JETBRAINS

QA In Kotlin: From a Compiler Backend To the Entire Ecosystem

Alexander Zakharenko, Artur Parpibaev (JetBrains)

JETBRAINS

Artur Parpibaev

Kotlin Automation / Performance QA Lead
Cat owner 🐱 / Love mountains 🏔️ / Fan of Korean culture 🇰🇷

JETBRAINS

Alexander Zakharenko

Kotlin Compiler QA Lead

Wine lover 🍷 / Love singing 🎤 / Fan of Paul McCartney 🎸

Talk Structure

- What is Kotlin and its ecosystem
- From users to machine code
- Levels of testing
- How we measure quality
- Conclusions

What is Kotlin

What is Kotlin

- Announced in 2011 (JVM)
- Version 1.0 released in 2016
- Kotlin Multiplatform released in 2017
- Preferred language for Android app developers since 2019
- Used by Google, Amazon, Meta, Netflix, Uber, etc.

What is Kotlin and its ecosystem

What is Kotlin and its ecosystem

Compiler

What is Kotlin and its ecosystem

Compiler

Code editor

What is Kotlin and its ecosystem

Compiler

Code editor

Libraries

What is Kotlin and its ecosystem

Compiler

Code editor

Libraries

Build system

What is Kotlin and its ecosystem

Compiler

Kotlin Multiplatform

Code editor

Libraries

Build system

What is Kotlin and its ecosystem

Compiler

Kotlin Multiplatform

Code editor

Libraries

Build system

Documentation

What is Kotlin and its ecosystem

Compiler

Kotlin Multiplatform

Code editor

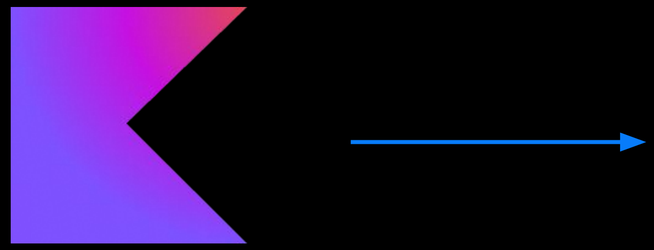
Libraries

Build system

Support

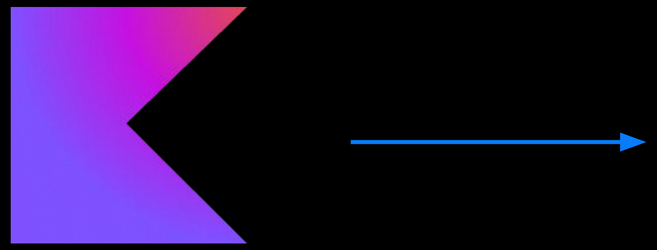
Documentation

What is Kotlin and its ecosystem



Compiler

What is Kotlin and its ecosystem

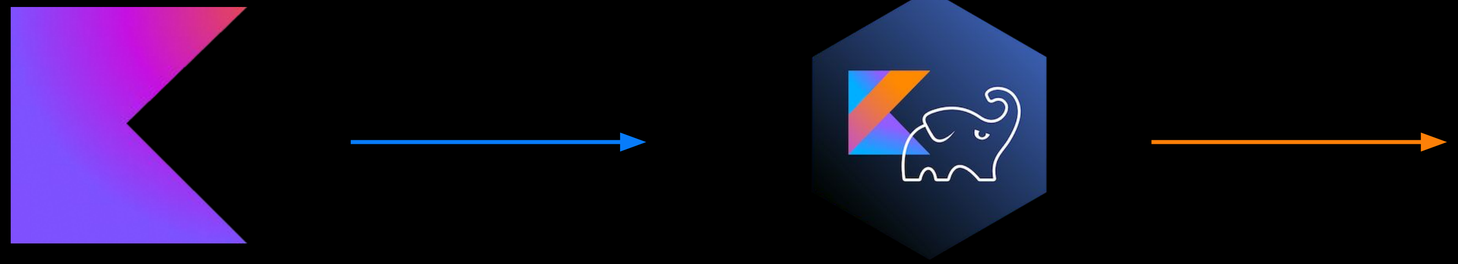


Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

```
➜  workdir cat main.kt
fun main() {
    println("Hello, Stockholm!")
    stockholm()
}
➜  workdir kotlinc-native -o main.kexe main.kt
main.kt:3:5: error: unresolved reference: stockholm
    stockholm()
    ^
```

What is Kotlin and its ecosystem

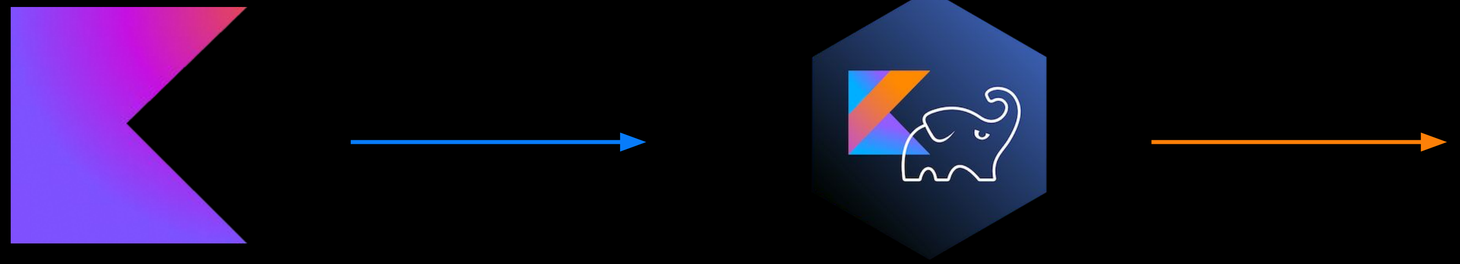


Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

What is Kotlin and its ecosystem



Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

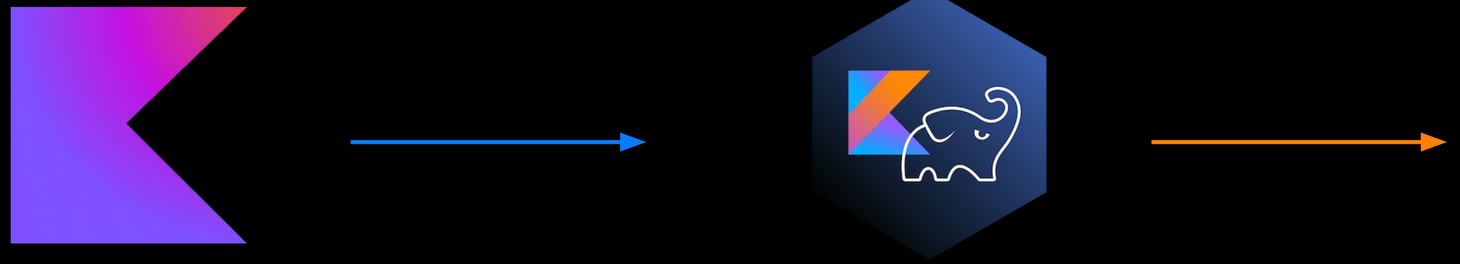
- Gradle / Maven support

```
➜  workdir kotlinc hello.kt -include-runtime -d
hello.jar
```

```
➜  workdir kotlinc-native -g -enable-assertions -module-name
org.example:enums -no-endorsed-libs -output enums.klib -produce library
-Xshort-module-name=enums -target macos_arm64 -Xmulti-platform hello.kt
➜  workdir kotlinc-native -g -enable-assertions -Xinclude=enums.klib
-no-endorsed-libs -output Enums.framework -produce framework -target
macos_arm64 -Xmulti-platform
➜  workdir clang -framework Foundation,Enums -F . -Xlinker -rpath
-Xlinker . -Xlinker Enums.framework/Versions/A/Enums main.m -o
enums_objc.kexe
```

```
➜  workdir ./gradlew build
```

What is Kotlin and its ecosystem



Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

- Gradle / Maven support
- Libraries

Kotlin

Type / to search

Overview | Repositories 116 | Projects 1 | Packages | People 31

- All
- Public
- Sources
- Forks
- Archived
- Templates

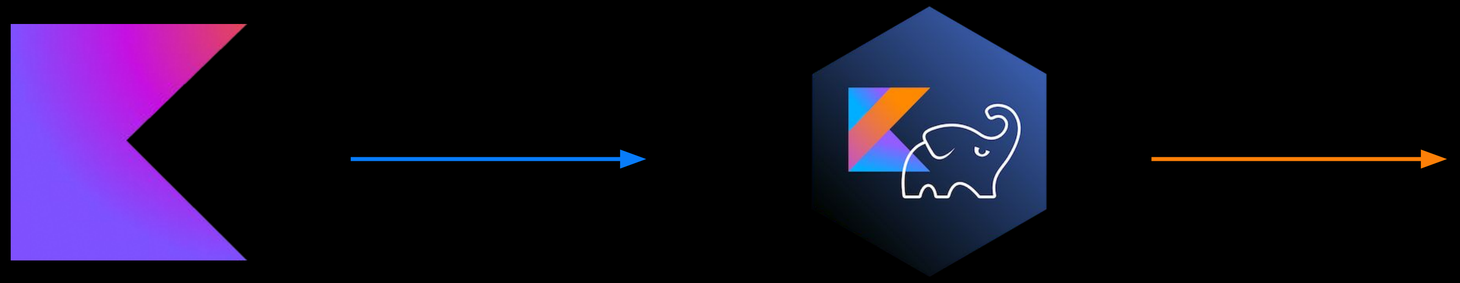
All

kotlinx sort:stars

18 repositories — Stars

kotlinx.coroutines	Library support for Kotlin coroutines	Kotlin	1.9k	14k
kotlinx.serialization	Kotlin multiplatform / multi-format serialization	Kotlin	660	5.8k
kotlinx-datetime	KotlinX multiplatform date/time library	Kotlin	121	2.7k
kotlinx.html	Kotlin DSL for HTML	Kotlin	134	1.7k
kotlinx-kover		Kotlin	54	1.5k
kotlinx-io	Kotlin multiplatform I/O library	Kotlin	69	1.5k
kotlinx.collections.immutable	Immutable persistent collections for Kotlin	Kotlin	64	1.3k
kotlinx-atomicfu	The idiomatic way to use atomic operations in Kotlin	Kotlin	69	1k
kotlinx-rpc	Add asynchronous RPC services to your multiplatform applications.	Kotlin	37	956
kotlinx-cli	Pure Kotlin implementation of a generic CLI parser.	Kotlin	71	928
kotlinx-benchmark	Kotlin multiplatform benchmarking toolkit	Kotlin	43	605
kotlinx-knit	Kotlin source code documentation management tool	Kotlin	19	310
kotlinx-nodejs	Kotlin external declarations for using the Node.js API from Kotlin code targeting JavaScript	Kotlin	19	219
kotlinx.reflect.lite	Lightweight library allowing to introspect basic stuff about Kotlin symbols	Kotlin	13	179
kotlinx-browser	Kotlin browser API	Kotlin	17	170

What is Kotlin and its ecosystem



Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

- Gradle / Maven support
- Libraries
- Dokka

kotlin-stdlib

Common JS JVM Native Wasm-JS Wasm-WASI

Kotlin Standard Library

The Kotlin Standard Library provides living essentials for everyday work with Kotlin. These include:

- Higher-order functions implementing idiomatic patterns (let, apply, kotlin.io.use, kotlin.synchronized, etc).
- Extension functions providing querying operations for collections (eager) and sequences (lazy).
- Various utilities for working with strings and char sequences.
- Extensions for JDK classes making it convenient to work with files, IO, and threading.

Packages

kotlin — Common JS JVM Native WASM-JS WASM-WASI

Core functions and types, available on all supported platforms.

kotlin.annotation — Common

Library support for the Kotlin annotation facility.

kotlin.collections — Common JS JVM Native WASM-JS WASM-WASI

Collection types, such as Iterable, Collection, List, Set, Map and related top-level and extension functions.

What is Kotlin and its ecosystem

Compiler

- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

- Gradle / Maven support
- Libraries
- Dokka

IDE Plugin

What is Kotlin and its ecosystem

Compiler

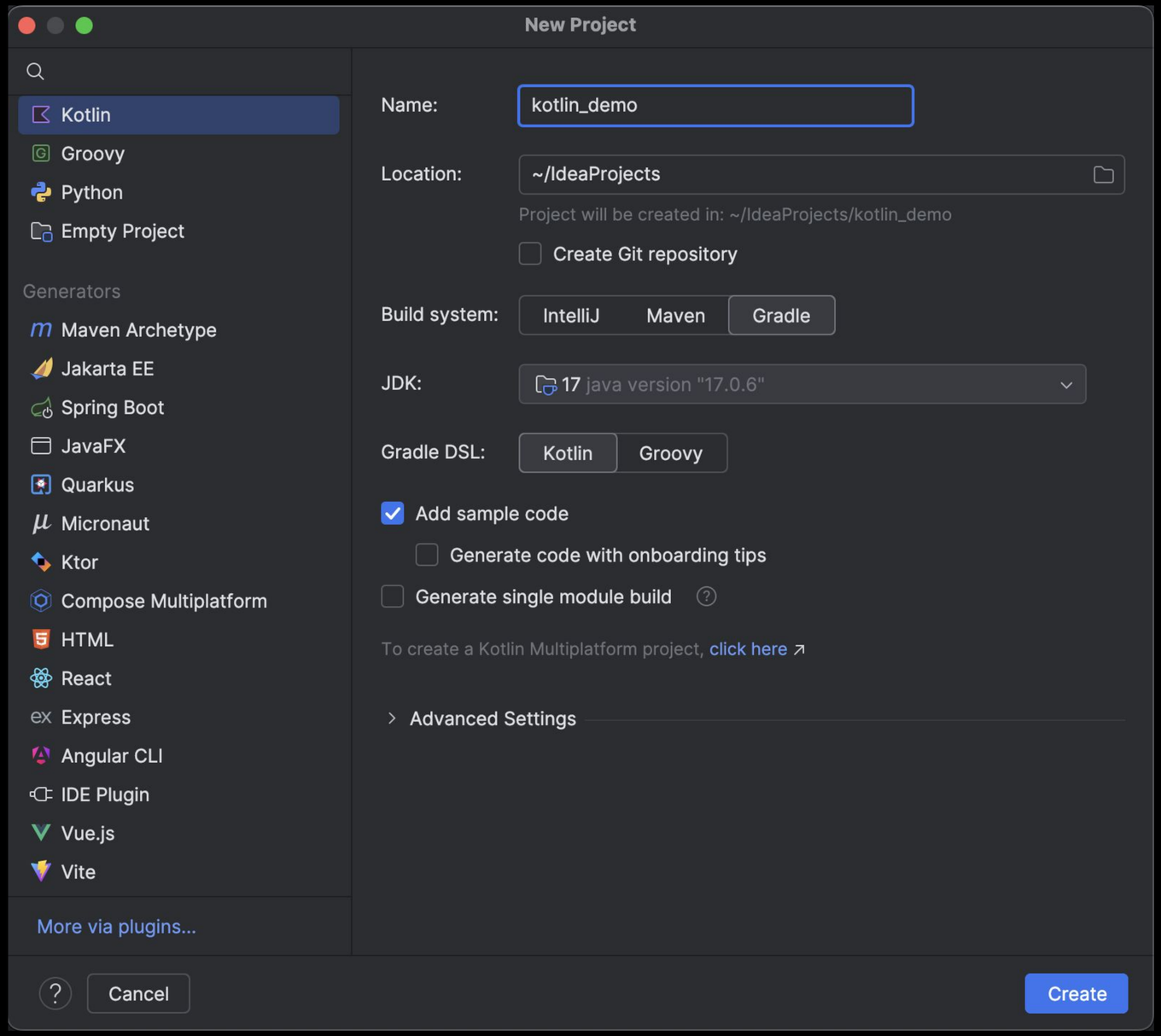
- Kotlin / JVM
- Kotlin / JS
- Kotlin / Native
- Kotlin / Wasm

Core Ecosystem

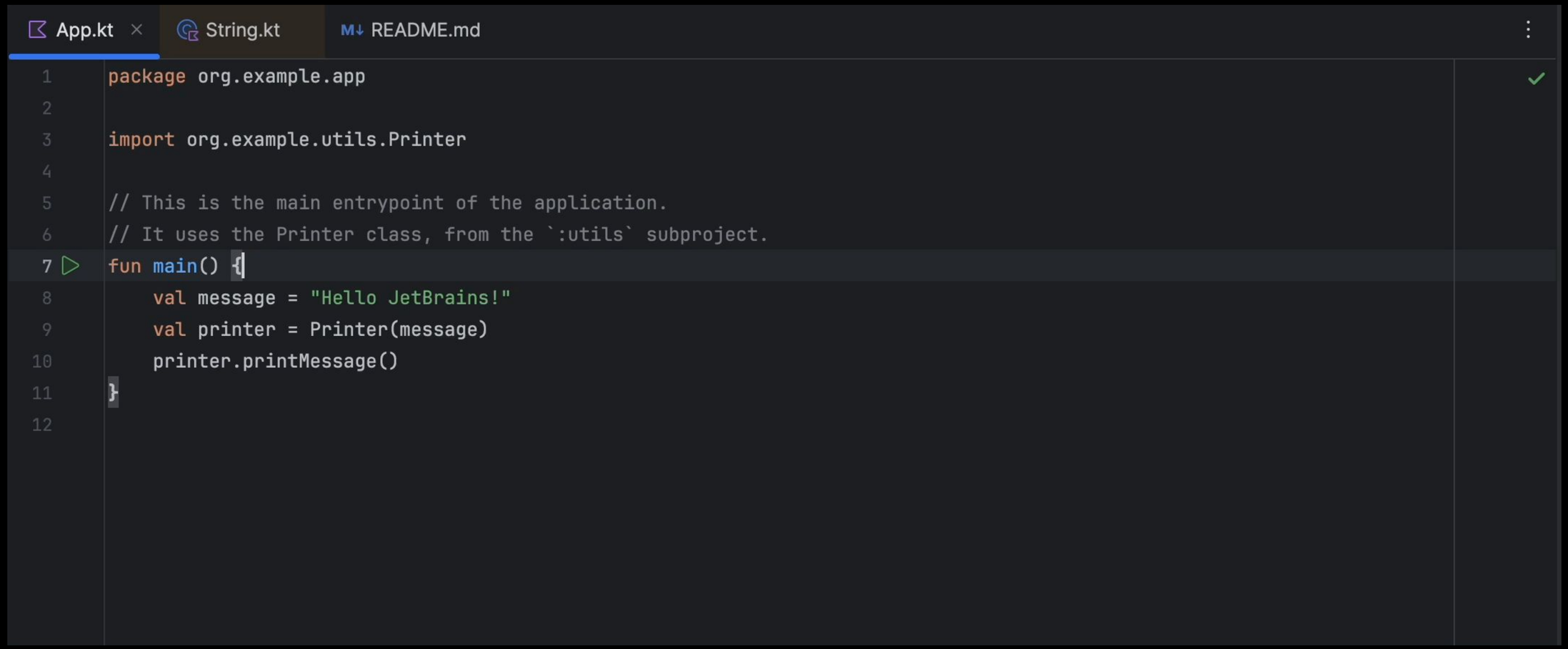
- Gradle / Maven support
- Libraries
- Dokka

IDE Plugin

- Endless number of features
  - Project wizards
  - Navigation
  - Quick fixes
  - Refactorings
  - ...



