



Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# The State of Kotlin Multiplatform — Webinar 2023



Jorge Luis Castro Medina · [Follow](#)

6 min read · Dec 3, 2023



265



1



...

## Kotlin Multiplatform

Share code on your terms

Reuse code across Android, iOS, web, desktop, and server-side while keeping native code if needed.



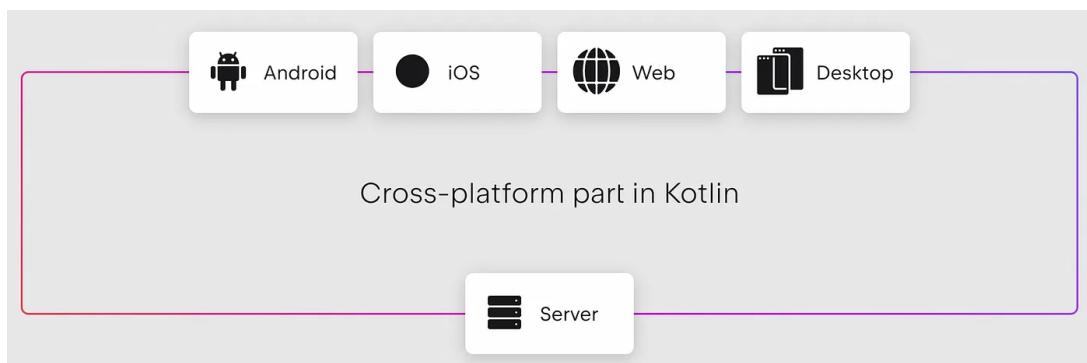
In 2023, Kotlin Multiplatform has garnered significant attention for its development, current status, and future potential. As more developers express interest in adopting KMP for cross-platform solutions, JetBrains responded with a series of webinars in the latter half of November. The inaugural webinar, focusing on the **The State of Kotlin Multiplatform**, aimed to address developer inquiries. Today, I'm thrilled to share a concise summary of the key insights from this informative session 😊.



[Svetlana Isakova & Márton Brau](#)

Let's get started!

## What is Kotlin Multiplatform?



*Kotlin Multiplatform allows you to create applications for various platforms and efficiently reuse code across them while retaining the benefits of native programming. Your multiplatform applications will work on different operating systems, such as iOS, Android, macOS, Windows, Linux, and others.*

— [Kotlin Multiplatform documentation](#)

The KMP code will be compiled into a .jar file for Android and a .framework file for iOS. Subsequently, we can incorporate it as a standard dependency in the Android/iOS project.

## Kotlin Multiplatform Is Stable and Production-Ready

The Jetbrain team [announced](#) that KMP is now stable a couple weeks ago, which means we can use it in production without any fear, of course, many

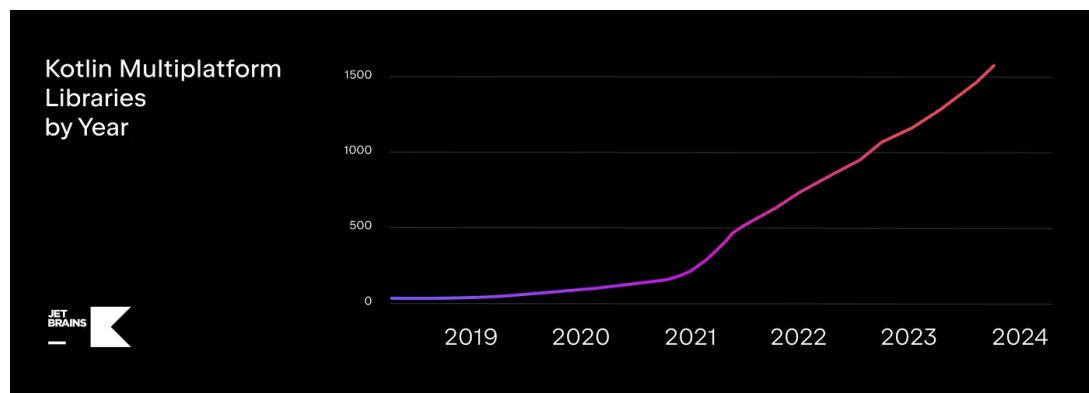
projects had been created with KMP, before this notice.



