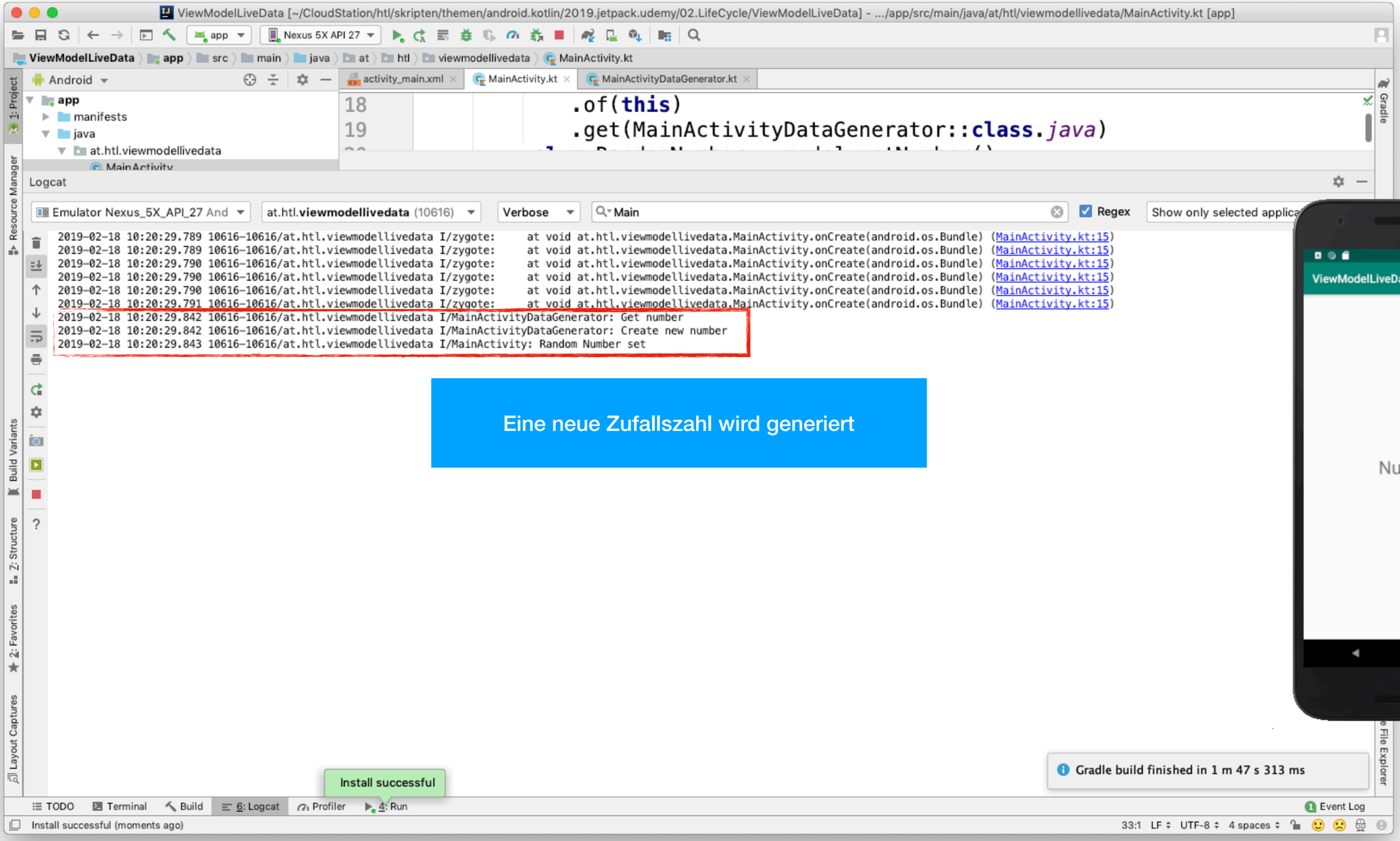
```
class MainActivityDataGenerator: ViewModel() {

    private lateinit var myRandomNumber: String

    fun getNumber(): String {
        Log.i(TAG, "Get number")

        if (!::myRandomNumber.isInitialized) {
            this.createNumber()
        }
        return myRandomNumber
    }

    // kotlin.random.Random
    private fun createNumber() {
        Log.i(TAG, "Create new number")
        myRandomNumber = "Number: " + (Random.nextInt(10 - 1) + 1)
    }

    companion object {
        private val TAG = MainActivityDataGenerator::class.java.simpleName
    }
}
```