Project wizard



Refactoring

What is Kotlin and its ecosystem

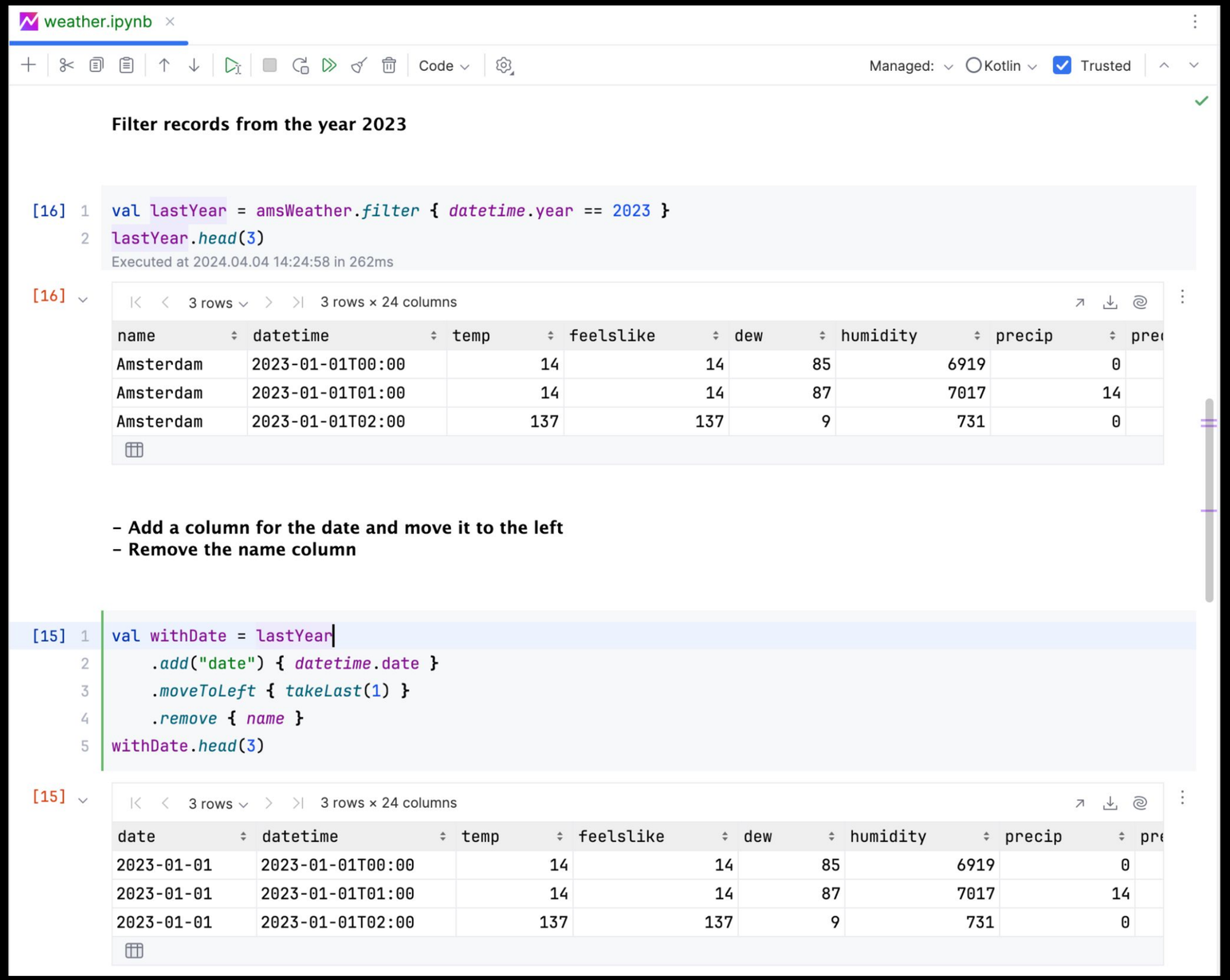
Kotlin as a Platform

What is Kotlin and its ecosystem

Kotlin as a Platform

Kotlin Ecosystem

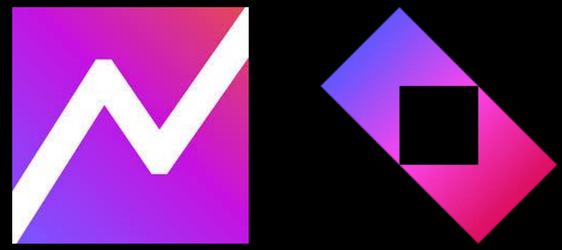
- Kotlin for Data Science



Kotlin Notebooks

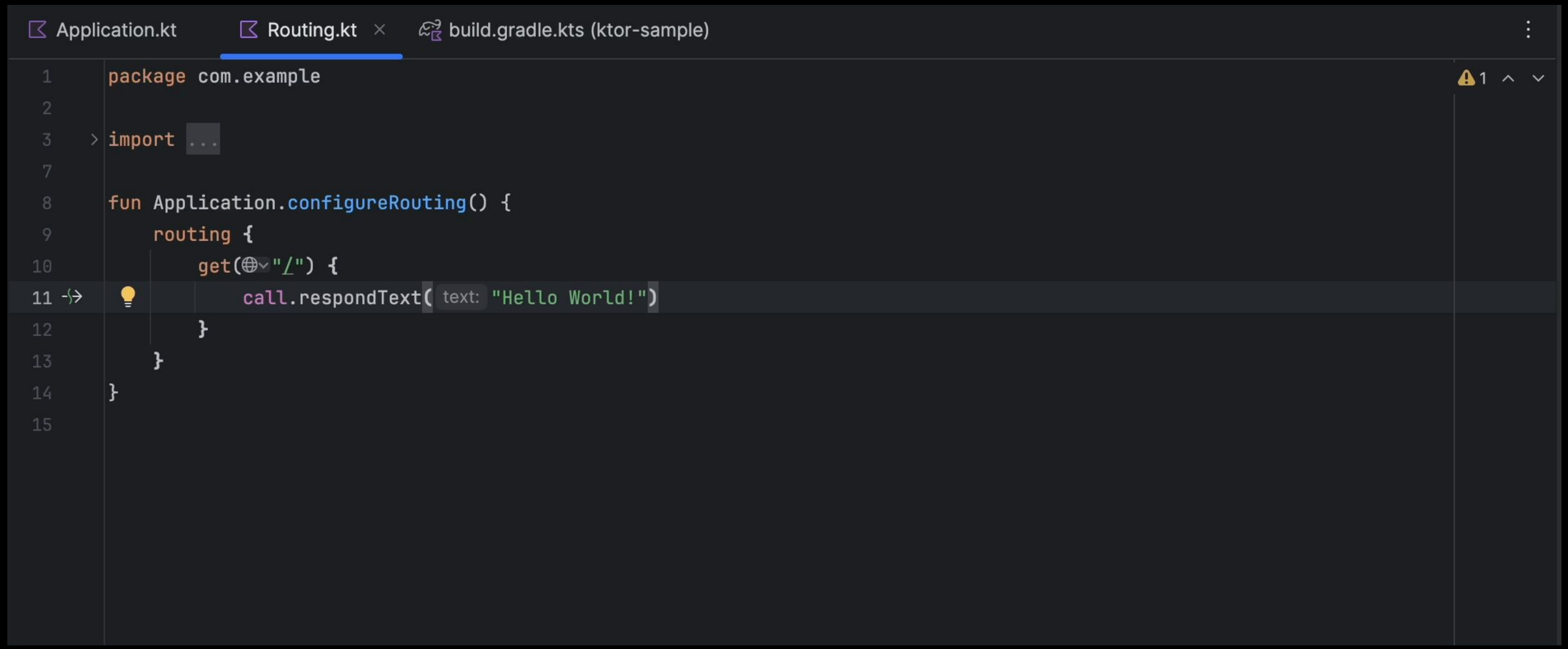
What is Kotlin and its ecosystem

Kotlin as a Platform



Kotlin Ecosystem

- Kotlin for Data Science
- Frameworks (Ktor, etc.)