Having Kotlin Multiplatform stable gives us:

- Strict compatibility guarantees: Check [kotlinlang.org/kmp-stability](https://kotlinlang.org/kmp-stability)
- Easier configuration
- Faster builds
- Better runtime performance
- New documentation

## Libraries



Since Beta, the number of KMP libraries has multiplied, offering diverse tools for networking, data storage, arithmetics, analytics, and more, enabling swift and confident app development.

If you want to see a list of such libraries, a good resource could be [Awesome Kotlin Multiplatform](#)

# Awesome Kotlin Multiplatform



PRs welcome awesome stars 2.2k maven-central v2.0.0-Beta1

Kotlin Multiplatform technology simplifies the development of cross-platform projects. It reduces time spent writing and maintaining the same code for different platforms while retaining the flexibility and benefits of native programming.

This list contains libraries which support iOS and Android targets in first place.

## Contents

<a href="#">Tooling</a>	<a href="#">Log</a>	<a href="#">Network</a>
<a href="#">Storage</a>	<a href="#">Device</a>	<a href="#">Dependency Injection</a>
<a href="#">Architecture</a>	<a href="#">Analytics</a>	<a href="#">Test</a>
<a href="#">Crypto</a>	<a href="#">File</a>	<a href="#">Language extensions</a>
<a href="#">Serializer</a>	<a href="#">Date-Time</a>	<a href="#">Asynchronous</a>
<a href="#">Compose UI</a>	<a href="#">Graphics</a>	<a href="#">Service SDK</a>
<a href="#">Arithmetic</a>	<a href="#">Resources</a>	<a href="#">Utils</a>

## Successful Stories

# Kotlin Multiplatform stories

kotl.in/kmp-case-studies

The grid displays case studies for several companies:

- TOUCHLAB**: Touchlab is a group of mobile professionals focused on...
- LEROY MERLIN**: Leroy Merlin is one of the largest international D...
- AUTODESK**: Autodesk is a multinational software...
- down dog**: Down Dog is a start-up company founded in California. Their app brings yoga to over 500k subscribers on both Android and iOS, with over 100k users completing a...
- Baidu**: Baidu is a leading artificial intelligence and the world's largest search engine. They...
- vmware**: VMware is an international software...
- Magic Technology**: Magic Technology provides software products to clients from various industries. It saves time and resources by using a K...
- 9GAG**: 9GAG opted for Kotlin Multiplatform Native after trying both Flutter Native. They gradually ad...
- fastwork**: fastwork is one of the largest professional freelancing platforms in Southeast Asia. In terms of both the number of users it...
- Unflow**: Unflow is a next-generation management system. Built on top of KMM, it allows you to create flows, and features, without...
- Careem**: Careem was established in 2012 and is now one of the largest ride-hailing companies in the Middle East and Africa. It has over 10 million users...
- chalk**: chalk is a platform for creating and sharing digital chalkboards. It uses KMM to provide a native experience on both iOS and Android...
- Worldline**: Worldline, in a strategic move to revamp its mobile payment service, chose to migrate from Java to Kotlin Multiplatform. The...
- netvirta**: NetVirta offers an array of solutions, including KMM-based mobile applications...
- MIREGO**: Mirego is an end-to-end digital product team. They build digital ecosystems that...
- Cash App**: Cash App is a mobile payment service that...

You can find the complete list of stories in the following [link](#), where you will discover the challenges people faced during the adaptation period, the various design choices they made, and the advantages they found after implementing KMP

## Default hierarchy template

One of the improvements the JetBrains team has been working on is enhancing the build configuration experience to make it easier to get started with Kotlin Multiplatform. Let's take a look at the code below.

```

kotlin {
    androidTarget()
    iosArm64()
    iosSimulatorArm64()

    sourceSets {
        val commonMain by getting
        val iosMain by creating {
            dependsOn(commonMain)
        }
        val iosArm64Main by getting {
            dependsOn(iosMain)
        }
        val iosSimulatorArm64Main by getting {
            dependsOn(iosMain)
        }
    }
}

```

### Configuring Targets

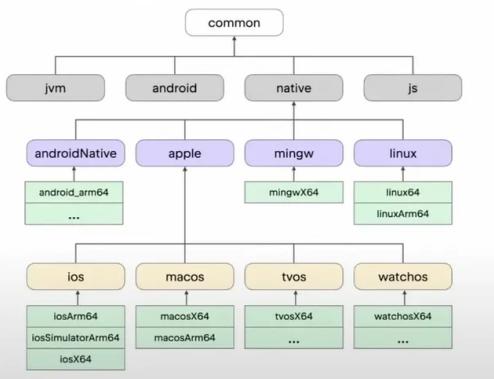
Now, let's review the `sourceSets` section; it contains the directories for the different targets of the KMP project. Even in the most common scenarios, some work will be needed to set them up. For example, if we consider the case of iOS, we'll need to create different `sourceSets` for the various architectures. Starting with Kotlin 1.9.20, simply declare the targets you need, and the `sourceSets` will be generated for you automatically. KMP uses something called the Default Hierarchy Template for this.