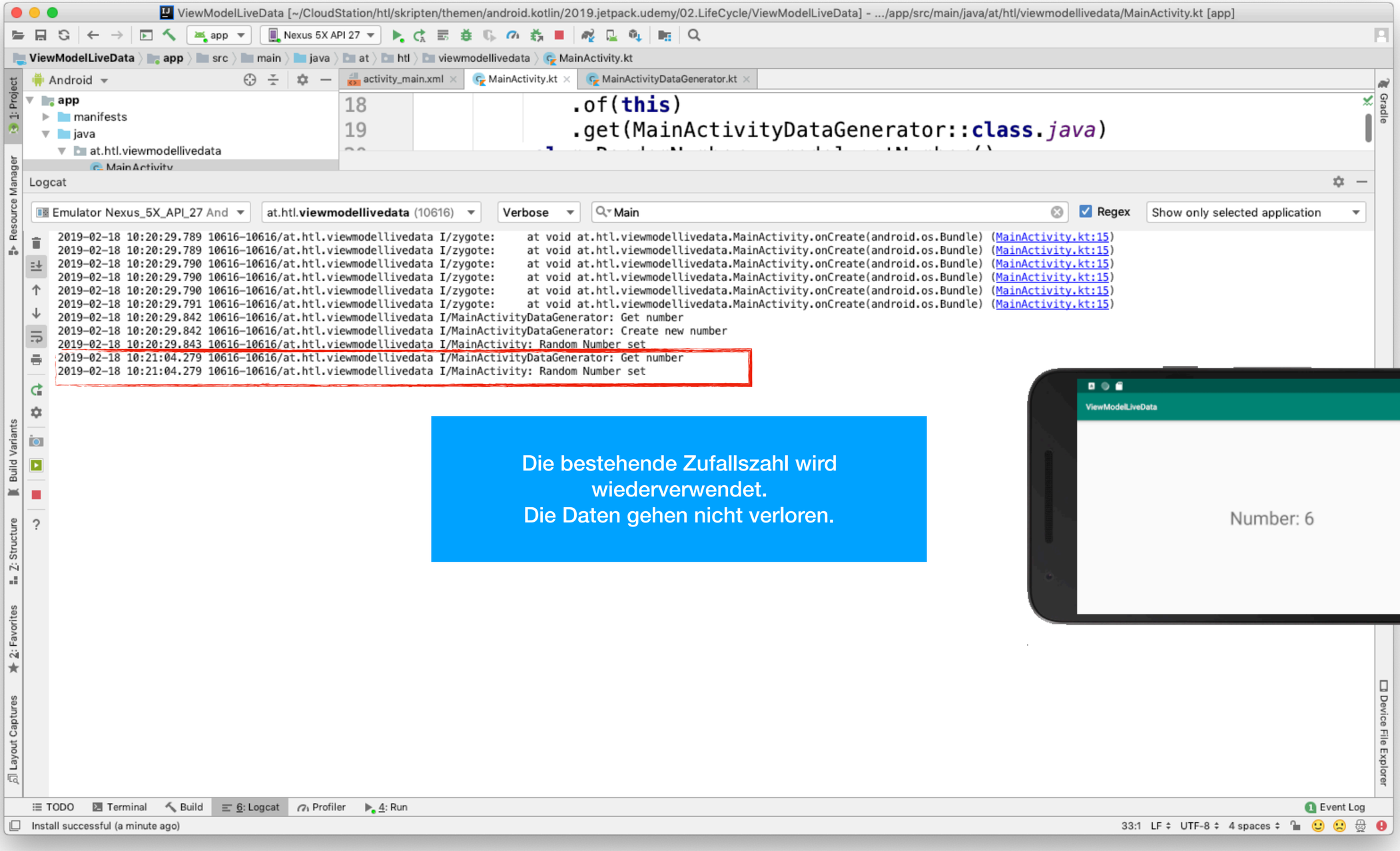
Wir verwenden den bereits bestehenden MainActivityDataGenerator als ViewModel

ViewModel in MainActivity verwenden

```
class MainActivity : AppCompatActivity() {

    // lateinit var data: MainActivityDataGenerator

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

        val model = ViewModelProviders
            .of(this)
            .get(MainActivityDataGenerator::class.java)
        val myRandomNumber = model.getNumber()

        //data = MainActivityDataGenerator()
        //val myRandomNumber = data.getNumber()
        tv_number.text = myRandomNumber

        Log.i(TAG, "Random Number set")
    }

    companion object {
        private val TAG = MainActivity::class.java.simpleName
    }
}
```