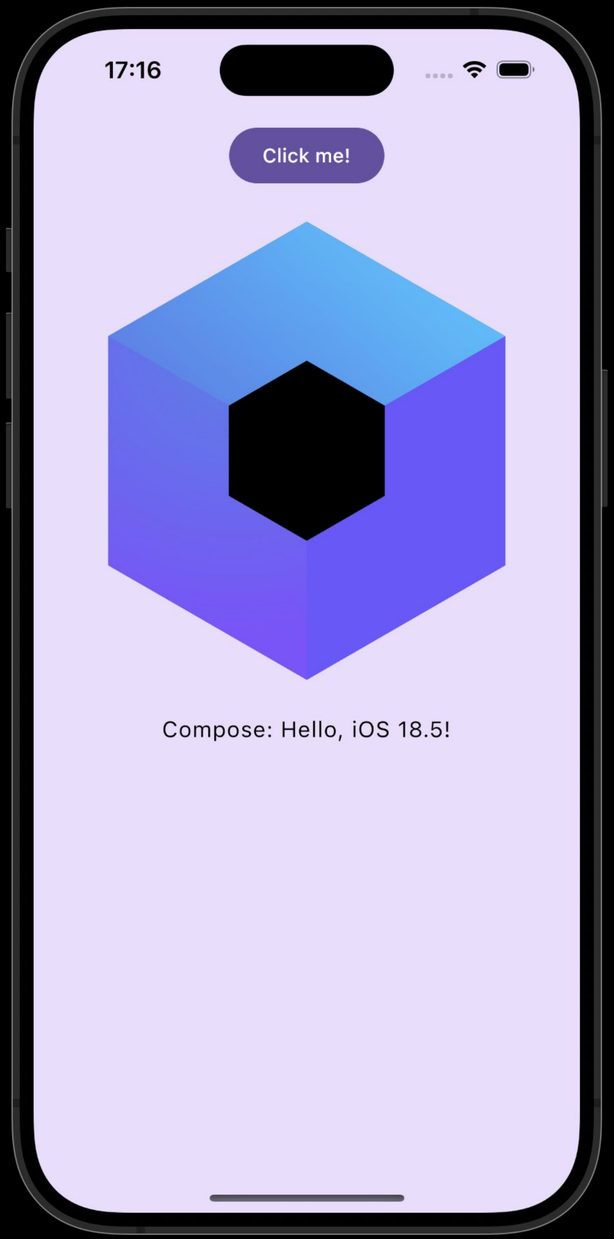
Ktor features in IDE

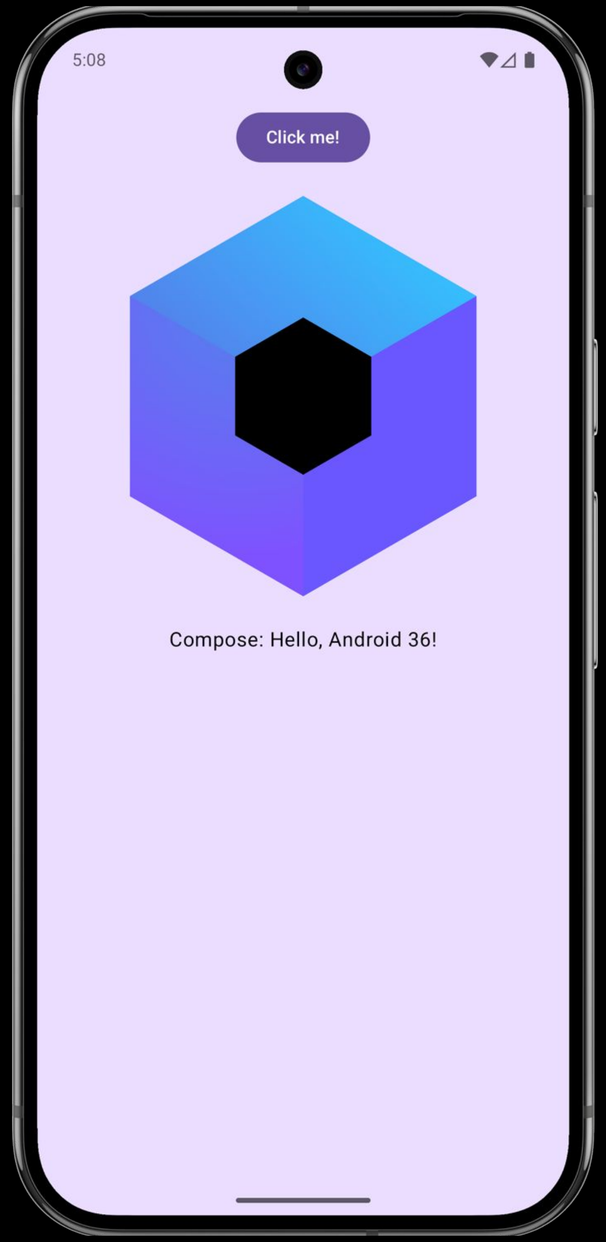
What is Kotlin and its ecosystem

Kotlin as a Platform

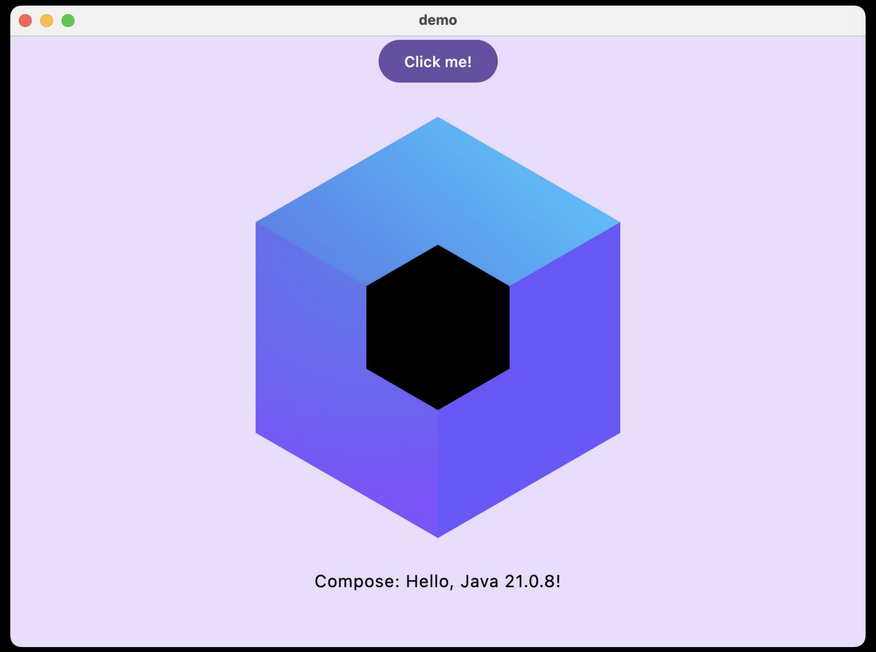
Kotlin Ecosystem

- Kotlin for Data Science
- Frameworks (Ktor, etc.)
- Compose Multiplatform



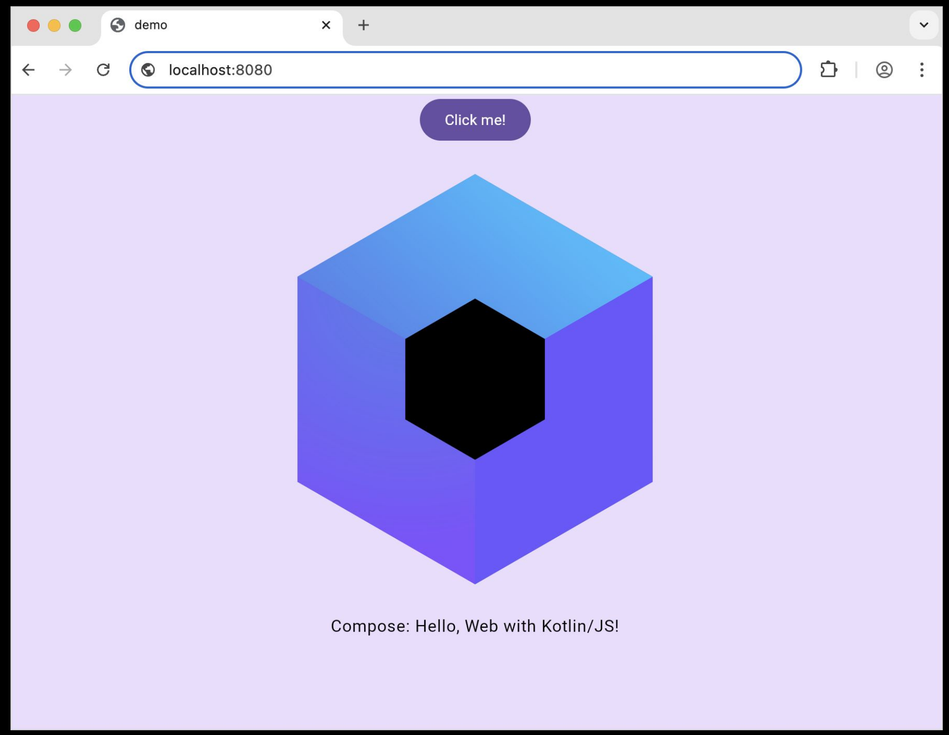


Mobile



Desktop

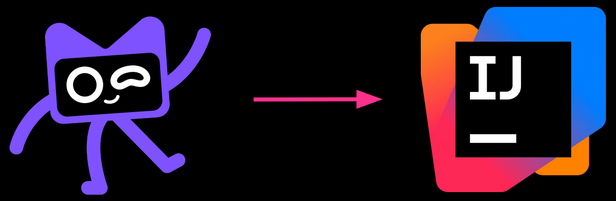
Web



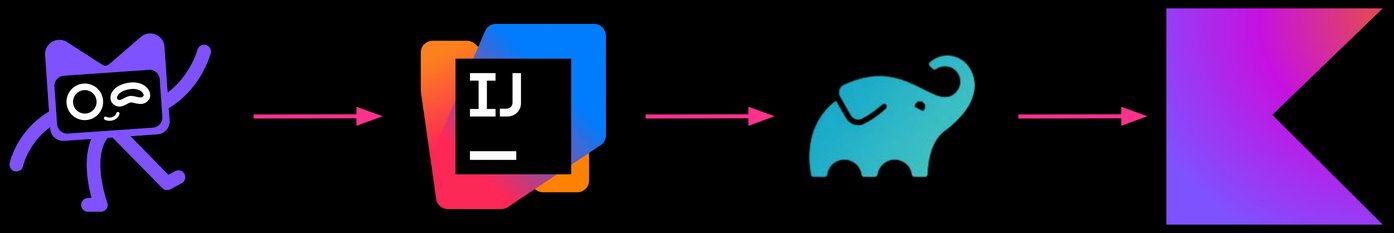
Compose Multiplatform

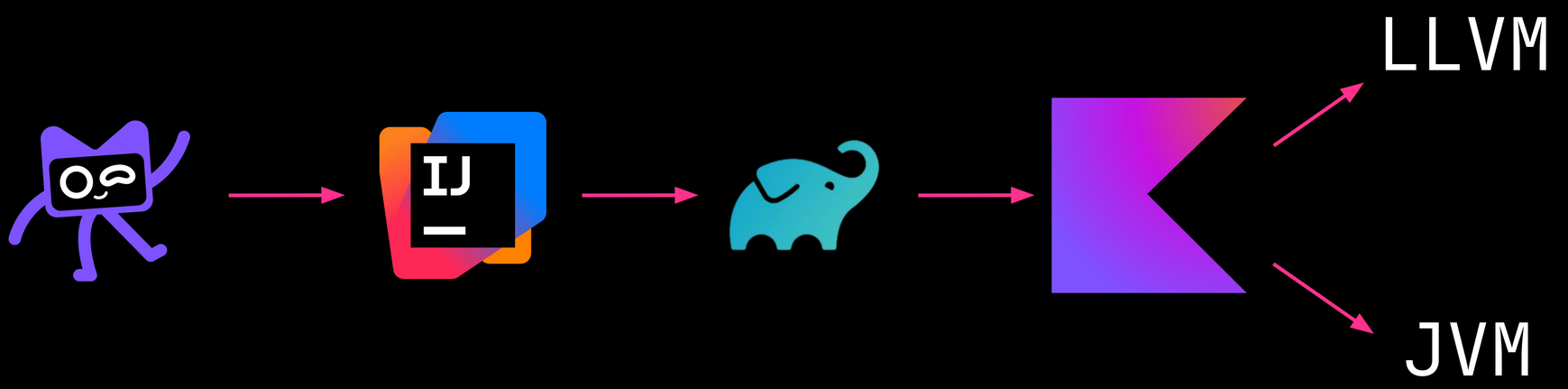

IJ
IJ

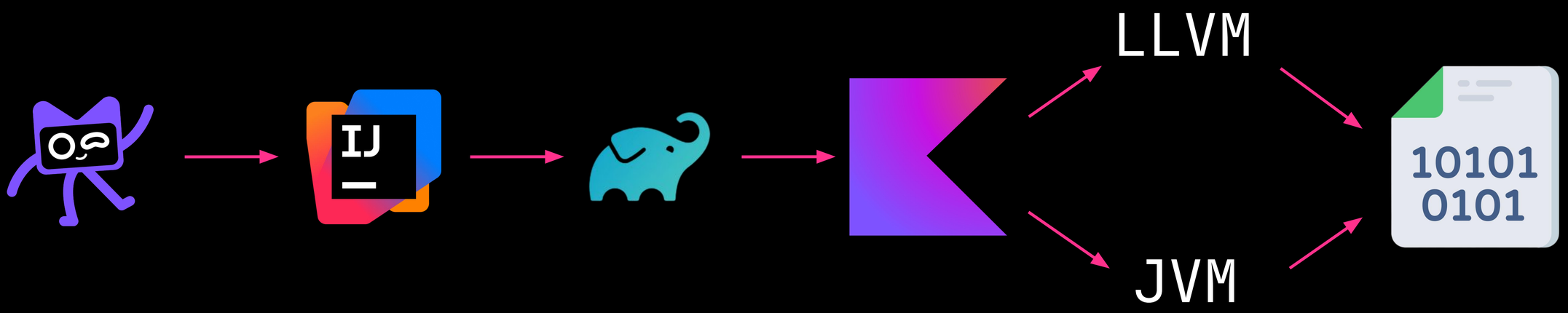
From users to machine code

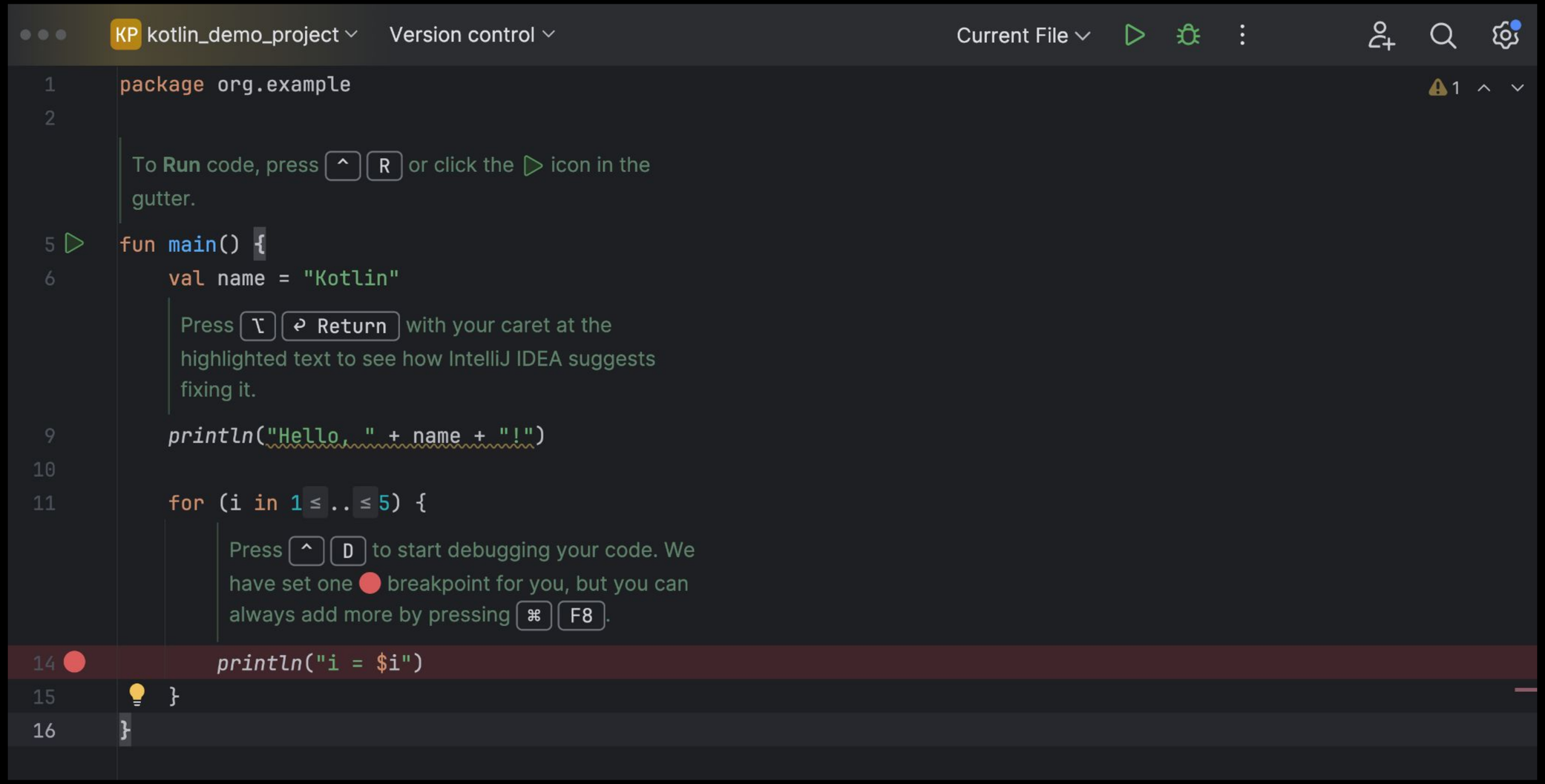



IJ

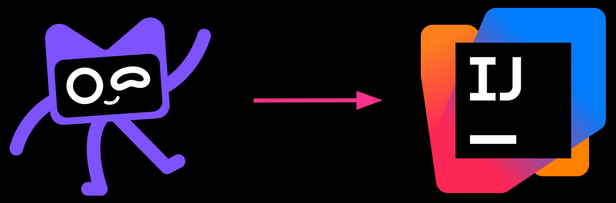

IJ


IJ
LLVM
JVM


IJ
LLVM
JVM
10101
0101



Entry point to the ecosystem



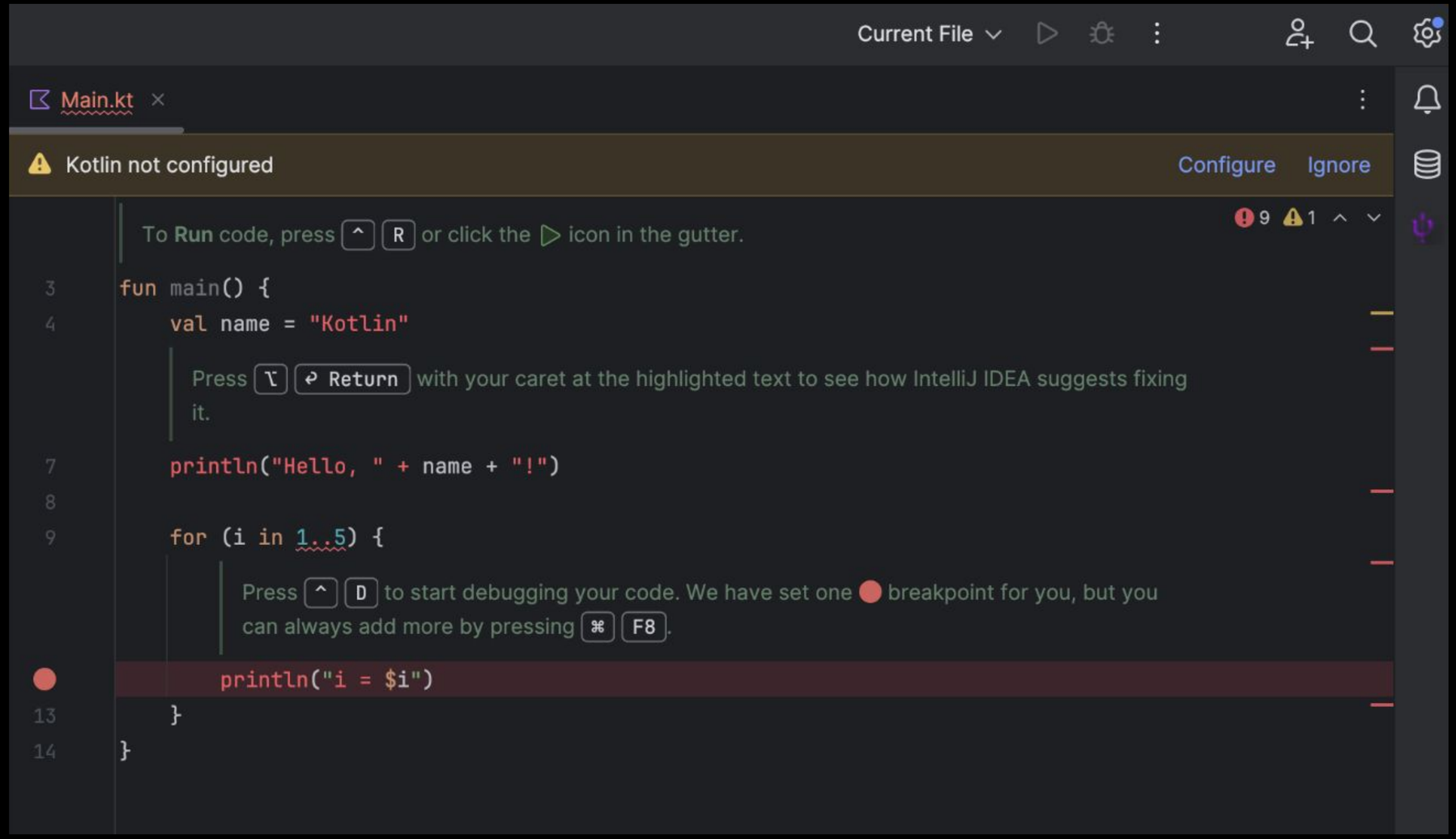
Kotlin IDE Plugin

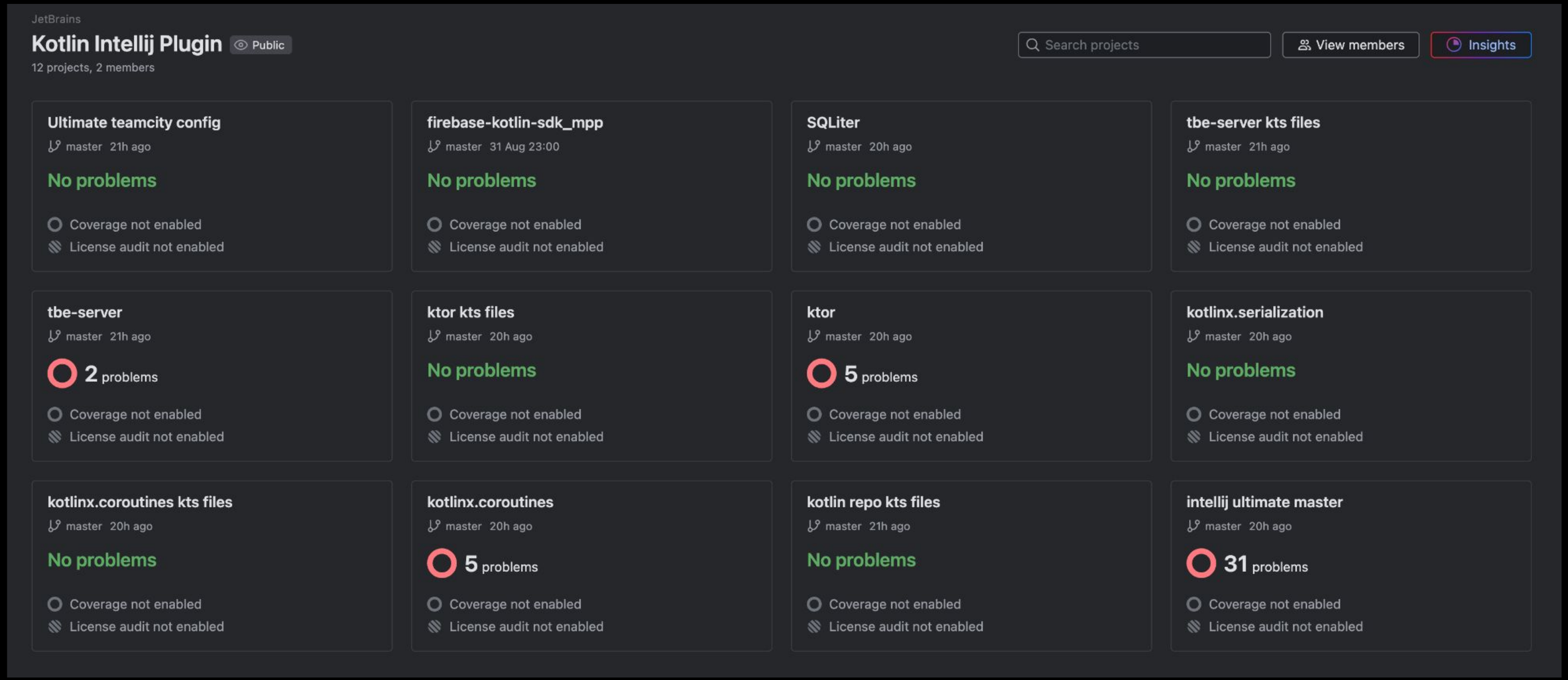
Perception of quality

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code


Current File
Main.kt
Kotlin not configured
Configure
Ignore
To Run code, press ^ R or click the ▷ icon in the gutter.
9 1
fun main() {
val name = "Kotlin"
Press ⌥ ⏎ Return with your caret at the highlighted text to see how IntelliJ IDEA suggests fixing it.
println("Hello, " + name + "!")
for (i in 1..5) {
Press ^ D to start debugging your code. We have set one breakpoint for you, but you can always add more by pressing ⌘ F8.
println("i = $i")
}
}



Red Code tests

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code ✅

  ⇒ Red Code tests

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code ✅

  ⇒ Red Code tests
- Most important Kotlin user scenarios work in a live IDE

Kotlin IDE Plugin

User Workflow scenarios

Scenarios

- Kotlin + Java developer
- Kotlin Android developer
- Kotlin Spring developer
- Kotlin Multiplatform developer
- Kotlin Fullstack (JVM + JS) developer
- and around 20 more