## Default hierarchy template

```

kotlin {
    androidTarget()
    iosArm64()
    iosSimulatorArm64()
}

```

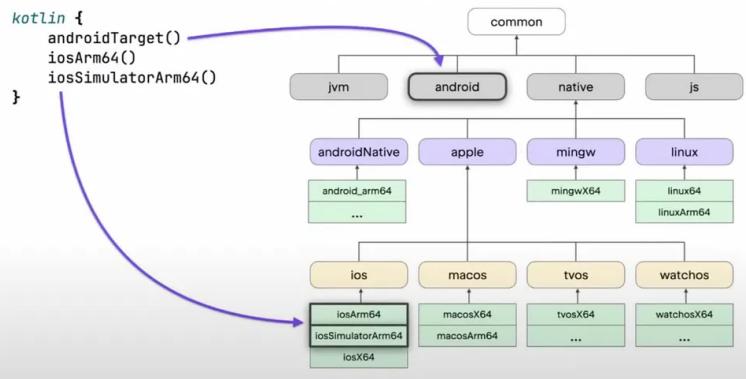


## Default hierarchy template

```

kotlin {
    androidTarget()
    iosArm64()
    iosSimulatorArm64()
}

```



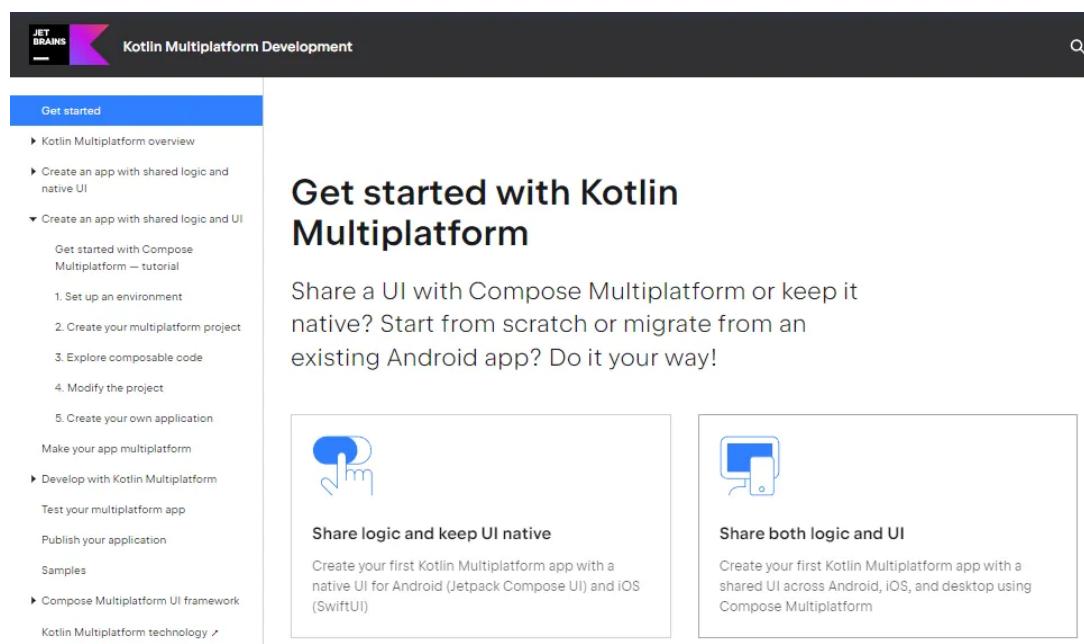
You can use `sourceSets` to define dependencies for each platform.

