Testing of Android Apps

David Bilík

Why testing?

How can you be sure that your code works?

You test it somehow if you write it.

But are you sure that you’ve covered all possible inputs / states of the app?

What if some dependency of your code changes. Are you sure that it still works?

What if somebody finds a bug in your app and you will fix it. Are you sure that it will not break again?

Why testing?

Testing should make you feel confident about your code.

Manual testing of test scenarios will not cover all possible states of api/system/user

Types of tests in Android ecosystem

Instrumented - requires Android DVM - runs on device/emulator

Unit Tests - runs locally on JVM

Instrumented tests

Usually for testing of UIs or extensive use of Android framework

Requires device or Android Emulator

- slower times due to apk installation
- slower times due to emulator starting
- slower times due to actual app start - activity init, transitions, …
- flaky tests due to emulator instability

Instrumented tests

Its not recommended to run instrumented tests on every merge to development due to its runtime

Services dedicated to device/emulator testings

- Firebase Test Lab
- Kobiton
- Amazon

Instrumented tests

Not used
history w

lue to

Instrumented tests

Currently investigating setup to run Instrumented tests on CI and apps modification to adapt these tests

Not part of this workshop 😞 not ready yet, someday in the future

Unit tests

Technically, even instrumented tests can be unit tests, however its more common to call JVM run tests Unit tests

These are simple java unit tests, that run locally on developers machine and does not require Android framework

Actually the Android framework is not present at all, tests has access only to android.jar to make code to compile but thats just an empty footprint of SDK. Calling something from this jar will ends up with exception or returning default values (depends on setup)

Unit tests

With that in mind the architecture of the app must be adapted to minimize the usage of Android framework.

Best practice is to use Android SDK only on View layer, that is not tested anyway (yet)

ackee

Unit tests

To make class testable it has to be designed well. It should not create any dependencies and every one should be provided via constructor or field injection via DI.

If we control creation of the object, we can pass dependencies via constructor and we dont need to worry in testing for DI setup. However if we dont control it (Nucleus Presenter), we need to setup Dagger graph with alternate dependencies and arrange somehow that this class is injected from the altered graph.

Mocking dependencies

But why should we mock dependencies? Wouldnt it make my class not “real”?

Good tests will simulate all possible behaviors that can happen and will test your class/function in all scenarios. Only way how to achieve this simulation is by mocking dependencies and providing our wanted behavior for given call.

Mocking dependencies

Also you dont want to rely on 3rd party services in your tests. If you will run your test and eg. api is down for some reason, your test will fail even if your implementation is 100% correct.

The same applies to testing of various data input. When I test signup, i dont want to call api and signup new users whenever my test run. I want to simulate this api call with mock

Mocking dependencies

Another reason for this is the relation to android framework. I dont want to have eg. Context in my Repository for retrieving strings because Context does not exist in unit testing.

ackee

Tools

JUnit - Testing framework that runs tests, contains set of asserts and allows to change behavior of app via Rules

Mockito - framework for mocking of classes, method calls, capturing of method parameters when called and verifying actual calls on mocked methods

Mockito-kotlin - Helper functions for Kotlin

Roboletric - Android framework “simulator” that uses actual implementation in android.jar instead of the empty one.

ackee

RxJava testing

TestObserver

If testing a method that returns Observable (or Single/Maybe/Completable), call .test() that returns TestObservable on which we can perform assertions

RxJava testing - async

RxJava enables easy asynchronous programming. But how to test in synchronous tests that your async code is ok?

If your Observable is guaranteed to end, you can call .awaitTerminalEvent() on TestObserver.

If you want to control the time continuum of Scheduler, there is TestScheduler

Otherwise you can force any scheduler to perform immediately with RxJavaPlugins, that can replace all schedulers with the provided one. There is Schedulers.trampoline() that ignores delays and perform synchronously.

ackee

Test structure

Three parts - Given/When/Then (Arrange/Act/Assert)

- Given - setup mocks/dependencies/state
- When - situation I want to test
- Then - asserts and checks that everything is as expected