Kotlin IDE Plugin

User Workflow scenarios

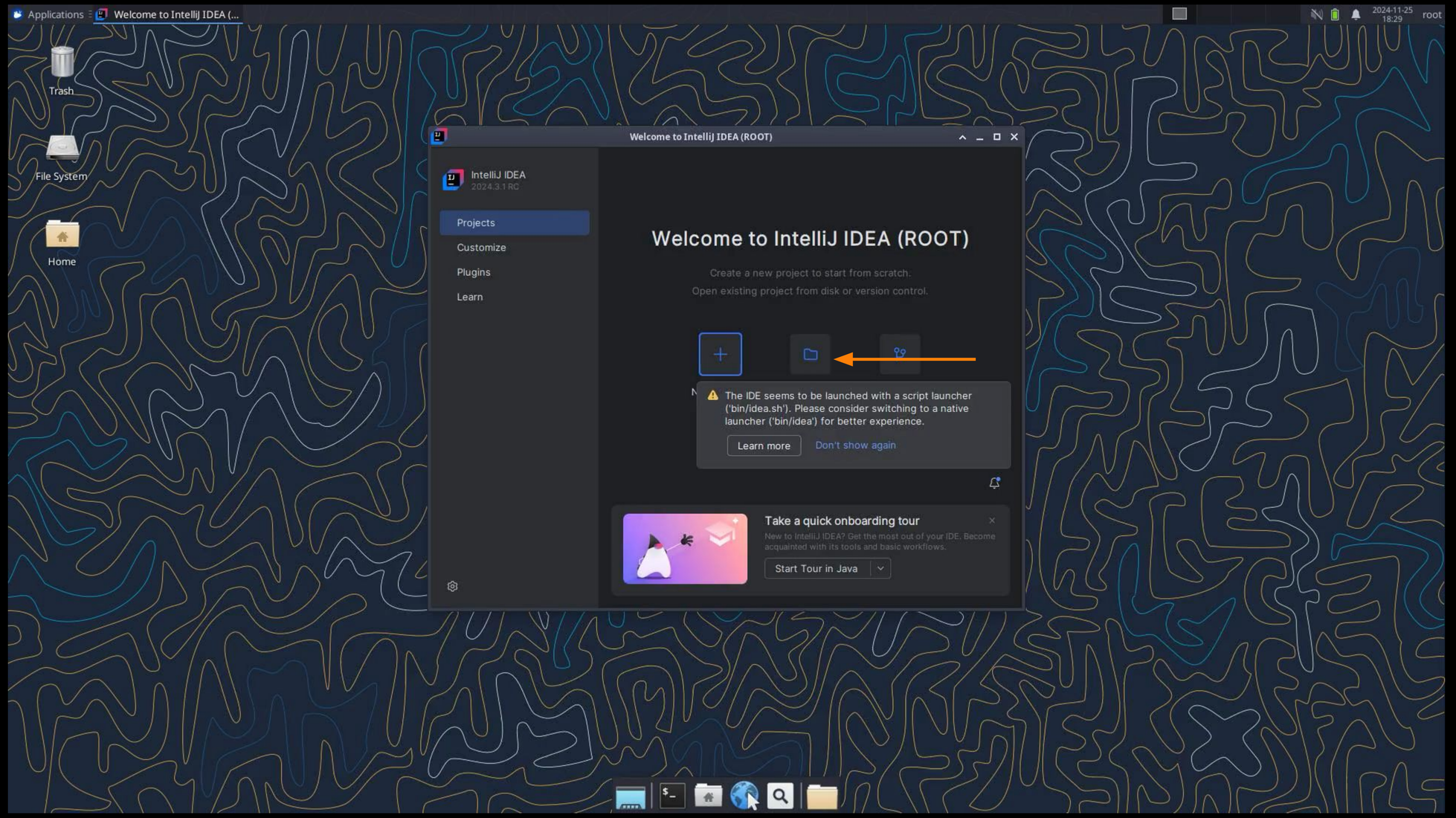
Scenarios

- Kotlin + Java developer
- Kotlin Android developer
- Kotlin Spring developer
- Kotlin Multiplatform developer
- Kotlin Fullstack (JVM + JS) developer
- and around 20 more

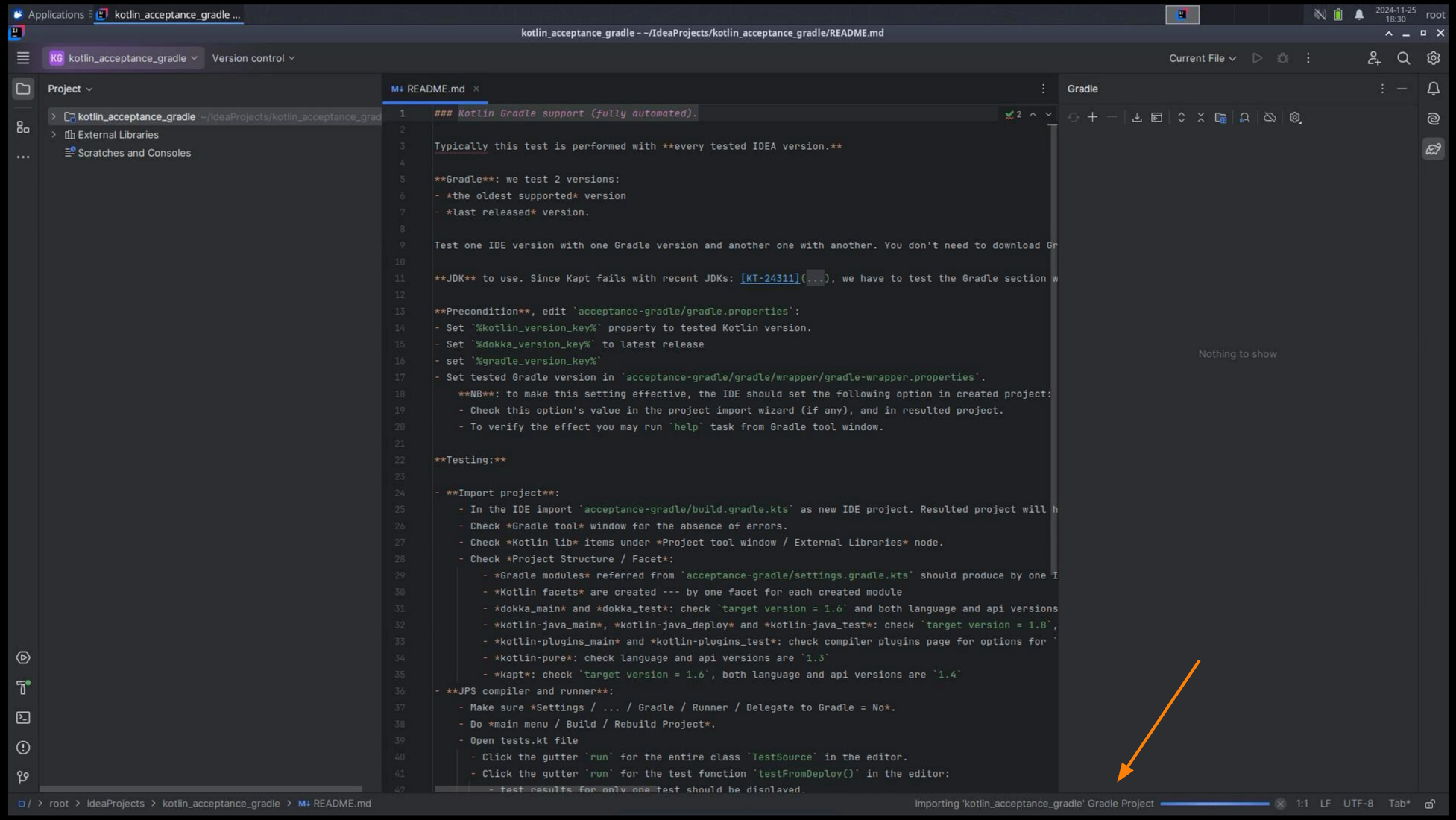
Use Cases

Typical developer session

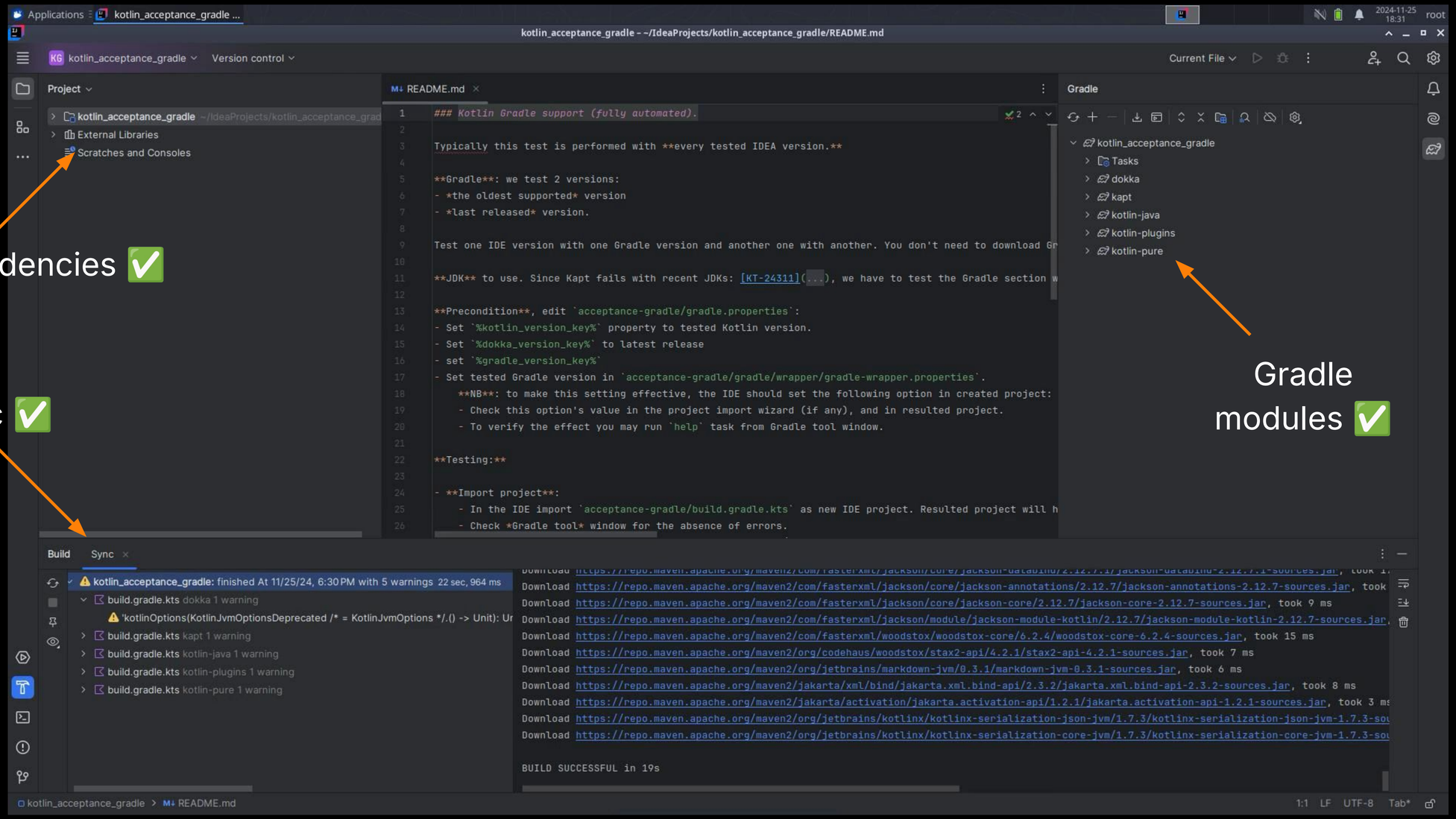
- Open project → assert import
- Open .kt file → assert analysis
- Navigate / refactor → assert K/IDE features
- Run project → assert run configurations
- Debug project → assert debugging features
- Run tests → assert Test Source features
- + unique cases for platforms



Developer opens a project



Let’s wait for the project import



We assert import success

```kotlin
/**
 * https://jetbrains.team/p/iuia/code/ui-tests-data/files/projects/kotlin_acceptance_maven/README.md
 * Kotlin Maven support.
 */
@TestFactory  alexander.khlopotov +5
@Execution(ExecutionMode.SAME_THREAD)
fun kotlinMavenAcceptance() = kotlinScenario("Kotlin Maven Acceptance") {
    startIdea

    kotlinTest("Open project test") {
        openProject(PROJECT_NAME)
    }

    kotlinTest("Check facets") {
        kotlinProject {
            mavenAcceptanceFacets
        }
    }

    kotlinTest("Check dependencies") {
        kotlinProject {
            mavenAcceptanceDependencies
        }
    }
```

How it looks in the framework code

Created by Alexander Khlopotov over 1 year ago Updated by Sergey Moryahin over 1 year ago

K/N libraries don't load and index after project import

kotlin-regression multiplatform KED-qa-autotest-covered

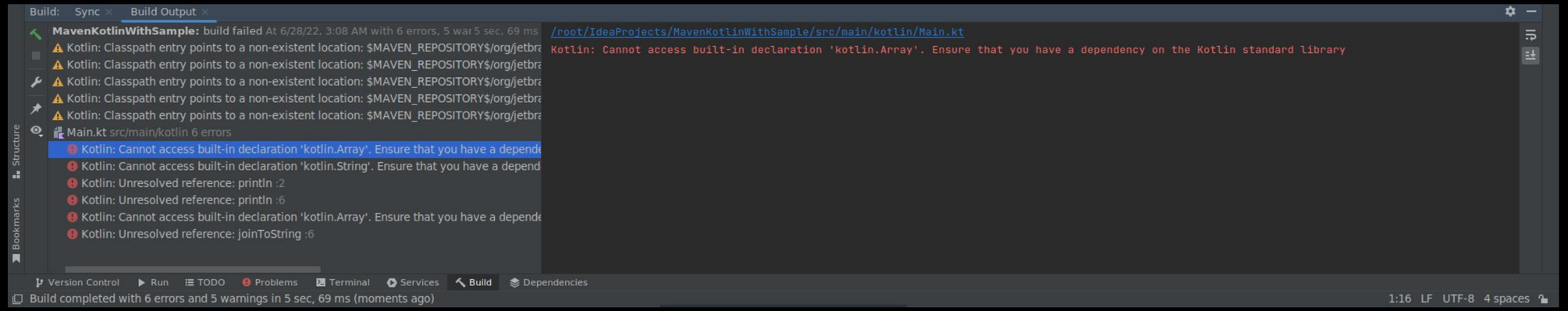
```
@OptIn(ExperimentalForeignApi::class)
fun main() {
    println("Hello from ${SystemParameters().getOS()} target!")
    println("Native library call: ${pthread_self()}")
    val file = platform.posix.fopen("run_results", "a")
    try {
        platform.posix.fputs("mac.main\n", file)
    } finally {
        platform.posix.fclose(file)
    }
}
```

Found Bugs Examples

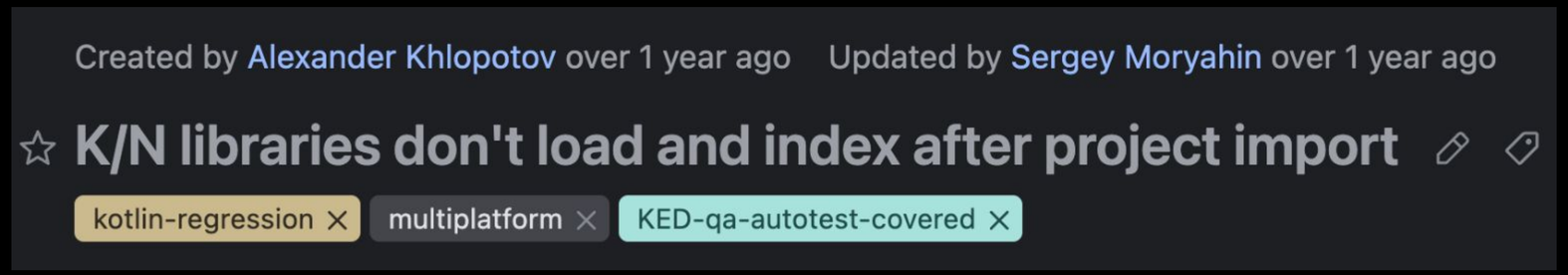
Created by Artur Parpibaev about 3 years ago Updated by Vladimir Naumenko 9 months ago

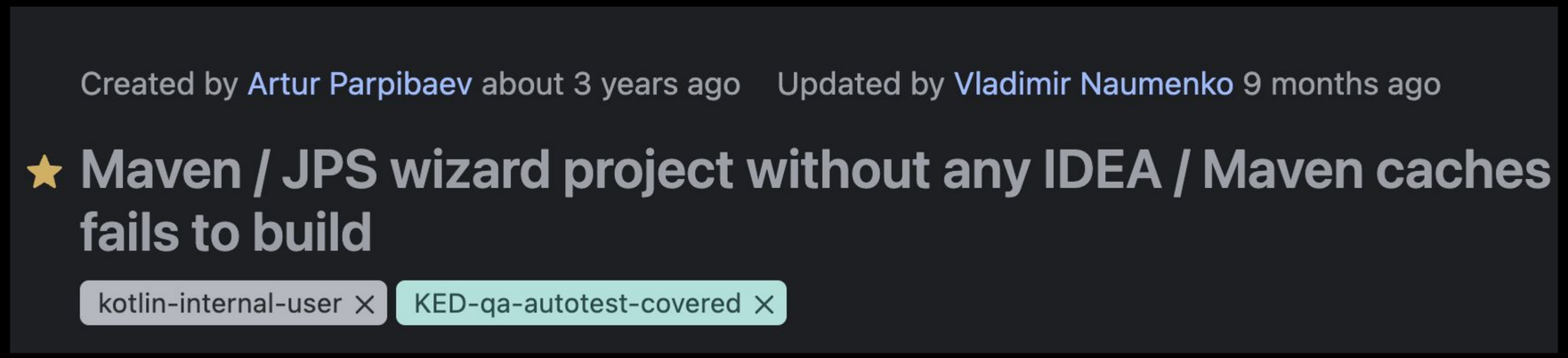
★ Maven / JPS wizard project without any IDEA / Maven caches fails to build