# Source set completion

```
kotlin {  
    androidTarget()  
    iosArm64()  
    iosSimulatorArm64()  
  
    sourceSets {  
        commonMain.dependencies {  
            implementation(libs.ktor.client.core)  
        }  
        androidMain.dependencies {  
            implementation(libs.ktor.client.okhttp)  
        }  
        iosMain.dependencies {  
            implementation(libs.ktor.client.darwin)  
        }  
    }  
}
```

## Get started with Kotlin Multiplatform

The JetBrains team has worked on another thing, a new documentation portal. It contains a detailed guide on how to start with KMP, how to publish your app, samples, and more. You can open it with just one click on the following link: [kotl.in/kmp-portal](https://kotl.in/kmp-portal).



The screenshot shows the official documentation for Kotlin Multiplatform Development. At the top, there's a navigation bar with the Jet Brains logo and the text "Kotlin Multiplatform Development". On the right side of the bar is a search icon. Below the bar, a sidebar on the left contains a "Get started" button, followed by a list of links: "Kotlin Multiplatform overview", "Create an app with shared logic and native UI", "Create an app with shared logic and UI", "Get started with Compose Multiplatform — tutorial", "1. Set up an environment", "2. Create your multiplatform project", "3. Explore composable code", "4. Modify the project", "5. Create your own application", "Make your app multiplatform", "Develop with Kotlin Multiplatform", "Test your multiplatform app", "Publish your application", "Samples", and "Compose Multiplatform UI framework". A "Kotlin Multiplatform technology" link with a dropdown arrow is also present. The main content area features a large heading "Get started with Kotlin Multiplatform" with a subtext: "Share a UI with Compose Multiplatform or keep it native? Start from scratch or migrate from an existing Android app? Do it your way!". Below this, there are two cards: "Share logic and keep UI native" (with an icon of a brain) and "Share both logic and UI" (with an icon of a smartphone). Both cards have brief descriptions and links to further resources.

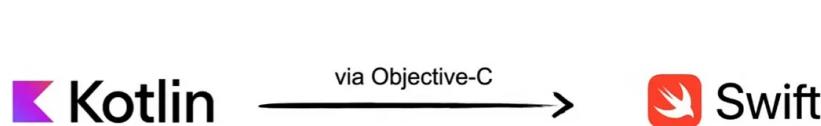
## Kotlin-Swift interopedia

During the webinar, a little about **Kotlin-Swift interopedia** is also mentioned, which refers to the current state of interoperability between Kotlin and Swift. This is a repository that was born from the community and improved by the JetBrains team. It covers different aspects such as functions,

exceptions, Data class, and so many others. check the following repository to see more: [kotlinlang.org/intro/interopedia](https://kotlinlang.org/intro/interopedia)

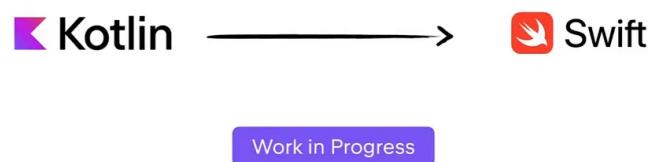
## Current Kotlin-to-Swift interoperability

Common Kotlin APIs can be used from the Swift code



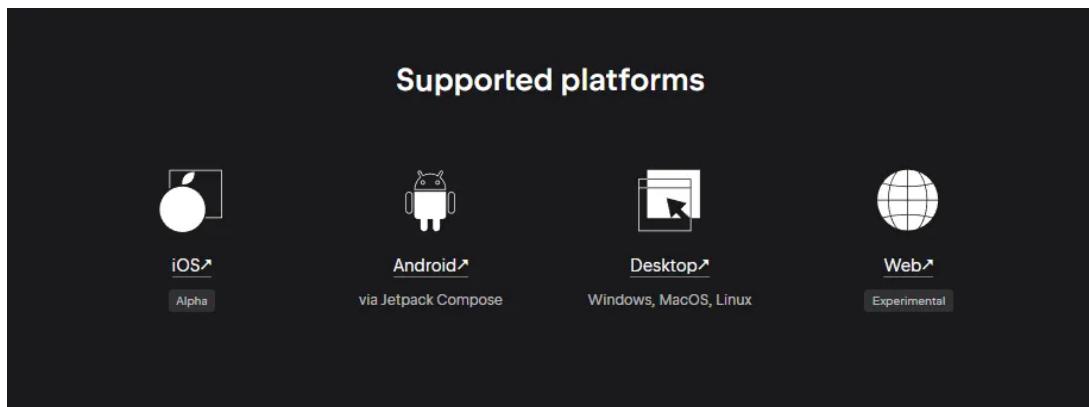
## Future plans: Direct Kotlin-to-Swift interoperability