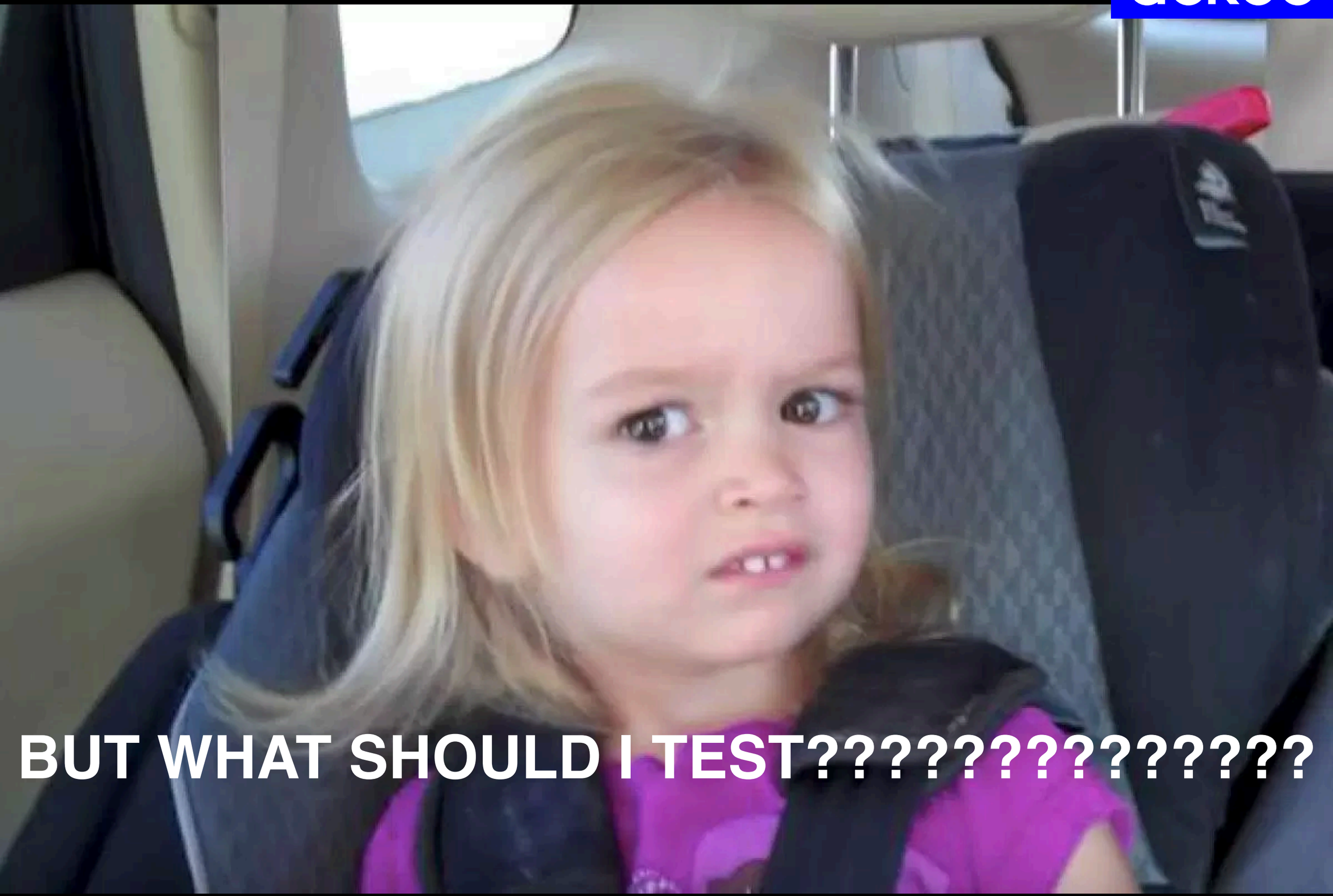
Naming - Sentence what should this test do like “Should log out user when 401 returns”. Because of Kotlin we can use spaces in test name (and even emoji 😒)

Length - As short as possible and as isolated as possible. Don’t test more than one scenarios - split it into multiple tests.

Test structure

Use the same programming techniques as everywhere else - OOP, functions, abstractions. For the sake of simplicity you can perform some code repetition but don’t be a 🐽 just because this is not an actual app code.

If a setup of a test takes too much time, step back a little and think about a design of your class. It's probably bad.

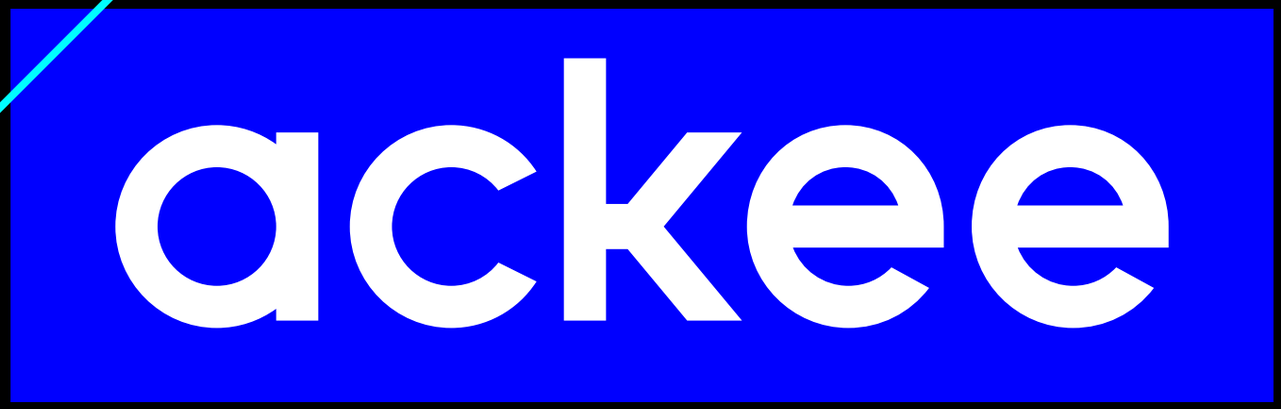

BUT WHAT SHOULD I TEST??????????????

What to test

Knowing what to test is a skill that can be mastered only by writing more tests.

Generally its something that contains even a little bit of logic like

- String formatting - Zonky and bank account number format
- Validation - sum of partial incomes must be lower than total income
- Transformation of data from api model to ui model
- Correct error handling - I don’t want HttpException in my Fragment or ViewModel, I want domain specific exception.
- State management - if User is logged out, I don’t want to make get/user request on API



Krásní lidé, krásné appky

www.ackee.cz