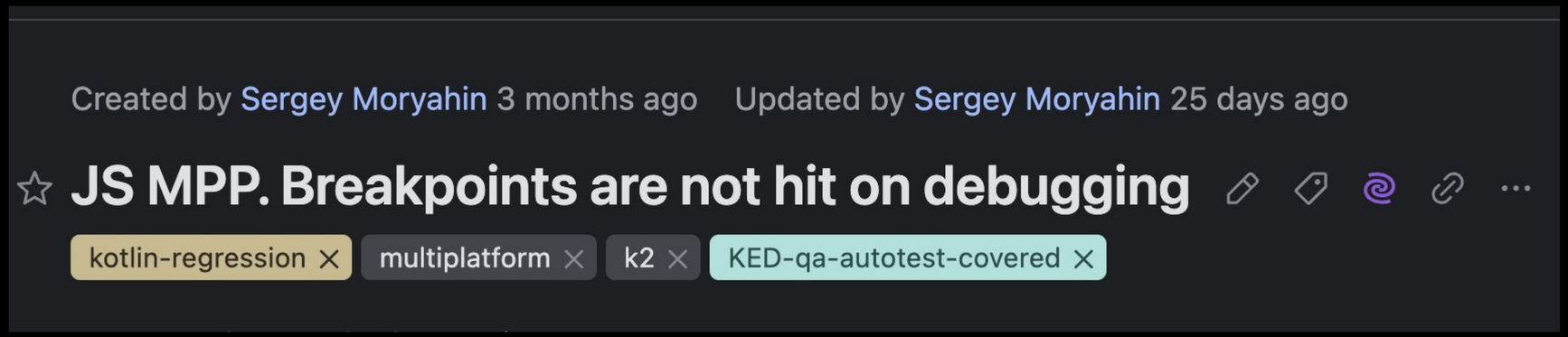
kotlin-internal-user × KED-qa-autotest-covered ×



Found Bugs Examples







Found Bugs Examples

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code ✅

  ⇒ Red Code tests
- Most important Kotlin user scenarios work in a live IDE ✅

  ⇒ User Workflow tests

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code ✅

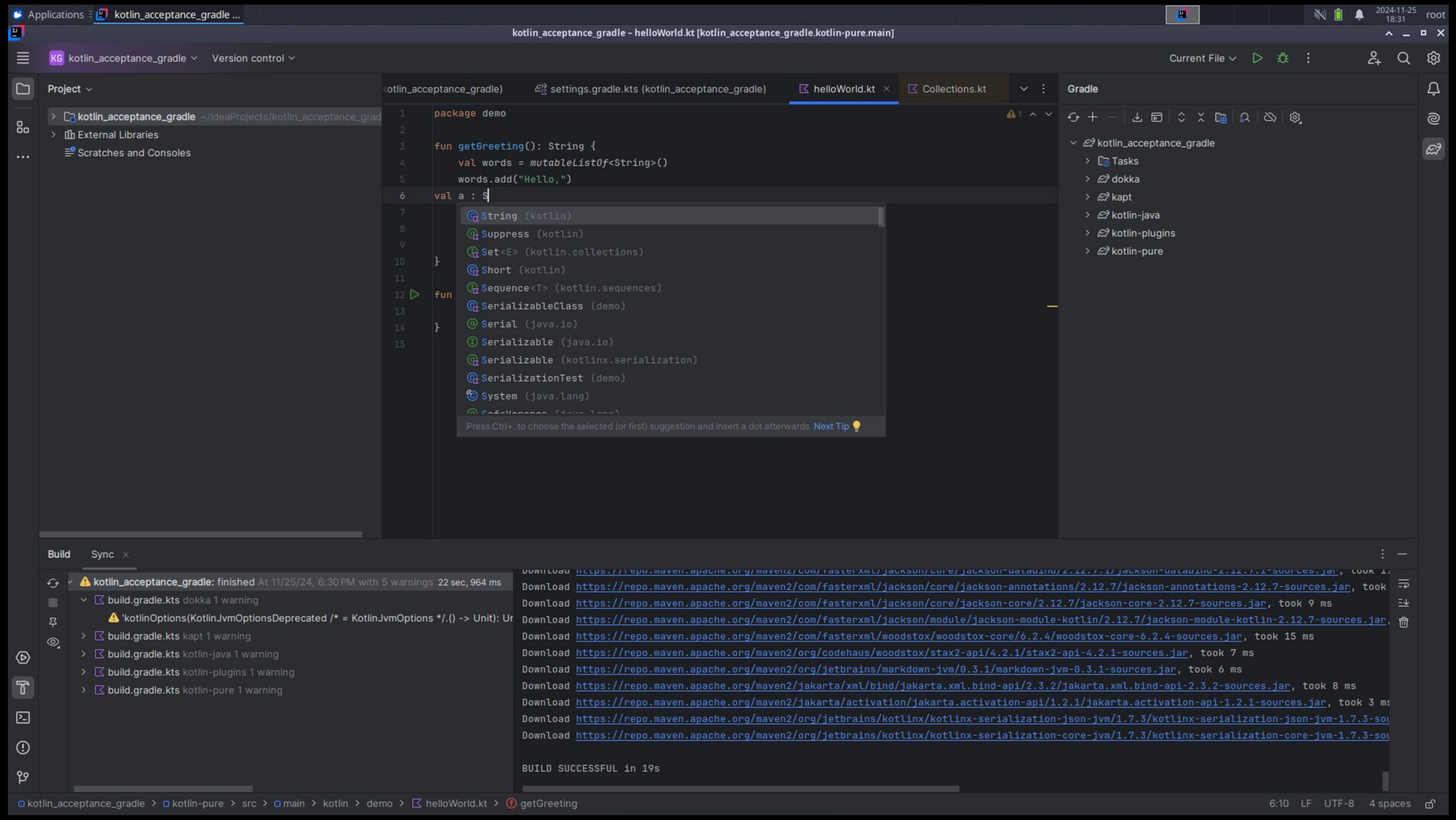
  ⇒ Red Code tests
- Most important Kotlin user scenarios work in a live IDE ✅

  ⇒ User Workflow tests
- Features with UI work in a live IDE

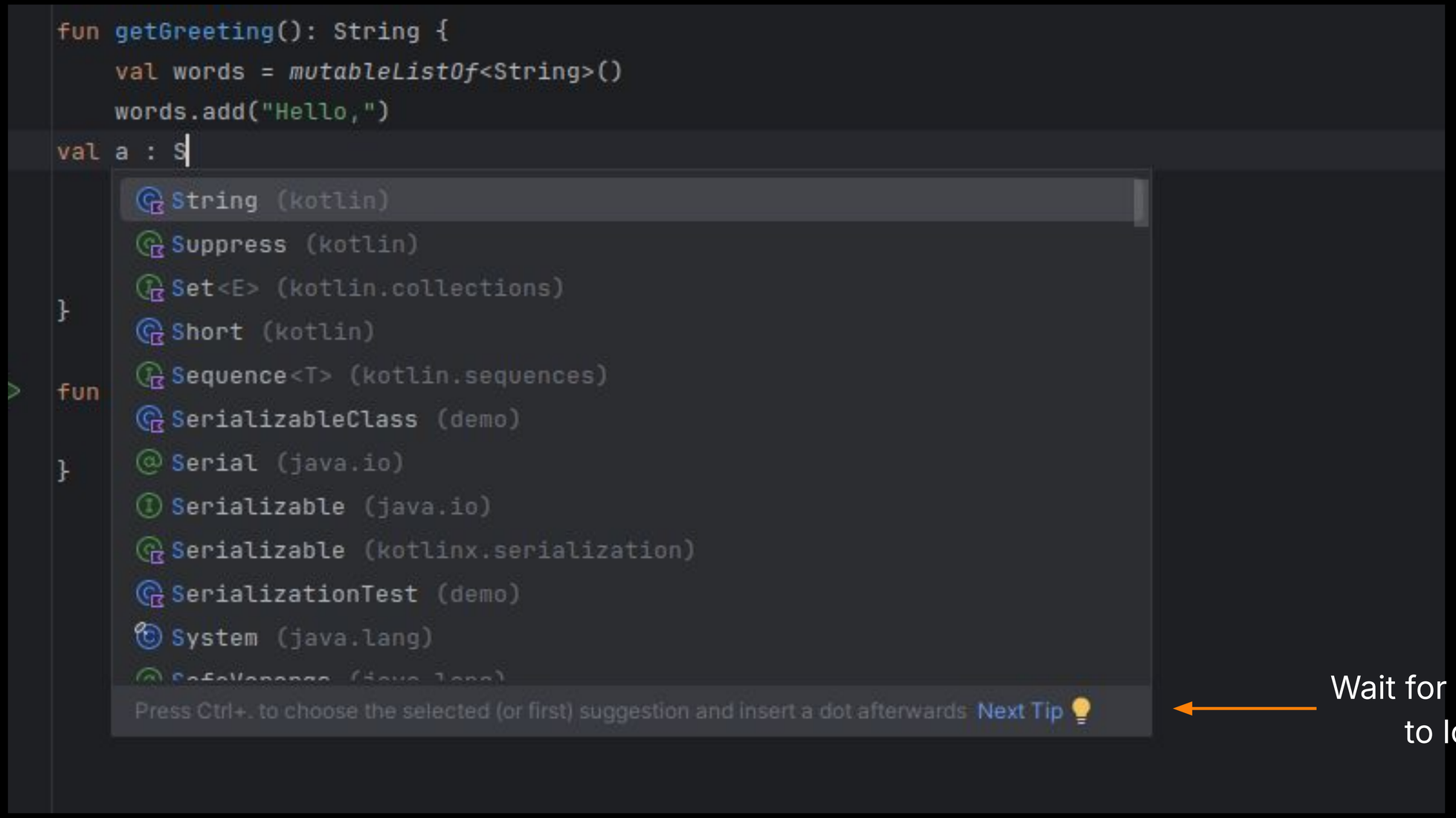
Kotlin IDE Plugin

UI testing of the UI-rich features

- More than 1000 UI-tests
  - Navigation
  - Code Completion
  - Quick Documentation
  - Find Usages
  - Refactorings
  - Code Analysis
  - Project Wizards
  - Convert Java to Kotlin
  - Debuggers (JVM, JS, Wasm, Native, Coroutines)
  - Kotlin Bytecode viewer
  - Add Kotlin module to project
  - ...



Code Completion check



Assert completion

Assert completion options & success

Kotlin IDE Plugin

Perception of quality

- Project opens without any red code ✅
  ⇒ Red Code tests
- Most important Kotlin user scenarios work in a live IDE ✅
  ⇒ User Workflow scenarios
- Features with UI work in a live IDE ✅
  ⇒ Feature-based UI-tests


IJ

Kotlin Build Toolchain

How can we assure a build system works well?

... probably by testing Kotlin Gradle Plugin? Or the whole build tooling for Kotlin?

Kotlin Build Toolchain

How can we assure a build system works well?

... probably by testing Kotlin Gradle Plugin? Or the whole build tooling for Kotlin?

Kotlin Build Tools QA Team

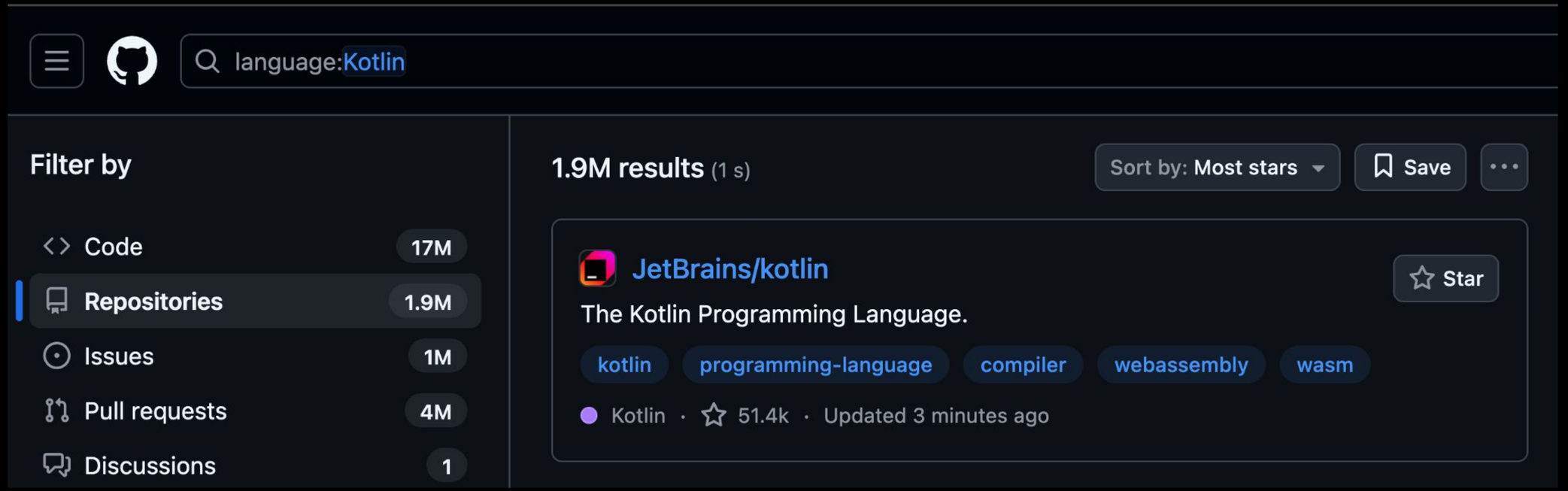
Kotlin Build Toolchain

And how can we assure we build any code smoothly...

... probably by building the entire world*?

*Kotlin projects in the most popular ways of using

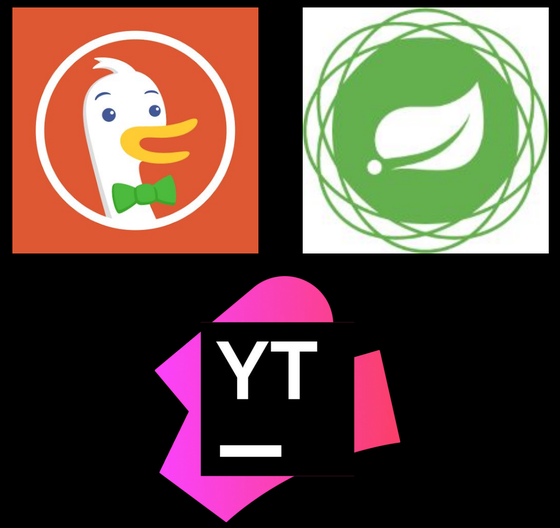
User Projects

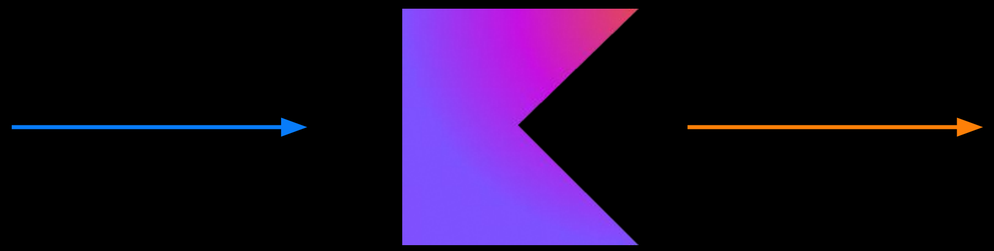


Does not look realistic

User Projects

The concept





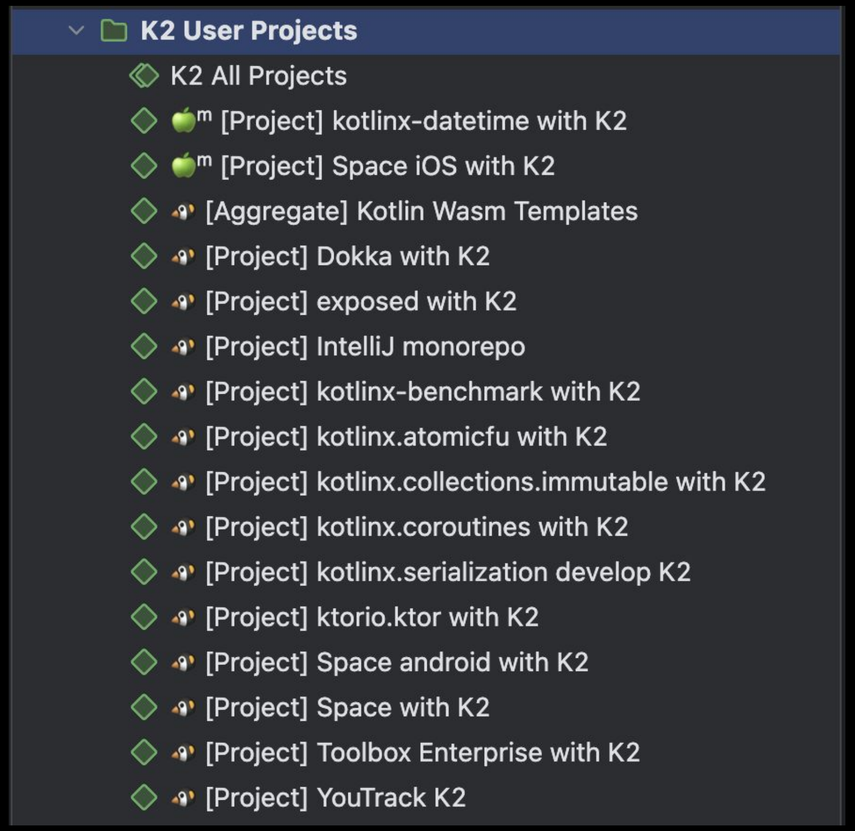
Take popular apps

Build with the latest dev Kotlin

Catch all the bugs

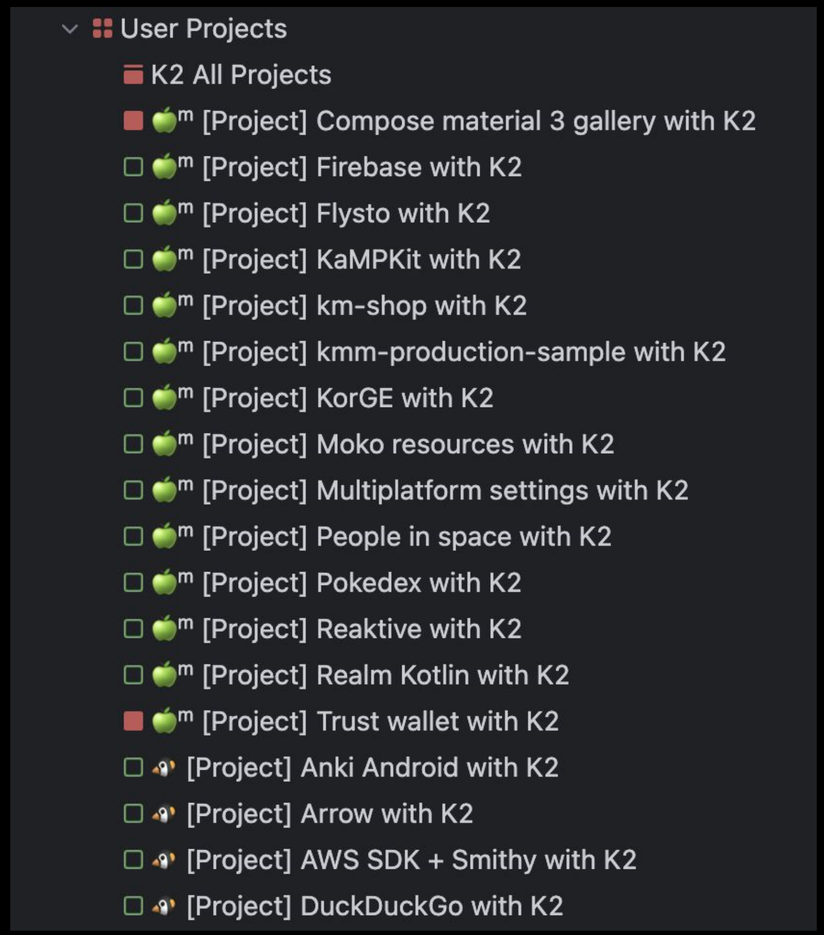
User Projects

- Important internal JetBrains Kotlin projects
  - Intellij IDEA
  - Kotlin
  - Kotlin Libraries
  - Teamcity
  - YouTrack
  - etc.