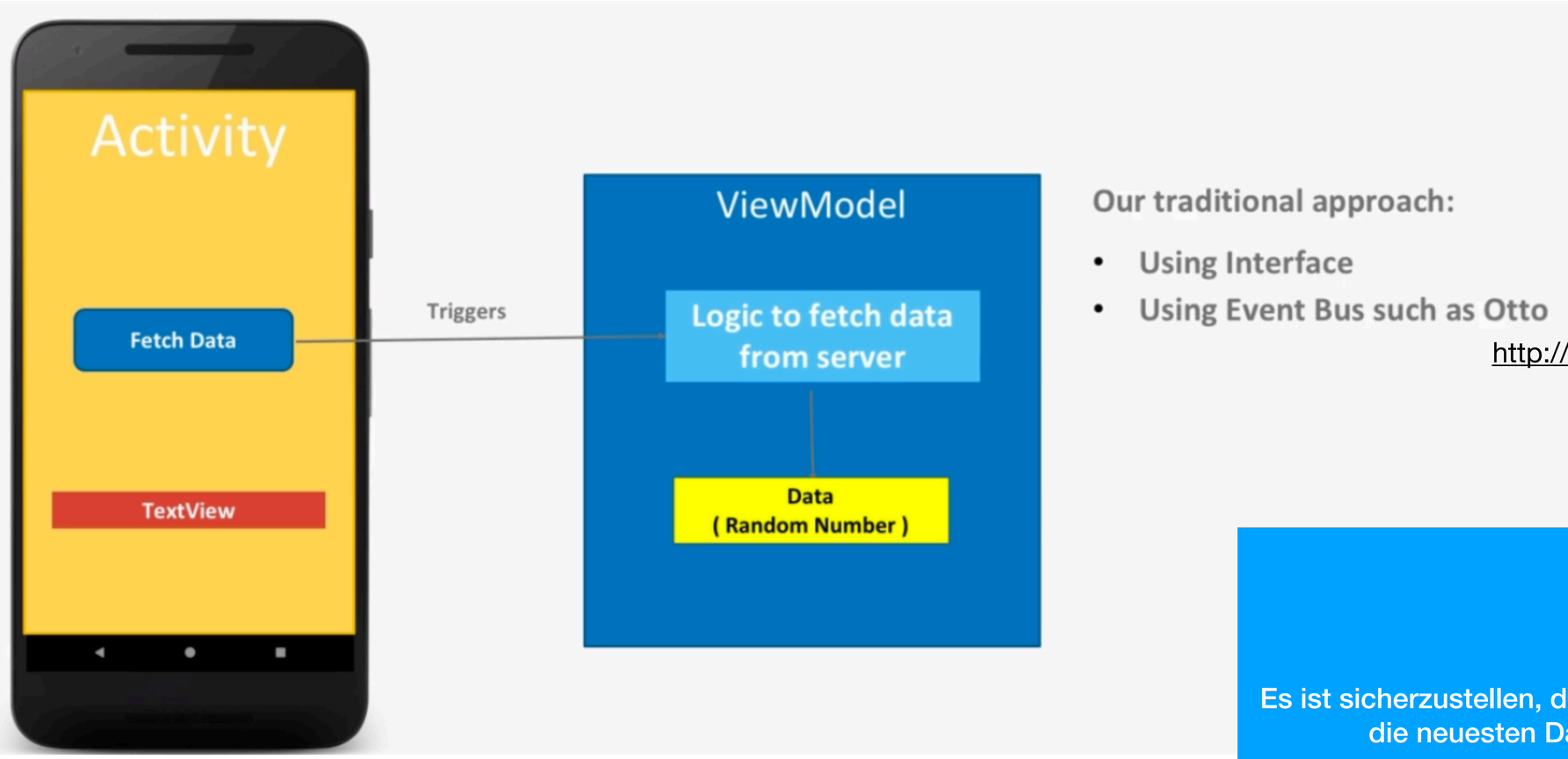
Beachte: Verwende ViewModelProviders


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivity.kt [app]
app
Nexus 5X API 27
ViewModelLiveData
app
src
main
java
at
htl
viewmodellivedata
MainActivity.kt
Android
activity_main.xml
MainActivity.kt
MainActivityDataGenerator.kt
app
manifests
java
at.htl.viewmodellivedata
MainActivity
18 .of(this)
19 .get(MainActivityDataGenerator::class.java)
Logcat
Emulator Nexus_5X_API_27 And
at.htl.viewmodellivedata (10616)
Verbose
Main
Regex
Show only selected applica
2019-02-18 10:20:29.789 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.789 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.791 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.842 10616-10616/at.htl.viewmodellivedata I/MainActivityDataGenerator: Get number
2019-02-18 10:20:29.842 10616-10616/at.htl.viewmodellivedata I/MainActivityDataGenerator: Create new number
2019-02-18 10:20:29.843 10616-10616/at.htl.viewmodellivedata I/MainActivity: Random Number set
Eine neue Zufallszahl wird generiert
ViewModelLiveData
Number: 6
Gradle build finished in 1 m 47 s 313 ms
Install successful
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Event Log
Install successful (moments ago)
33:1
LF
UTF-8
4 spaces


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivity.kt [app]
app
Nexus 5X API 27
ViewModelLiveData
app
src
main
java
at
htl
viewmodellivedata
MainActivity.kt
Android
activity_main.xml
MainActivity.kt
MainActivityDataGenerator.kt
app
manifests
java
at.htl.viewmodellivedata
MainActivity
18 .of(this)
19 .get(MainActivityDataGenerator::class.java)
Logcat
Emulator Nexus_5X_API_27 And
at.htl.viewmodellivedata (10616)
Verbose
Main
Regex
Show only selected application
2019-02-18 10:20:29.789 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.789 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.790 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.791 10616-10616/at.htl.viewmodellivedata I/zygote: at void at.htl.viewmodellivedata.MainActivity.onCreate(android.os.Bundle) (MainActivity.kt:15)
2019-02-18 10:20:29.842 10616-10616/at.htl.viewmodellivedata I/MainActivityDataGenerator: Get number
2019-02-18 10:20:29.842 10616-10616/at.htl.viewmodellivedata I/MainActivityDataGenerator: Create new number
2019-02-18 10:20:29.843 10616-10616/at.htl.viewmodellivedata I/MainActivity: Random Number set
2019-02-18 10:21:04.279 10616-10616/at.htl.viewmodellivedata I/MainActivityDataGenerator: Get number
2019-02-18 10:21:04.279 10616-10616/at.htl.viewmodellivedata I/MainActivity: Random Number set
Die bestehende Zufallszahl wird wiederverwendet.
Die Daten gehen nicht verloren.
10:21
ViewModelLiveData
Number: 6
TODO
Terminal
Build
6: Logcat
Profiler
4: Run
Event Log
Install successful (a minute ago)
33:1
LF
UTF-8
4 spaces

