

# Testing

With Kotlin

# Kotlin with Mockito

- In Kotlin: Classes and functions are final by default
- Mockito however needs to overwrite those classes and functions
- Mockito allows the following configuration to bypass this issue :
  - Im 'resources'-Ordner eine Datei erstellen: "mockito-extensions/org.mockito.plugins.MockMaker"
  - Inhalt des files: "mock-maker-inline"



# com.nhaarman.kotlin library

- More expressive writing of mocks. Examples:

```
this.repository = mock()
```

Creates a mock

```
verify(repository,times(2)).loaded(any())
```

Verifies how many times »loaded« was called

```
viewModel = mock<SomeViewModel>() {  
    on { isActive } doReturn true  
}
```

Mocks a certain behaviour

- Mockito returns null-values when called with »any«. The library covers this issue.

# Use Mockito with Kotlin

1. Add the mockito library to gradle.build: **"org.mockito:mockito-core:2.13.0»**
2. Add the nhaarman library to gradle.build:  
**"com.nhaarman.mockitokotlin2:mockito-kotlin: : 2.0.0-alpha03»**
3. Allow overwriting final classes with the MockMaker file described previously

# Test LiveData and MutableLiveData

1. Create a mock of an `android.arch.lifecycle.Observer`
2. Add this observer with «`observeForever`» to the `LivData` to be tested
3. If `MutableLiveData` is used, the main thread check needs to be bypassed with:

```
@get:Rule  
var rule:TestRule = InstantTaskExecutorRule()
```

# Testing Exercise

Look for "TODO:chapter\_03\_testing\_exercise"

Branch: chapter\_03\_testing\_exercise

# Quellen

- <https://medium.com/google-developers/7-steps-to-room-27a5fe5f99b2>
- <https://medium.com/google-developers/testing-room-migrations-be93cdb0d975>
- [https://pbochenski.pl/blog/07-12-2017-testing\\_livedata.html](https://pbochenski.pl/blog/07-12-2017-testing_livedata.html)
- [http://static.javadoc.io/org.mockito/mockito-core/2.12.0/org/mockito/Mockito.html#Mocking\\_Final](http://static.javadoc.io/org.mockito/mockito-core/2.12.0/org/mockito/Mockito.html#Mocking_Final)