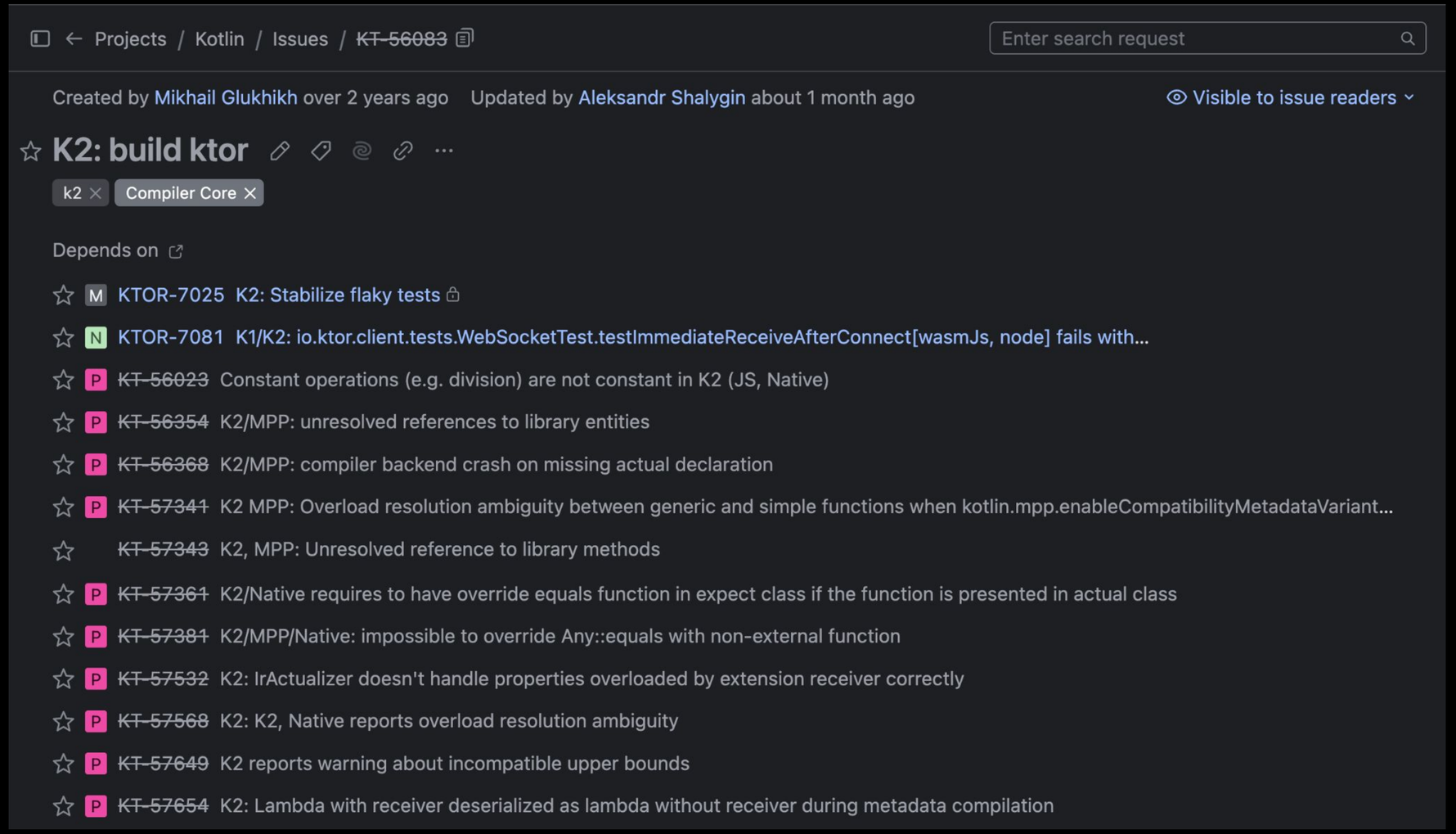
User Projects

- Important internal JetBrains Kotlin projects
- Open-source largest and diverse Kotlin projects
  - JVM
  - JS
  - Mobile Multiplatform
    - iOS
    - Android
  - Native Desktop



User Projects

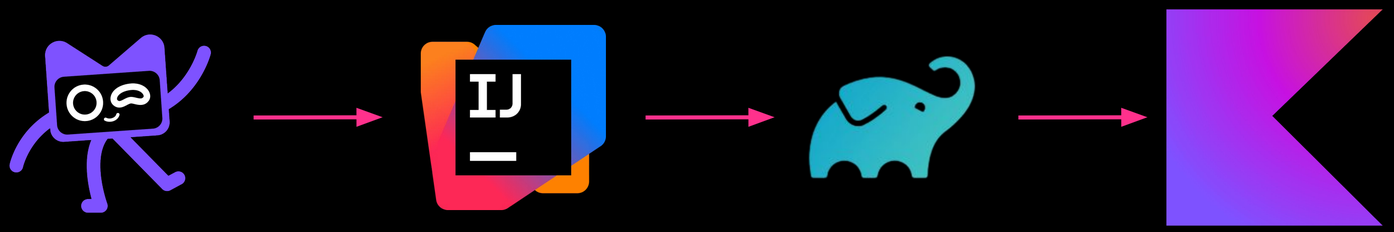
- Important internal JetBrains Kotlin projects
- Open-source largest and diverse Kotlin projects



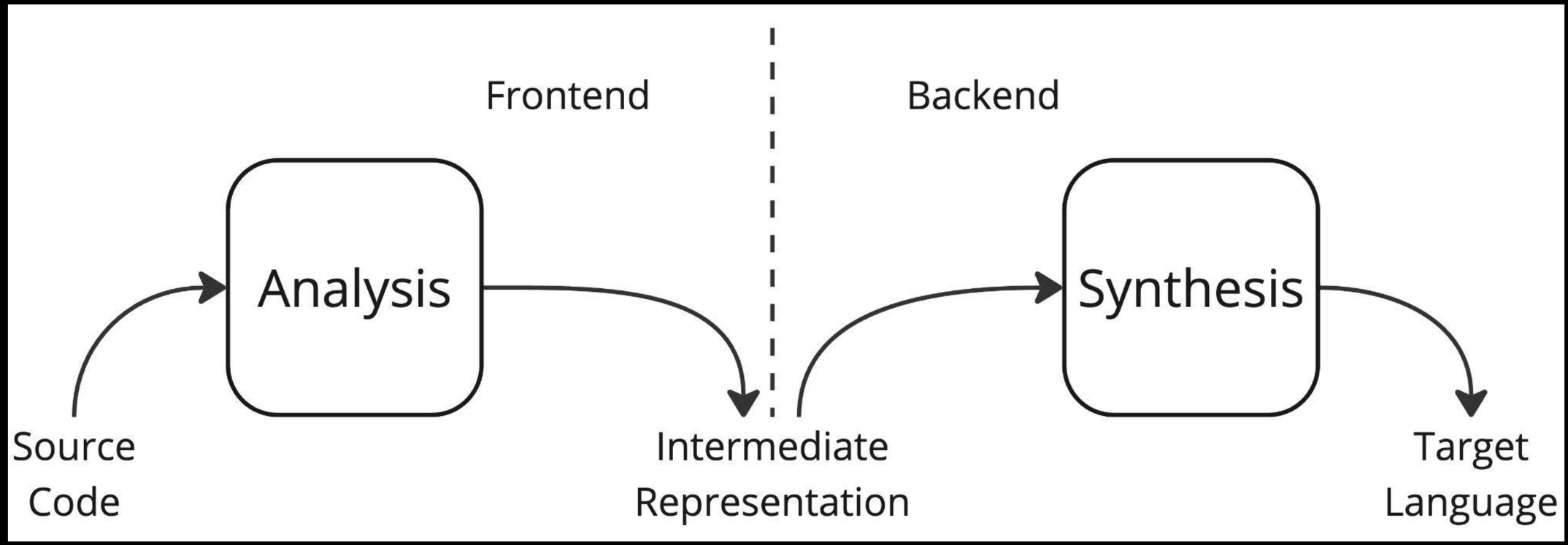
Over 40 major+ regressions was found

In 2 years

On a single project out of 50


IJ

Compiler


Frontend
Backend
Analysis
Synthesis
Source
Code
Intermediate
Representation
Target
Language

Compiler frontend

- Analysis (i.e. syntax highlighting)
- Language features
- Intermediate Representation (IR) generation

Created by **Aleksandra Arsenteva** over 1 year ago Updated by **Aleksandra Arsenteva** 11 months ago Visible to issue readers

Destructuring declaration shouldn't be possible in declaration in when

The following code is green and runs successfully but does nothing:

```kotlin
data class Foo(val name: String)

fun main() {
    val foo = Foo("John")
    when (val (x) = foo) {
        is String -> println("1")
        is Foo -> println("2")
        //else -> println(x)        //Unresolved reference 'x'.
    }
}
```

Kotlin detected

According to the spec it should be forbidden:

> The scope of this property is limited to the `when` expression, including both conditions and control structure bodies of the expression. As its form is limited to a simple "assignment-like" declaration with an initializer, this property does not allow getters, setters, delegation or destructuring.

Just a simple code snippet with a 🪲

Created by **Aleksandra Arsenteva** 8 months ago Updated by **Kirill Rakhman** 8 months ago Visible to issue readers

Context parameters: "IllegalStateException: Cannot find variable a: R|kotlin/String| in local storage " when context from another local function is called

```kotlin
fun outerFun() {
    context(a: String)
    fun local1() { }

    fun local2() {
        a
    }
}
```

Kotlin detected

Result:

```
e: org.jetbrains.kotlin.util.FileAnalysisException: While analysing /Users/Aleksandra.Arsenteva/repositories/qa-materials-spac
        at org.jetbrains.kotlin.util.AnalysisExceptionsKt.wrapIntoFileAnalysisExceptionIfNeeded(AnalysisExceptions.kt:57)
        at org.jetbrains.kotlin.fir.FirCliExceptionHandler.handleExceptionOnFileAnalysis(Utils.kt:251)
        at org.jetbrains.kotlin.fir.backend.Fir2IrConverter.runSourcesConversion(Fir2IrConverter.kt:712)
        at org.jetbrains.kotlin.fir.backend.Fir2IrConverter.access$runSourcesConversion(Fir2IrConverter.kt:71)
        at org.jetbrains.kotlin.fir.backend.Fir2IrConverter$Companion.generateIrModuleFragment(Fir2IrConverter.kt:702)
        at org.jetbrains.kotlin.fir.pipeline.Fir2IrPipeline.runFir2IrConversion(convertToIr.kt:159)
        at org.jetbrains.kotlin.fir.pipeline.Fir2IrPipeline.convertToIrAndActualize(convertToIr.kt:126)
        at org.jetbrains.kotlin.fir.pipeline.ConvertToIrKt.convertToIrAndActualize(convertToIr.kt:97)
        at org.jetbrains.kotlin.fir.pipeline.ConvertToIrKt.convertToIrAndActualize$default(convertToIr.kt:72)
        at org.jetbrains.kotlin.cli.jvm.compiler.legacy.pipeline.JvmCompilerPipelineKt.convertToIrAndActualizeForJvm(jvmCompil
```

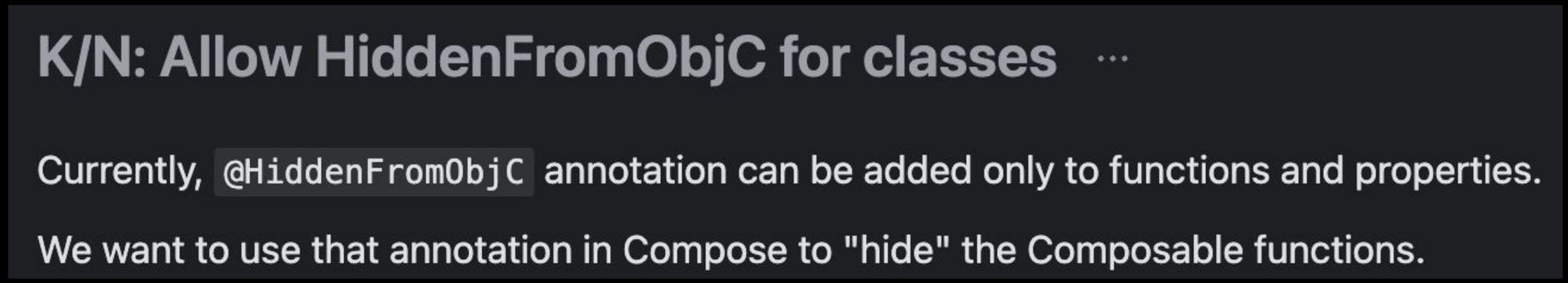
Just a simple code snippet and a compiler crash 💥

Compiler backend

- JVM
  - Java interop
- Native
  - C, ObjC interops, Swift Export
  - Dealing with targets
  - Pict
    - It helps with the problem of unclear coverage while having a lot of flags — we trust pairwise testing
- WASM and JS
  - JS/TS interop

Compiler backend

https://youtrack.jetbrains.com/issue/KT-56464/K-N-Allow-HiddenFromObjC-for-classes



A task 💪🏽

Compiler backend

```
1 @file:OptIn(kotlin.experimental.ExperimentalObjCRefinement::class)
2
3 @HiddenFromObjC
4 class Hidden
5
6 fun Hidden.foo() = 5
```

Bash

Still a task 💪

Compiler backend

```
@file:OptIn(kotlin.experimental.ExperimentalObjCRefinement::class)

@HiddenFromObjC
class Hidden

fun Hidden.foo() = 5
```

Bash

Compilation:

```
➜  hiddenfromobjc_extension_bug /Users/Alexander.Zakharenko/Downloads/kotlin-native-prebuilt-macos-aarch64-1.9.0-Beta-162/bin/
➜  hiddenfromobjc_extension_bug /Users/Alexander.Zakharenko/Downloads/kotlin-native-prebuilt-macos-aarch64-1.9.0-Beta-162/bin/
error: compilation failed: Shouldn't be exposed: deserialized class Hidden
```

A bug 😍

Compiler backend

Autotests

```
import kotlin.test.*

@Test
fun addition() {
    assertEquals(42, 40 + 2)
}

@Test
fun multiplication () {
    assertEquals(42, 21 * 2)
}
```

Compiler backend

Autotests

```
import kotlin.test.*

@Test
fun addition() {
    assertEquals(42, 40 + 2)
}

@Test
fun multiplication () {
    assertEquals(42, 21 * 2)
}
```

```
Gradle tasks and arguments: :nativeCompilerTest --init-script
/opt/buildAgent/plugins/gradle-runner/scripts/init_since_8.gradle
--init-script /opt/buildAgent/temp/agentTmp/build-scan-init.gradle
-Pteamcity=true --no-watch-fs -Pkotlin.build.scan.url=XXX
-Dscan.tag.kotlin-dev -Pkotlin.build.testRetry.maxRetries=0
-Pkotlin.build.isObsoleteJdkOverrideEnabled=true --parallel
--continue -Pkotlin.native.enabled=true
-Pkotlin.internal.native.test.nativeHome=/opt/buildAgent/work/17e640
964edd2053/test_dist -Pkotlin.incremental=false
-Pkotlin.internal.native.test.mode=TWO_STAGE_MULTI_MODULE
-Pkotlin.internal.native.test.optimizationMode=DEBUG
-Pkotlin.internal.native.test.cacheMode=STATIC_EVERYWHERE
-Pkotlin.internal.native.test.gcType=CMS
-Pkotlin.internal.native.test.gcScheduler=ADAPTIVE
-Pkotlin.internal.native.test.alloc=CUSTOM
-Pkotlin.internal.native.test.target=ios_simulator_arm64
-Pbuild.number=2.1.0-dev-8366
-Pkotlin.native.tests.tags=frontend-fir -Dorg.gradle.daemon=false -s
-b build.gradle.kts
```

