Automating Jetpack Compose app with Appium Espresso driver

Edvinas Liutvaitis

Flo is: an essential health partner.

You can:

Improve your **sexual health** and **enhance pleasure.**

Track your cycle with **impressive 2 days accuracy,** thanks to our AI based engine.

Explore your hormones, **learn** about hygiene and safety, **and break taboos.**

Engage Flo's Symptom Checker to avoid up to **12 years diagnostic delay** on medical conditions that affect up to 40% of women

Join **8M women who've conceived with the app. Manage pregnancy.** Plus, enjoy extra support from the Flo community.

Prepare for what your body is about to do, learn the underlying reasons and ways to alleviate, **stay in control.**

Jetpack Compose

Modern toolkit for building native android ui.
Why?

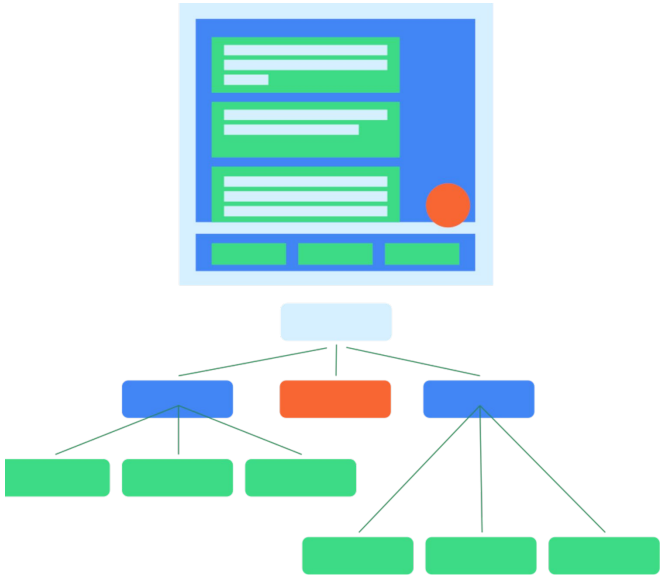
- Less Code
- No more XML-based layouts and views
  just one language
- Intuitive
- Declarative Approach
- Accelerate Development

Flo

Is, there any alternative in Appium?

Yes, there is- UiAutomator2 still useful

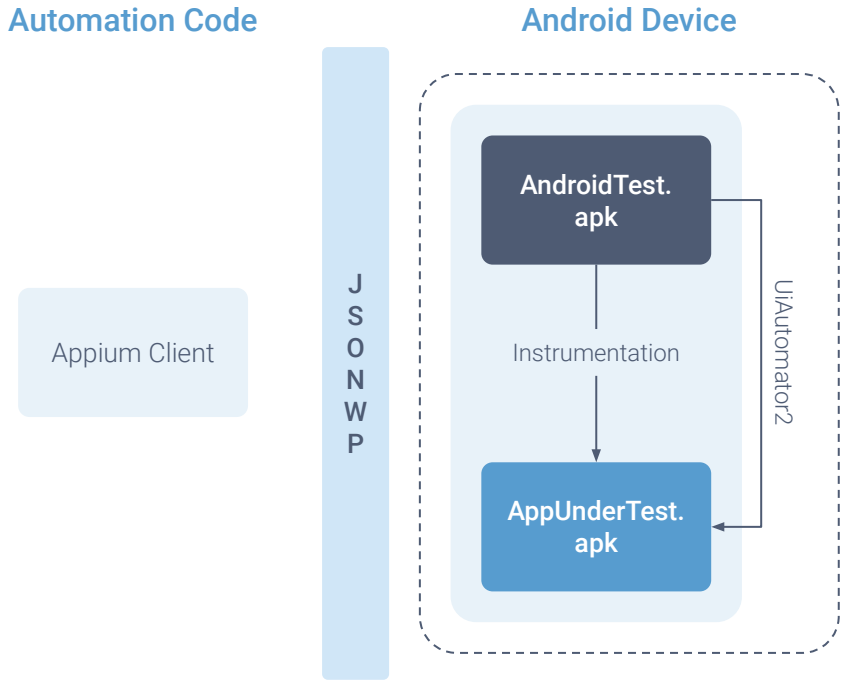
- Semantic tree structure helps accessibility service to understand what is on the screen.
- Jetpack compose provides semantic property called **testTag**
- Set `testTagsAsResourceId = true`



The Espresso way

How it works high-level overview

- Android device has 2 apks
  - One for driving the device under test (AndroidTest.apk)
  - Second one is App Under Test (AUT.apk)
- Driver.apk uses instrumentation when talking with AUT.apk



Espresso building gradle server

- Set dependencies for compose using `appium:espressoBuildConfig` capability
  - `composeVersion`
  - `additionalAppDependencies`
  - `additionalAndroidTestDependencies`

Espresso to Compose: Appium Settings API

- Use settings api to switch between compose and espresso context
    - **appium:settings[driver]: “compose”** allows interaction with Jetpack Compose based application interfaces
    - Can be switched back to espresso view at any time passing **“espresso”** keyword

Flo

Backdoor extension

Using backdoor extension

- Backdoor can be invoked in tree different locations
  - Application
  - Activity
  - Element

Flo

Things to know when using backdoor

- Only **public** methods can be invoked
- When using kotlin need to add **open** modifier
- If still not able to invoke the method check **proguard**

Demo