Making common Kotlin API look more natural from the Swift side



## Compose Multiplatform

Try Compose Multiplatform: [jb.gg/compose](https://jb.gg/compose)

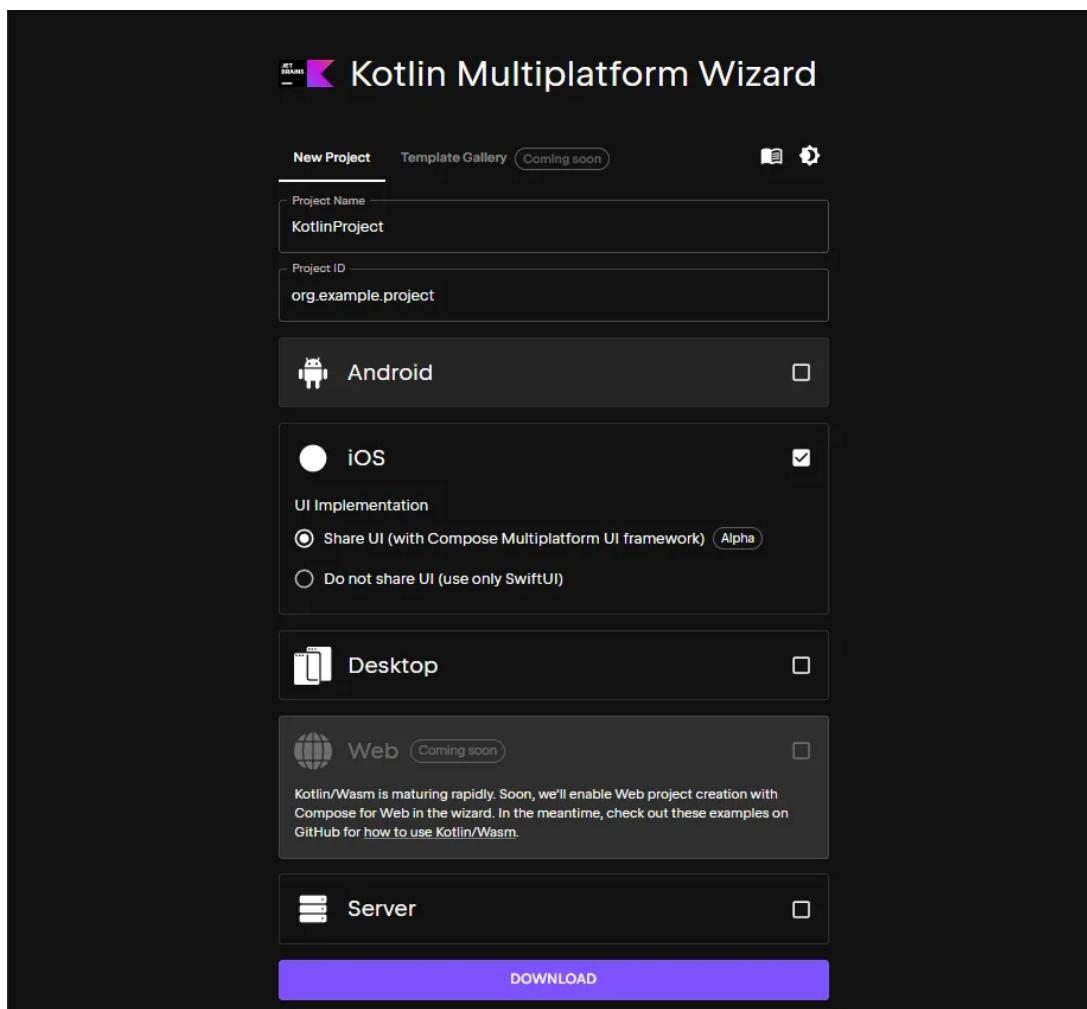


Recent improvements: 1.5.0

- Dialogs, popups
- Window insets
- Natural scrolling for iOS
- 120 Hz refresh rate
- Stabilized test framework for desktop
- All Material 3 components now available: ModalBottomSheet, SearchBar, DockedSearchBar, ExposedDropdownMenuBox
- TextField improvements for iOS
- Compiler caching compatible, faster incremental build
- K2 support preview

## Kotlin Multiplatform Wizard

Since now, we have a new easier way to create KMP Projects called Kotlin Multiplatform Wizard. This is a web-based wizard that provides the latest state and dependency versions. You can open it from the following link: [kmp.jetbrains.com](https://kmp.jetbrains.com). For instance, you can choose the platforms you want to target and whether you want to share the UI or not.