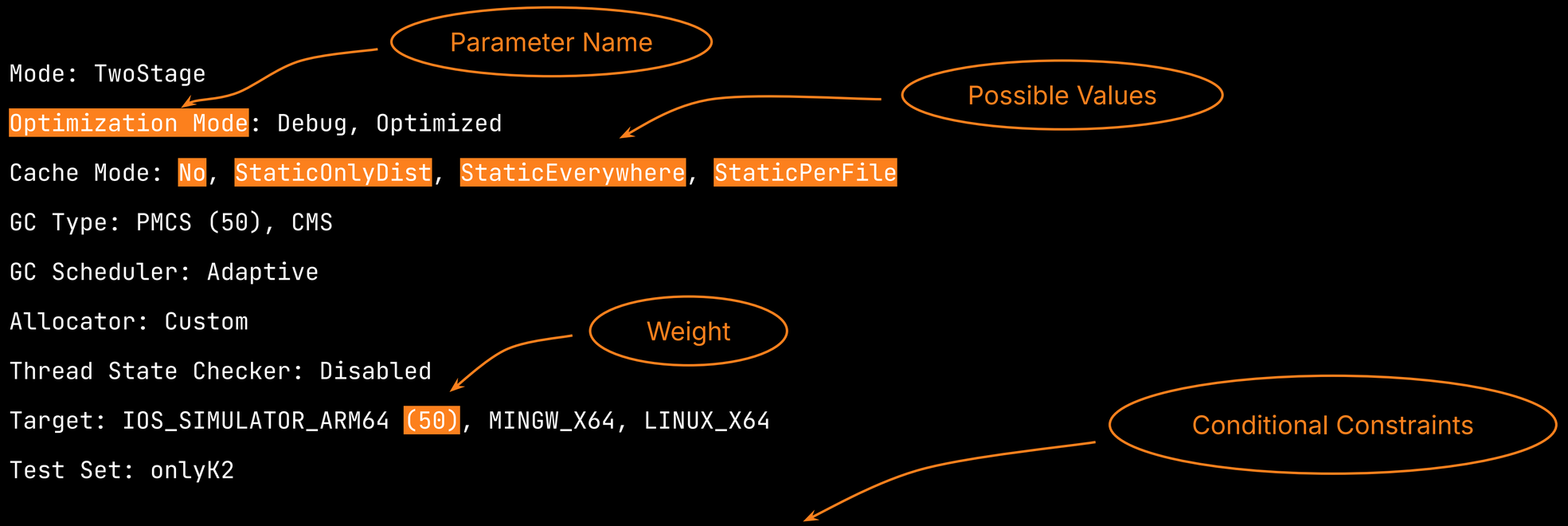
Compiler backend

Autotests

- Different aggregate test configurations
- Tagged tests
- Optimized test configurations
  - Pairwise testing as we have a lot of compiler flags — for each pair of input parameters, tests all possible discrete combinations of those parameters
    - Coverage is clarified at the level of configurations
    - Less tests
  - Some “must have” configurations on top

Compiler backend

Autotests — PICT



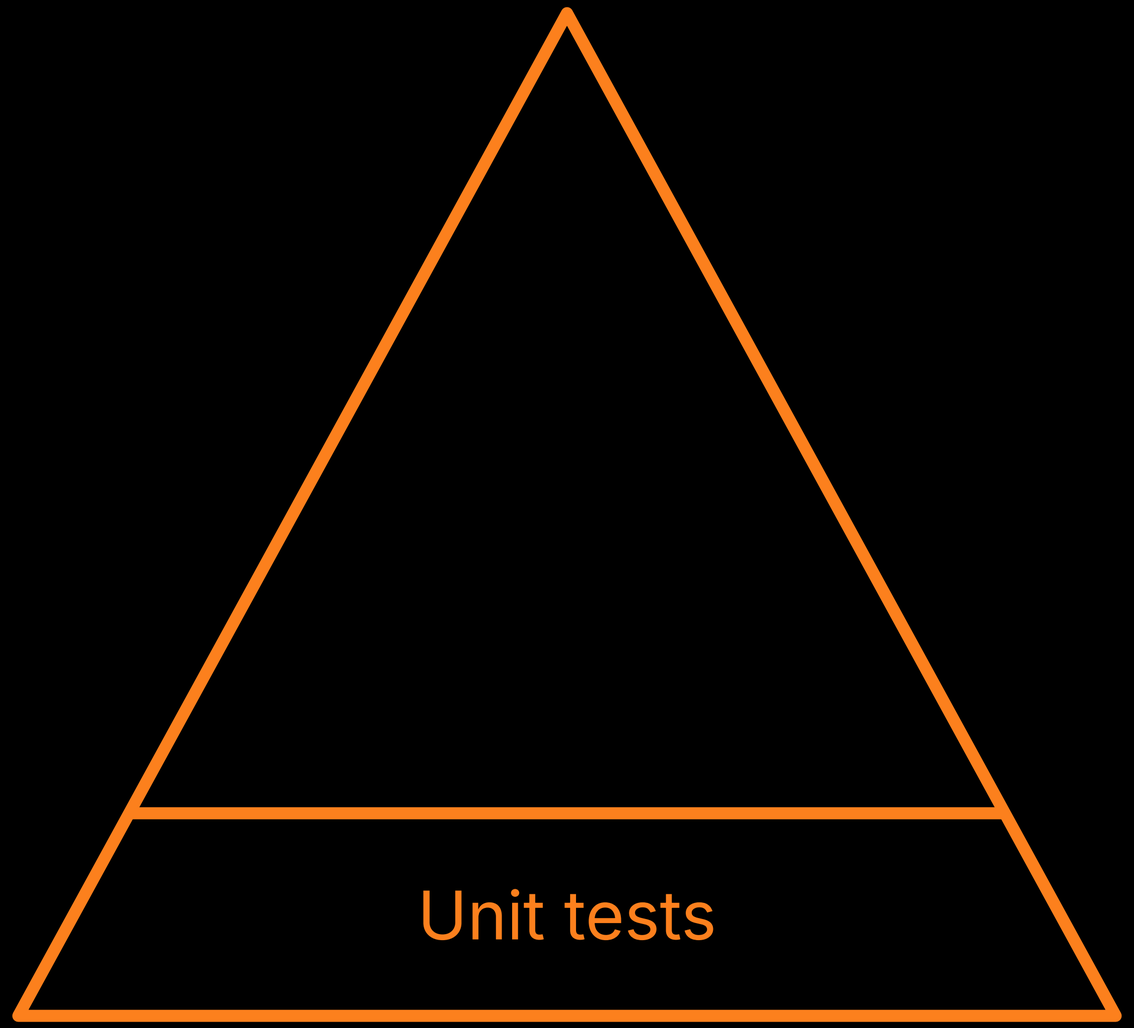
Compiler backend

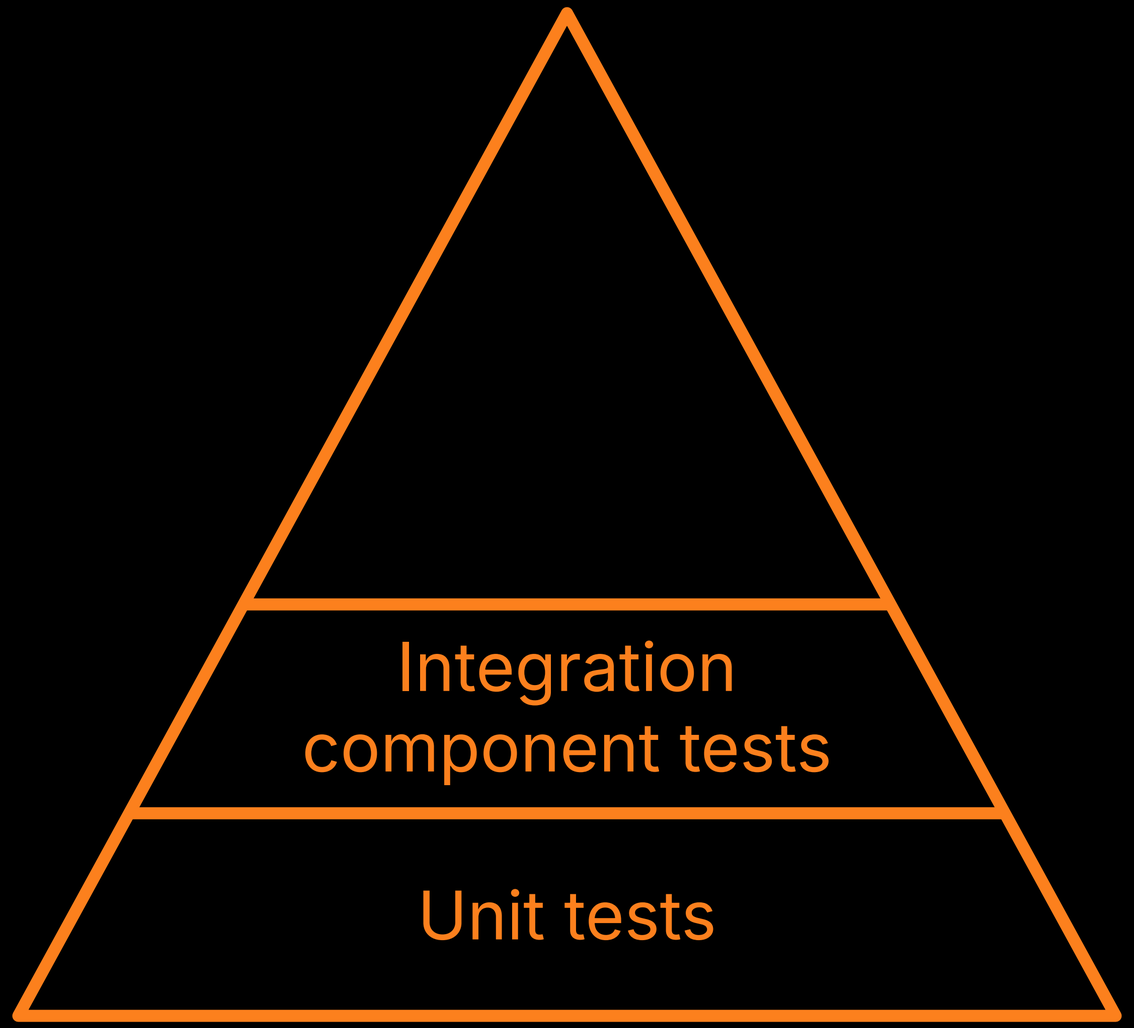
Autotests — PICT

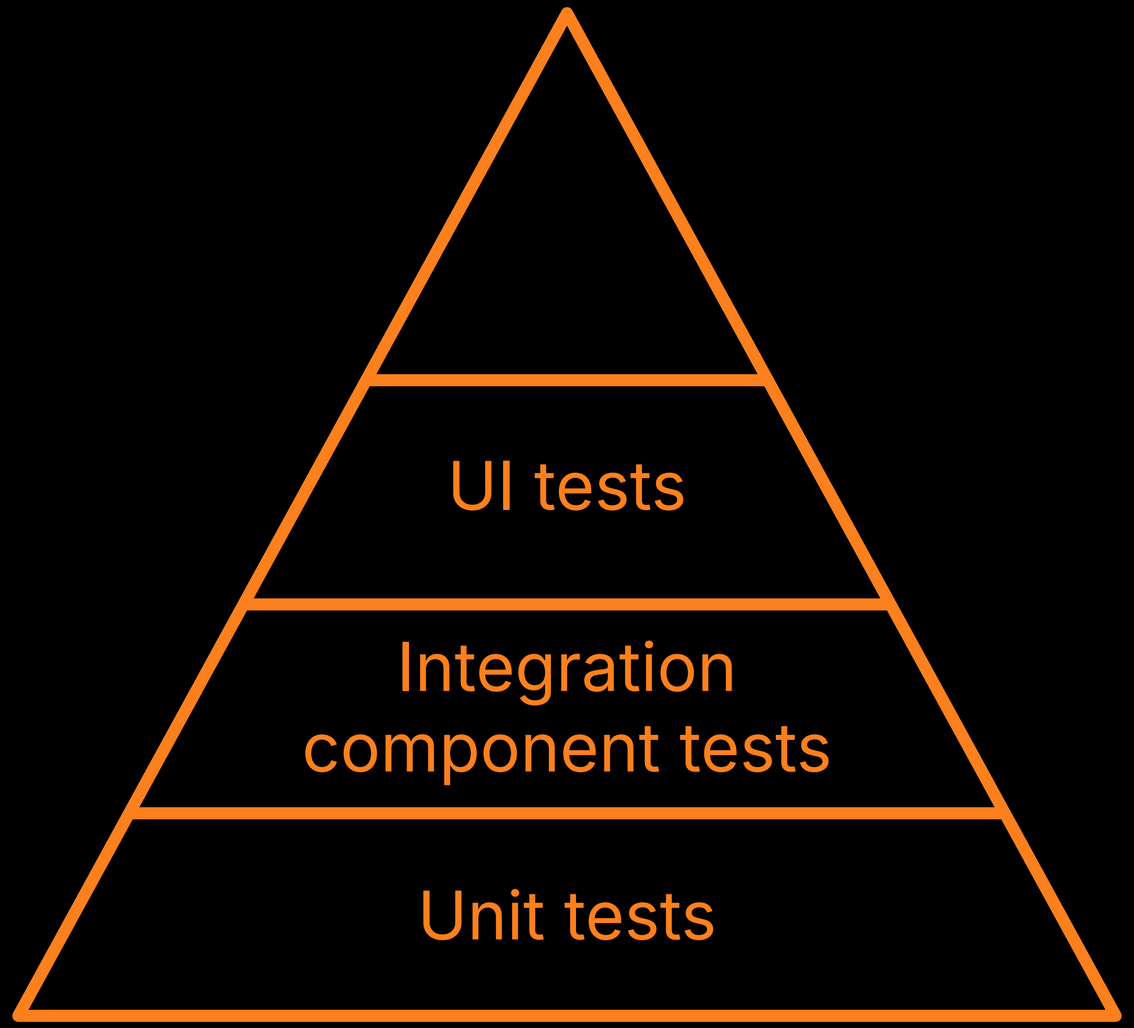
Mode	Optimization Mode	Cache Mode	GC Type	GC Scheduler	Allocator	Thread State Checker	Target	Test Set
TwoStage	Debug	StaticOnlyDist	PMCS	Adaptive	Custom	Disabled	LINUX_X64	onlyK2
TwoStage	Debug	StaticPerFile	CMS	Adaptive	Custom	Disabled	LINUX_X64	onlyK2
TwoStage	Debug	StaticEverywhere	CMS	Adaptive	Custom	Disabled	IOS_SIMULATOR_ARM64	onlyK2
TwoStage	Optimized	No	PMCS	Adaptive	Custom	Disabled	IOS_SIMULATOR_ARM64	onlyK2
TwoStage	Debug	StaticEverywhere	PMCS	Adaptive	Custom	Disabled	LINUX_X64	onlyK2
TwoStage	Debug	StaticPerFile	PMCS	Adaptive	Custom	Disabled	IOS_SIMULATOR_ARM64	onlyK2
TwoStage	Debug	No	CMS	Adaptive	Custom	Disabled	MINGW_X64	onlyK2
TwoStage	Debug	StaticOnlyDist	CMS	Adaptive	Custom	Disabled	IOS_SIMULATOR_ARM64	onlyK2
TwoStage	Optimized	No	CMS	Adaptive	Custom	Disabled	LINUX_X64	onlyK2
TwoStage	Debug	No	PMCS	Adaptive	Custom	Disabled	MINGW_X64	onlyK2