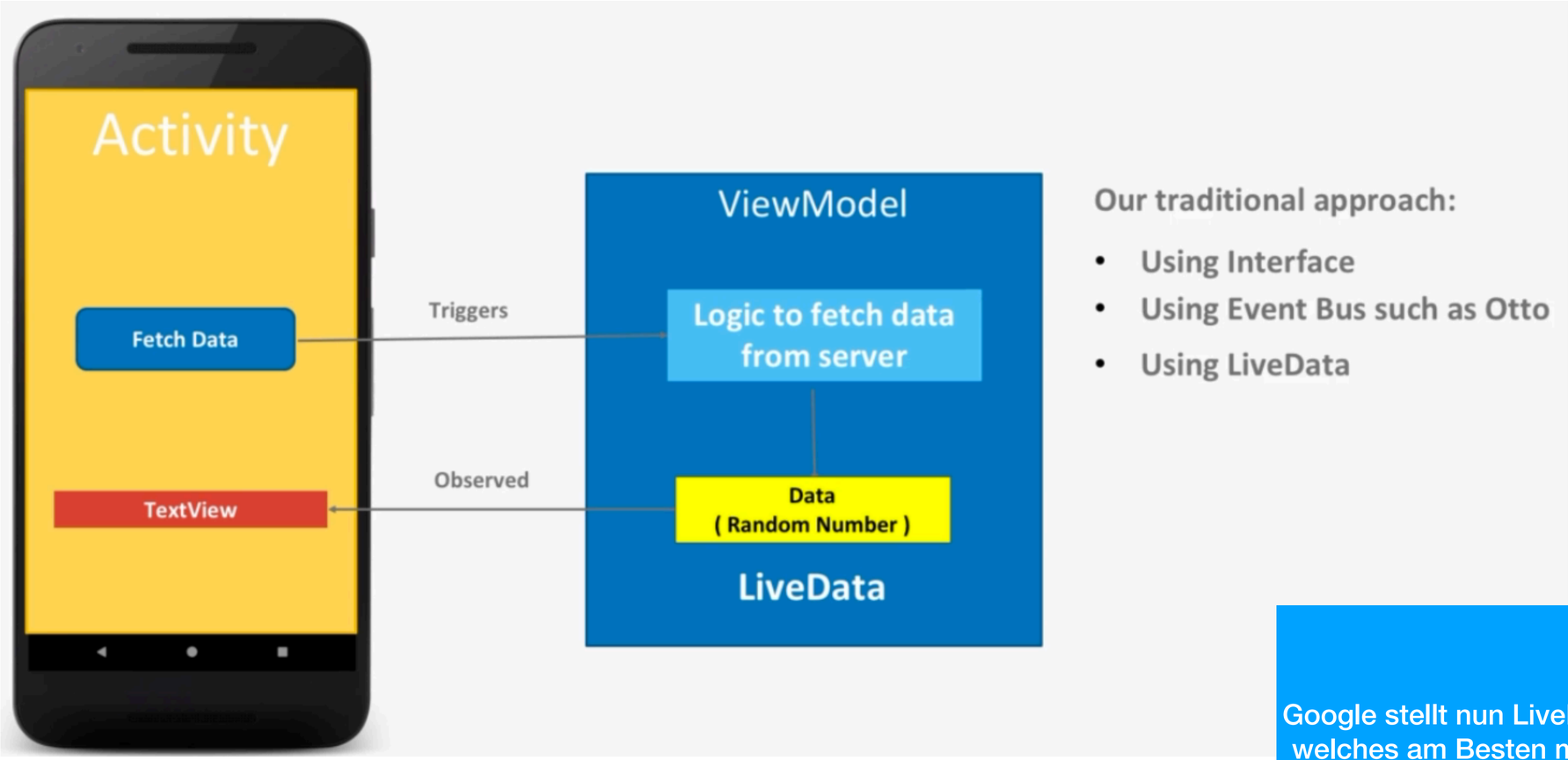
LiveData

Problem

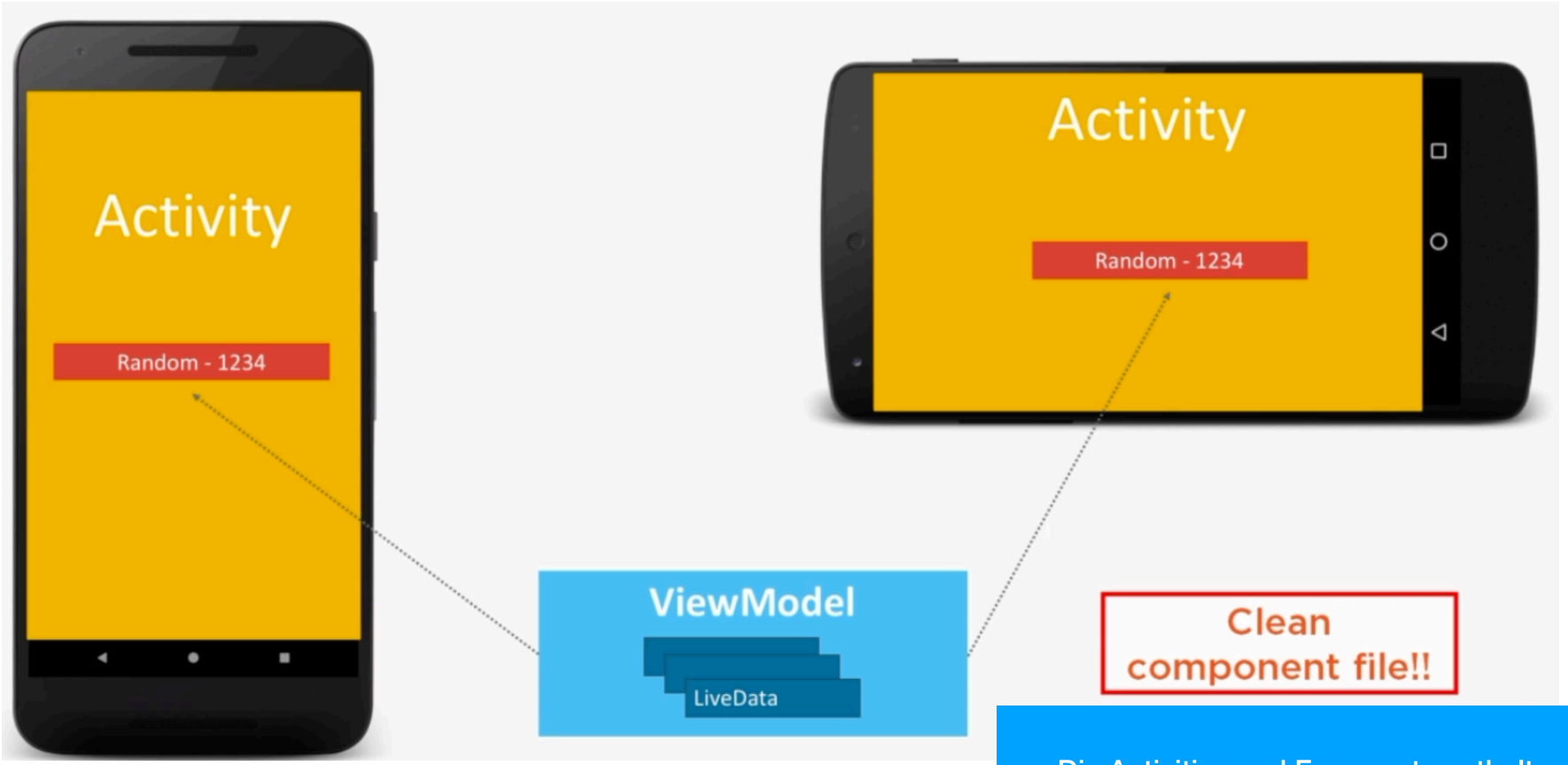


Es ist sicherzustellen, daß die Activity immer die neuesten Daten anzeigt.

Lösung



Google stellt nun LiveData zur Verfügung, welches am Besten mit dem ViewModel kombiniert wird.