[kmp.jetbrains.com](https://kmp.jetbrains.com)

## Fleet

Now it is time to talk about Fleet, and at this point, a small comparison of this tool with Android Studio is made. Normally, Android Studio is the IDE used to work on a KMP project, but we have a small detail, and that is that Swift files are listed as unsupported text files, whereas Fleet arrives with compatibility for Swift 😎🚀.

But what is Fleet? Fleet is the next-generation IDE by JetBrains, built from scratch, based on 20 years of experience developing IDEs. JetBrains Fleet

uses the IntelliJ code-processing engine, with a distributed IDE architecture and a reimagined UI. It is important to mention that Fleet is currently in preview.

# Fleet

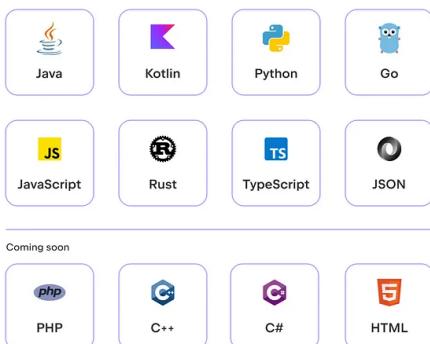
## Public Preview



<https://www.jetbrains.com/fleet/>

Fleet is polyglot, which means it supports many languages. The most interesting for us is that it supports Kotlin and Swift.

## Fleet is polyglot



## Language or stack

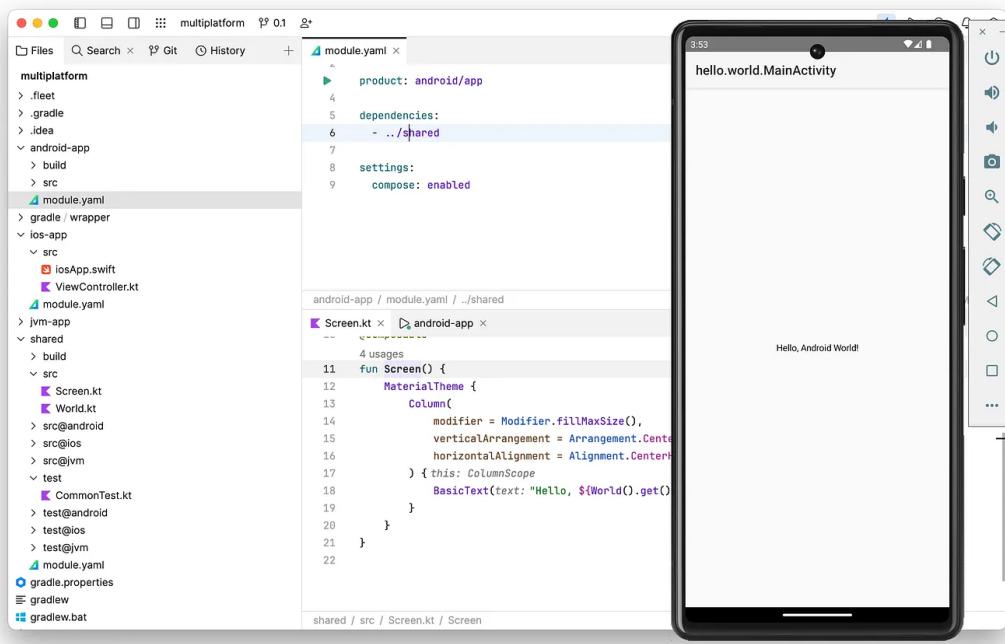
JS	JSX	CSS	SASS	LESS
Markdown	Go	Rust	TOML	Python
C#	CPP	Swift	Java	Kotlin
TypeScript	SQL	YAML	XML	Shell
HTML	PHP	Groovy	Maven	Gradle
SCSS	JSON			

## Fleet: Summing it up

- The next-generation polyglot IDE by JetBrains
  - Provides cross-language support for Kotlin and Swift
  - Is free during a public preview

This is the latest tool recently announced by the JetBrains team. Amper is a tool for project configuration. Its goal is to enhance the user experience in project configuration and the ease of use of tools, that is, compatibility within the IDE, while providing a straightforward initial experience.