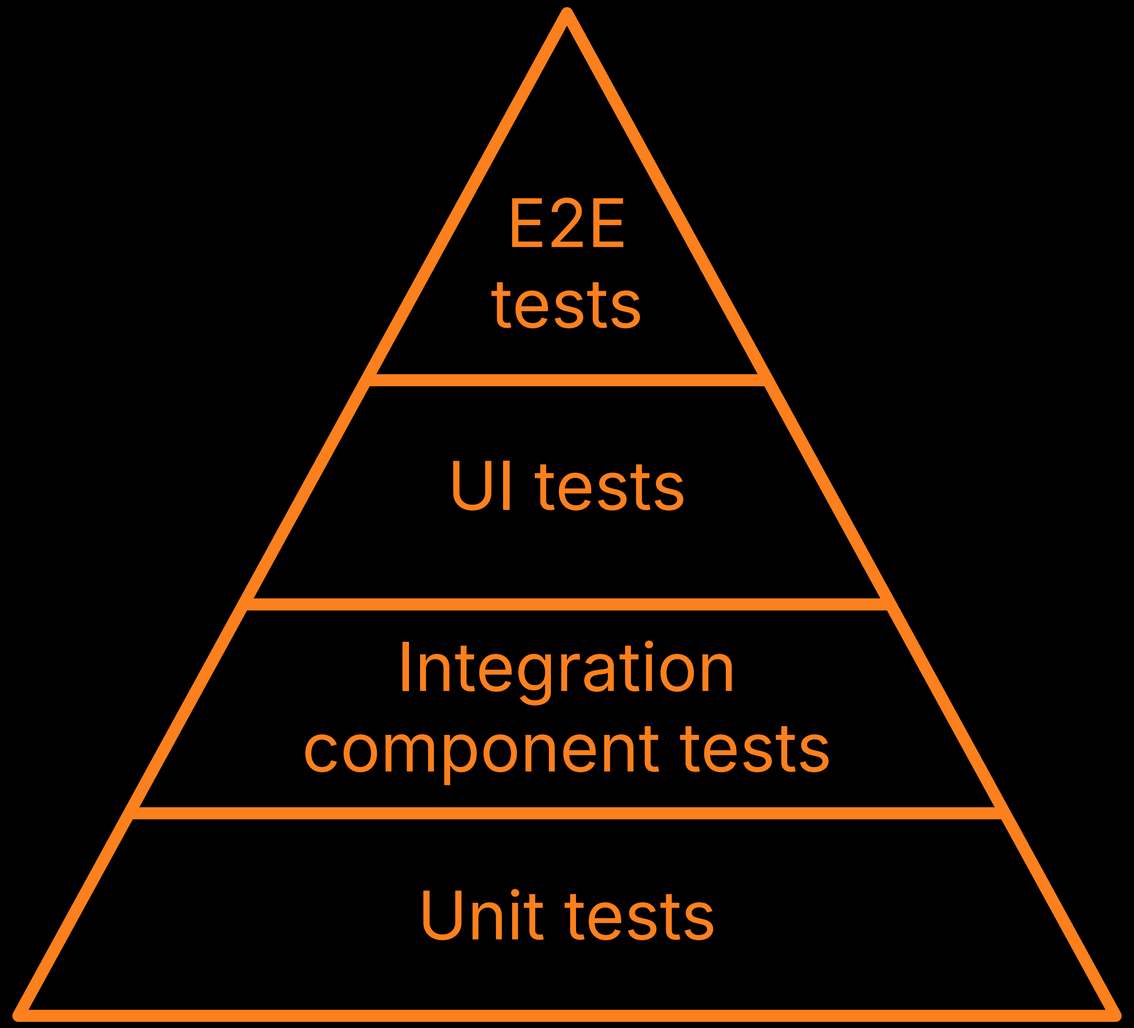

Kotlin is a great language
Such a pleasure to test it

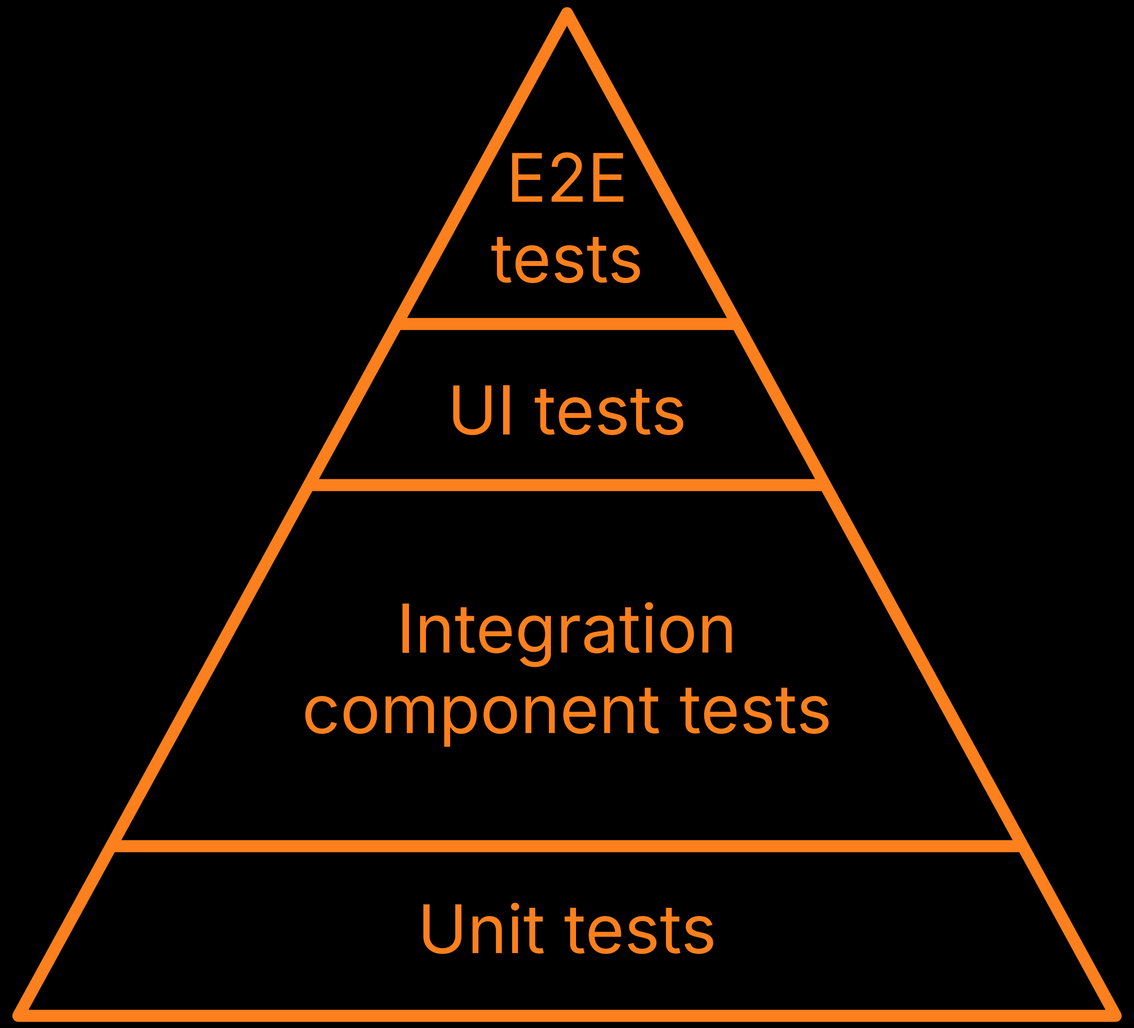
Levels of testing

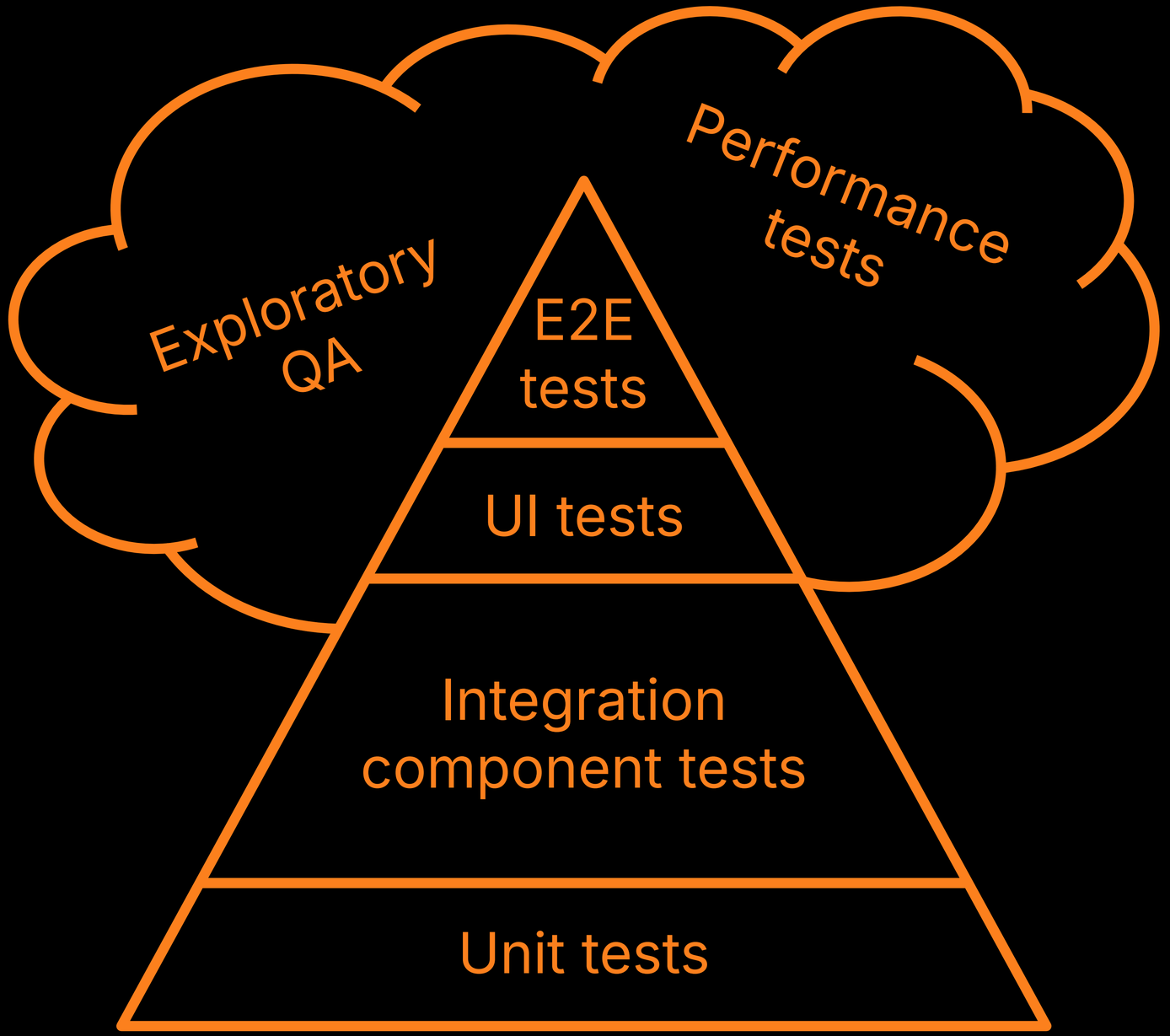

Unit tests


Integration
component tests
Unit tests


UI tests
Integration
component tests
Unit tests


E2E
tests
UI tests
Integration
component tests
Unit tests


E2E
tests
UI tests
Integration
component tests
Unit tests


Performance tests
Exploratory QA
E2E tests
UI tests
Integration component tests
Unit tests

It’s all fine. But what did we miss?

How we measure quality

How we measure quality

Vsevolod Tolstopyatov

Kotlin Project Lead

“Hey Artur, do we have a reliable way to measure quality?”

How we measure quality

Vsevolod Tolstopyatov

Kotlin Project Lead

“Hey Artur, do we have a reliable way to measure quality?”

🚪🏃

How we measure quality

Or the excuses not to

- Coverage model is very complicated and expensive
  - Instead – the quality criterias in each QA team

How we measure quality

Or the excuses not to

- Coverage model is very complicated and expensive
  - Instead – the quality criterias in each QA team

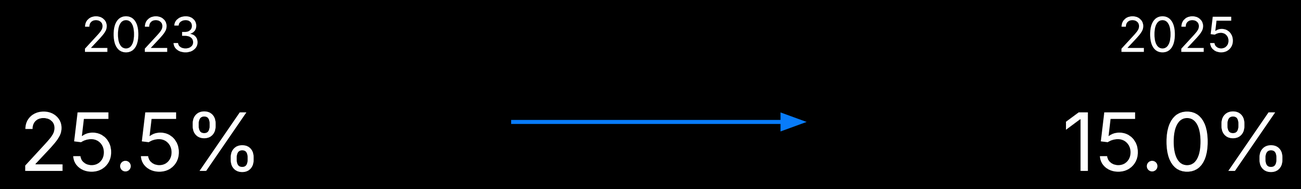
  ⇒

- **Feedback** ❗
  - Monitor regressions reported by users vs devs + QAs
  - Closely look at the JetBrains internal developers issues
  - Analyze the internal and external survey data

How we measure quality

Or the excuses not to

According to Kotlin Developer Survey, the dissatisfaction with Quality and Stability improved a lot



How we measure quality

Or the excuses not to

We are lucky to work without coming up with pseudo-metrics of quality :)

How we measure quality

Or the excuses not to

We are lucky to work without coming up with pseudo-metrics of quality :)

CSAT for Kotlin ~90% 🎉

CSAT for Kotlin support in IDE ~85% 🎉

How we measure quality

Or the excuses not to

We are lucky to work without coming up with pseudo-metrics of quality :)

CSAT for Kotlin ~90% 🎉*

CSAT for Kotlin support in IDE ~85% 🎉*

* but of course there is a room for improvements

How we measure quality

Releases

We Are Looking For EAP Champions!

Ekaterina Petrova
November 23, 2022

The Kotlin Early Access Preview (EAP) is a program where the Kotlin team ships a few Beta and Release Candidate builds before every release. Any Kotlin developer can participate in the Early Access Preview, try out the latest Kotlin features before they are released, and help the Kotlin team gather the feedback necessary to stabilize a new language version.

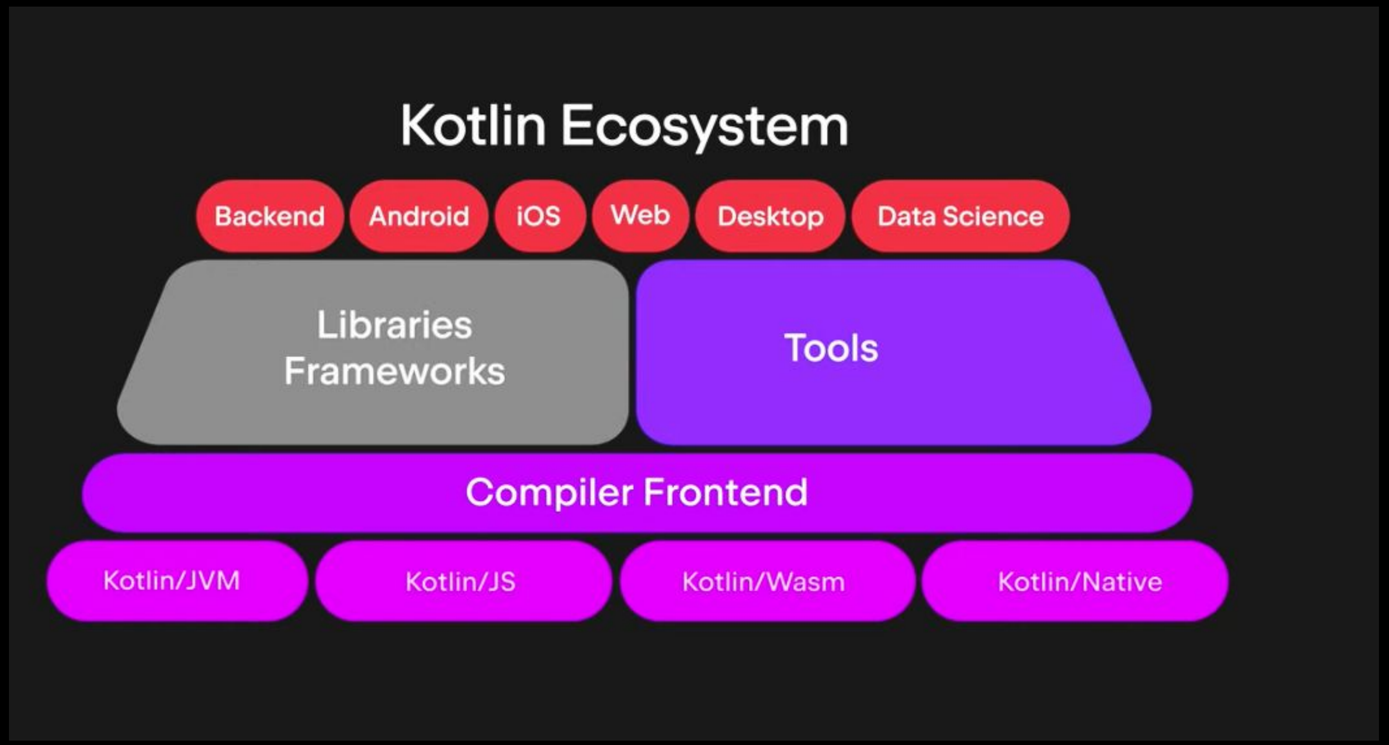
Square

> How to install Kotlin 2.2.20
>
> If you run into any problems:
>
> Special thanks to our EAP champions 🥇👏
>
> Further reading

Conclusions

Takeaways

- Programming language is an entire ecosystem

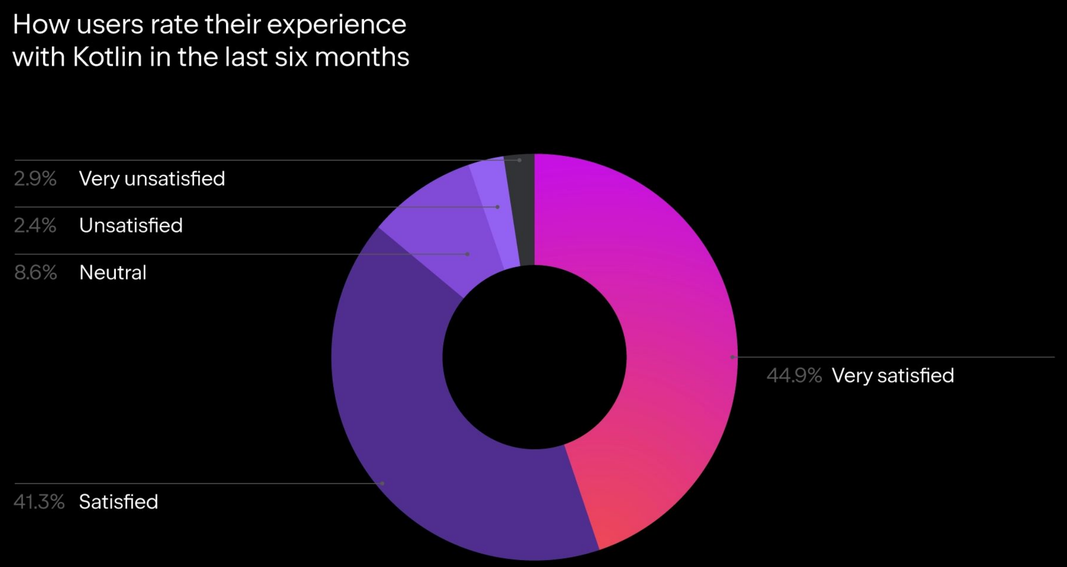


Takeaways

- Programming language is an entire ecosystem
- In such a complex system imitation of a user from different angles is a must

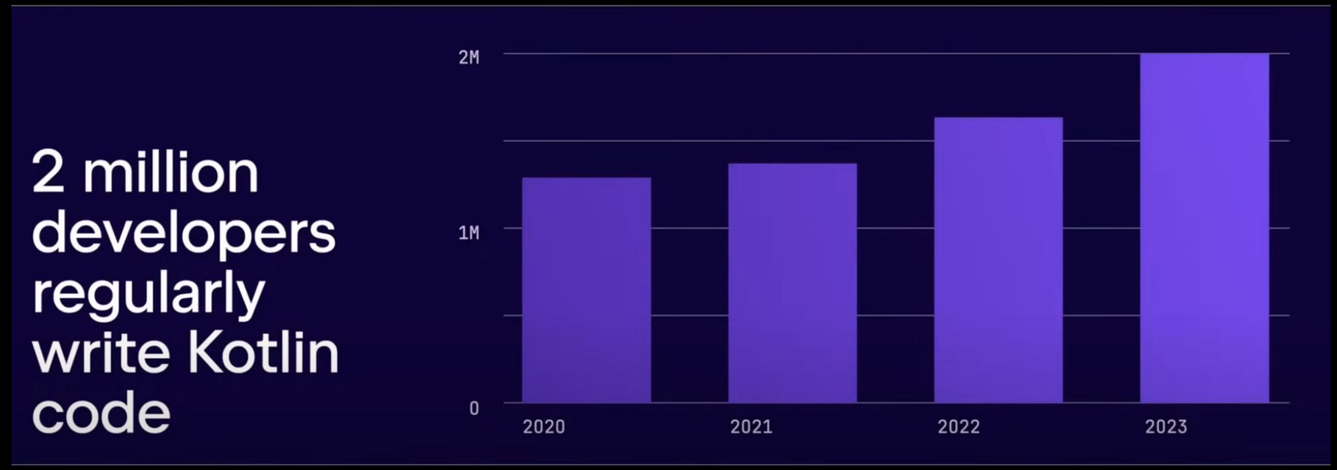
Takeaways

- Programming language is an entire ecosystem
- In such a complex system imitation of a user from different angles is a must
- We don’t need to calculate coverage to satisfy our users



Takeaways

- Programming language is an entire ecosystem
- In such a complex system imitation of a user from different angles is a must
- We don’t need to calculate coverage to satisfy our users
- Even though it is tooling for developers, it is still a product that depends on the feedback



JETBRAINS

Artur

Do connect with us on LinkedIn!

Alexander

Thank you!

JETBRAINS

We are hiring!

Thank you!