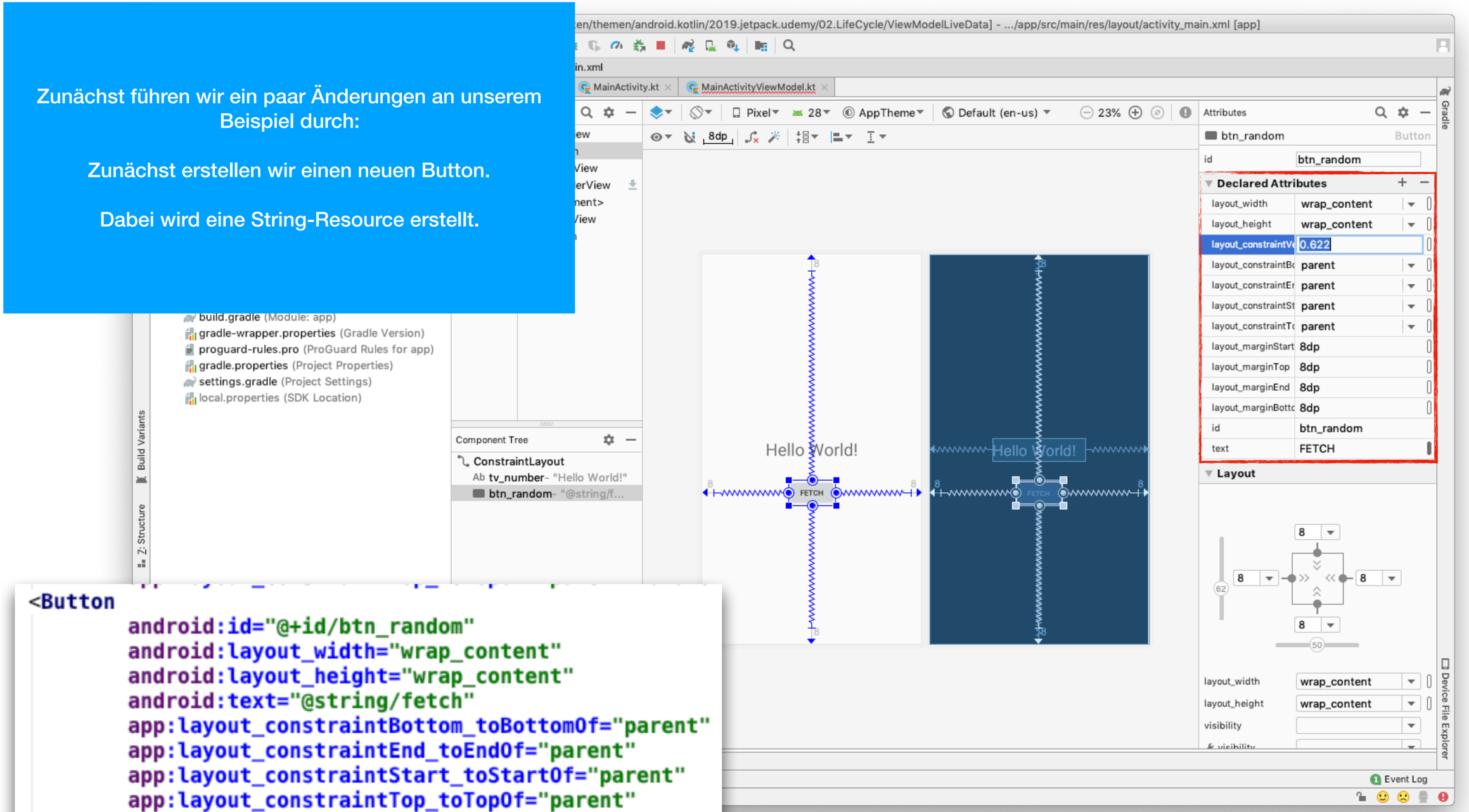


Die Activities und Fragments enthalten keinen Code zur Aktualisierung der Daten. Sie sind nur für die Anzeige der Daten verantwortlich

Zunächst führen wir ein paar Änderungen an unserem Beispiel durch:

Zunächst erstellen wir einen neuen Button.

Dabei wird eine String-Resource erstellt.



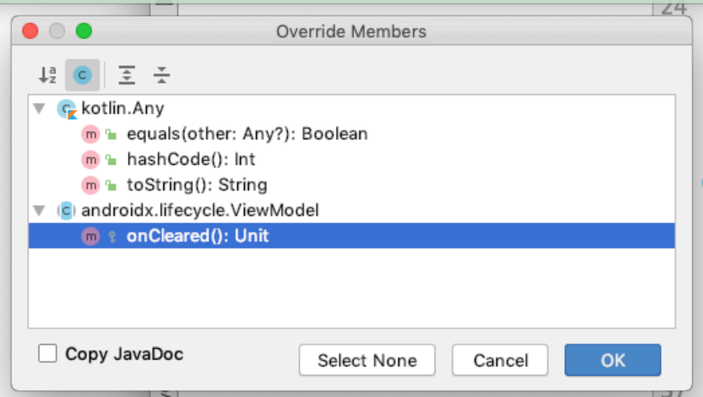

```
<Button
        android:id="@+id/btn_random"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/fetch"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.622"/>
```

MainActivityViewModel.kt

Wir verwenden MutableLiveData, da wir keine Zugriff auf Methoden freigeben, die die Daten ändern könnten. MutableLiveData wird von LiveData abgeleitet

```
package androidx.lifecycle;

/**
 * {@link LiveData} which publicly exposes {@link #setValue(T)}
 *
 * @param <T> The type of data hold by this instance
 */
/WeakerAccess/
public class MutableLiveData<T> extends LiveData<T> {
    @Override
    public void postValue(T value) { super.postValue(value); }

    @Override
    public void setValue(T value) { super.setValue(value); }
}
```

Override Methods via ^O (Ctrl+O for Win/Linux)

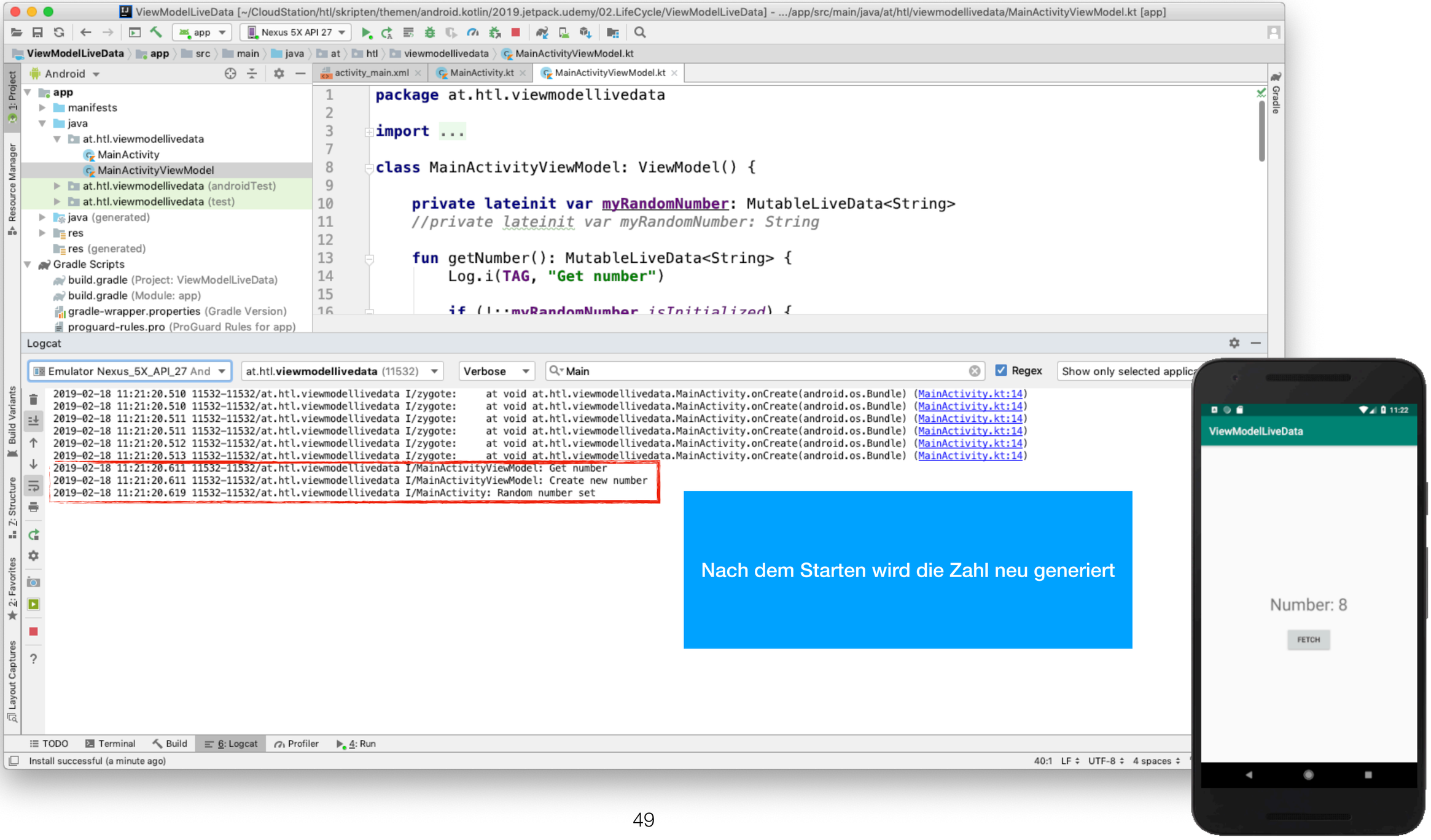


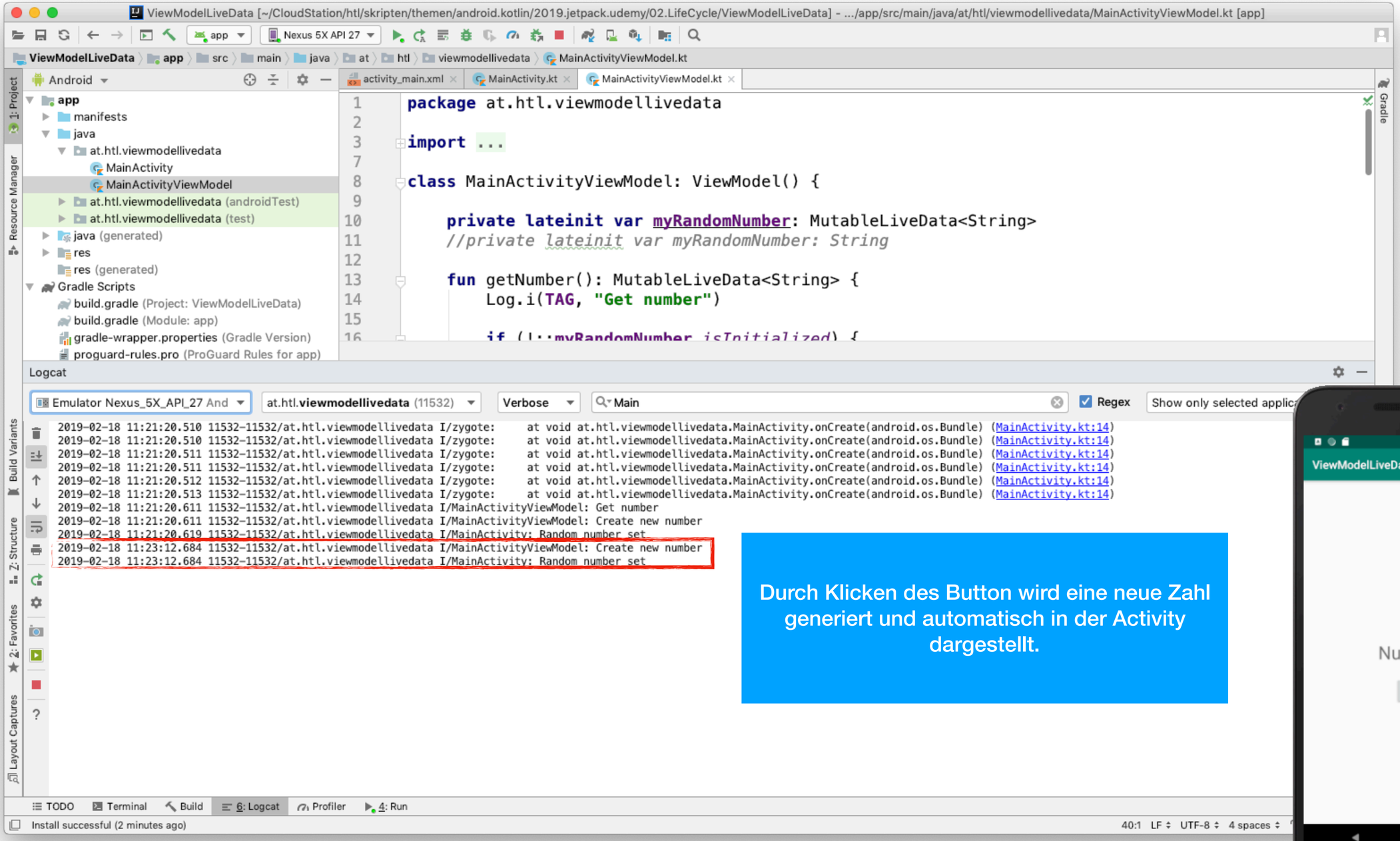
MainActivity.kt

```
package at.htl.viewmodellivedata

import ...

class MainActivity : AppCompatActivity() {

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

        val model = ViewModelProviders
            .of(this)
            .get(MainActivityViewModel::class.java)
        val myRandomNumber = model.getNumber()

        myRandomNumber.observe(this, Observer<String> { number ->
            tv_number.text = number
            Log.i(TAG, "Random number set")
        })

        btn_random.setOnClickListener {
            model.createNumber()
        }

    }

    companion object {
        private val TAG = MainActivity::class.java.simpleName
    }
}
```

die MainActivity beobachtet das ViewModel und aktualisiert die View

onClickListener für den Button


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivityViewModel.kt [app]
ViewModelLiveData › app › src › main › java › at › htl › viewmodellivedata › MainActivityViewModel.kt
activity_main.xml
MainActivity.kt
MainActivityViewModel.kt
package at.htl.viewmodellivedata
import ...
class MainActivityViewModel: ViewModel() {
private lateinit var myRandomNumber: MutableLiveData<String>
//private lateinit var myRandomNumber: String
fun getNumber(): MutableLiveData<String> {
Log.i(TAG, "Get number")
Logcat
2019-02-18 11:21:20.611 11532-11532/at.htl.viewmodellivedata I/MainActivityViewModel: Get number
2019-02-18 11:21:20.611 11532-11532/at.htl.viewmodellivedata I/MainActivityViewModel: Create new number
2019-02-18 11:21:20.619 11532-11532/at.htl.viewmodellivedata I/MainActivity: Random number set
Nach dem Starten wird die Zahl neu generiert
ViewModelLiveData
Number: 8
FETCH


ViewModelLiveData [~/CloudStation/htl/skripten/themen/android.kotlin/2019.jetpack.udemy/02.LifeCycle/ViewModelLiveData] - .../app/src/main/java/at/htl/viewmodellivedata/MainActivityViewModel.kt [app]
ViewModelLiveData › app › src › main › java › at › htl › viewmodellivedata › MainActivityViewModel.kt
activity_main.xml
MainActivity.kt
MainActivityViewModel.kt
package at.htl.viewmodellivedata
import ...
class MainActivityViewModel: ViewModel() {
private lateinit var myRandomNumber: MutableLiveData<String>
//private lateinit var myRandomNumber: String
fun getNumber(): MutableLiveData<String> {
Log.i(TAG, "Get number")
Logcat
2019-02-18 11:21:20.611 11532-11532/at.htl.viewmodellivedata I/MainActivityViewModel: Get number
2019-02-18 11:21:20.611 11532-11532/at.htl.viewmodellivedata I/MainActivityViewModel: Create new number
2019-02-18 11:21:20.619 11532-11532/at.htl.viewmodellivedata I/MainActivity: Random number set
2019-02-18 11:23:12.684 11532-11532/at.htl.viewmodellivedata I/MainActivityViewModel: Create new number
2019-02-18 11:23:12.684 11532-11532/at.htl.viewmodellivedata I/MainActivity: Random number set
Durch Klicken des Button wird eine neue Zahl generiert und automatisch in der Activity dargestellt.
ViewModelLiveData
Number: 4
FETCH

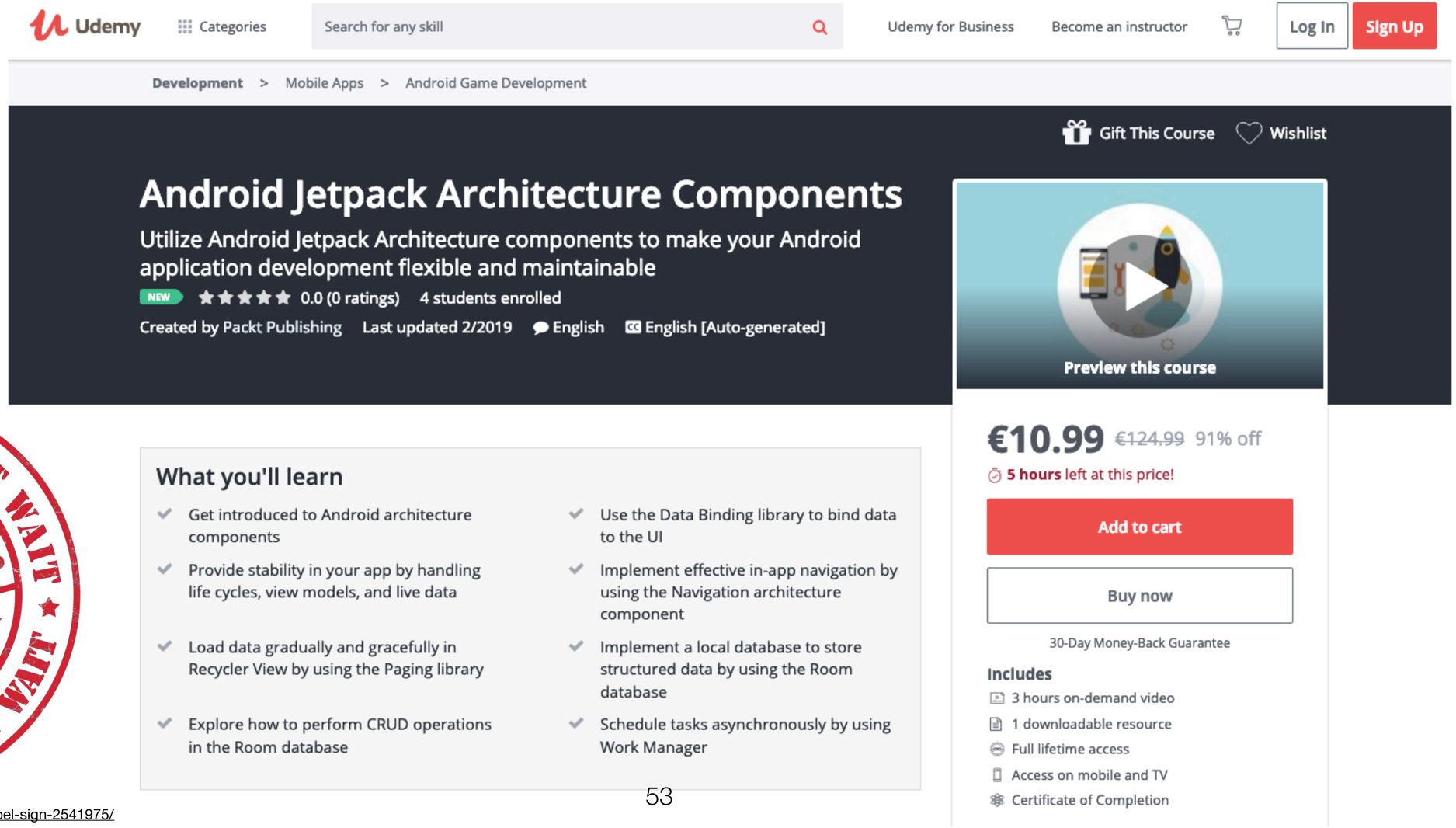
Wie gehts weiter?

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

Noch Fragen?

Source

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



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