## Amper



Here is an example of Amper's manifest file for a Kotlin Multiplatform shared library that can be used with JVM, Android, and iOS applications:

```

product:
  type: lib
  platforms: [ jvm, android, iosArm64, iossimulatorArm64, iosX64 ]

# Shared Compose Multiplatform dependencies:
dependencies:
  - org.jetbrains.compose.foundation:foundation:1.5.0-rc01: exported
  - org.jetbrains.compose.material3:material3:1.5.0-rc01: exported

# Android-only dependencies
dependencies@android:
  # Integration compose with activities
  - androidx.activity:activity-compose:1.7.2: exported
  - androidx.appcompat:appcompat:1.6.1: exported

# iOS-only dependencies with a dependency on a CocoaPod
# Note that CocoaPods dependencies are not yet implemented in the prototype
dependencies@ios:
  - pod: 'FirebaseCore'
    version: '~> 6.6'

settings:
  # Enable Kotlin serialization
  kotlin:
    serialization: json

# Enable Compose Multiplatform framework
compose: enabled

```

You can start by checking Amper's GitHub repository where you will find more details and many usage examples: [github.com/JetBrains/amper](https://github.com/JetBrains/amper)

## Links:

Github: [github.com/JetBrains/amper](https://github.com/JetBrains/amper)

Slack channel: [jb.gg/amper-slack](https://jb.gg/amper-slack)

## Q&A

In this article, I didn't cover the questions and answers as they take up approximately 1 hour of the webinar. I recommend checking the video at the end of the talk in the Q&A section for more details.

## Sources

- [kotlin-multiplatform-stable](#)
- [Strict compatibility guarantees](#)
- [Case studies](#)
- [Kotlin Multiplatform Wizard](#)

- [Fleet](#)
- [Amper](#)

If you like my content and want to support my work, you can give me a cup of coffee ☕ 😊



## Follow me in

- Twitter: [@devjcastro](#)
- Linkedin: [devjcastro](#)

Kotlin Multiplatform

Kotlin

Android App Development

IOS App Development

Mobile App Development



**Written by Jorge Luis Castro Medina**

890 Followers

Follow



I'm a Software Engineer passionate about mobile technologies, and I like everything related to software design and architecture

More from Jorge Luis Castro Medina



 Jorge Luis Castro Medina

## Update your Android app signing key from a new Keystore with Play...

Here is a short article to explain how to update your signing key from a keystore whe...

5 min read · Feb 17, 2023

 20 

+ 

 Jorge Luis Castro Medina

## How to build a CI/CD Pipeline for Android with Jenkins and Docker...

Jenkins in Docker: A Solution for Android CI/CD

8 min read · Oct 12, 2023

 144 

+ 



 Jorge Luis Castro Medina

## How to deploy your Android App to the Internal track in the Play Store...

A step-by-step guide on how to use Fastlane to deploy your Android app to the Internal...

6 min read · Jun 16, 2023

 11 

+ 

 Jorge Luis Castro Medina

## Kotlin Multiplatform—This is the way.

Revolutionizing App Development: Unleashing the Power of Kotlin Multiplatform

5 min read · Jul 24, 2023

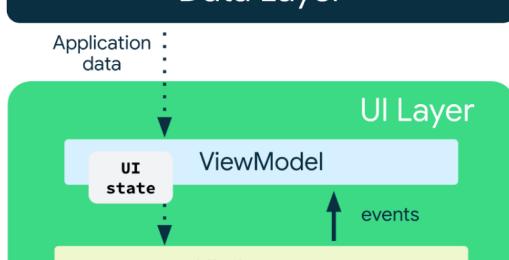
 272 

+ 



[See all from Jorge Luis Castro Medina](#)

Recommended from Medium



Berk Özyurt

## UIState in Kotlin: Basic Information and Usage Tips

Hello everyone. In this article, I'll briefly introduce you to the basic usage of UIState i...

8 min read · Feb 11, 2024

👏 143    🎧 1

Bookmark ...



Michael Szczepanik in MobilePeople

## It's time for the NATIVE mobile development to end.

Dive into the future of app development: Discover why it's time to shift from native to...

12 min read · Feb 11, 2024

👏 158    🎧 29

Bookmark ...

## Lists



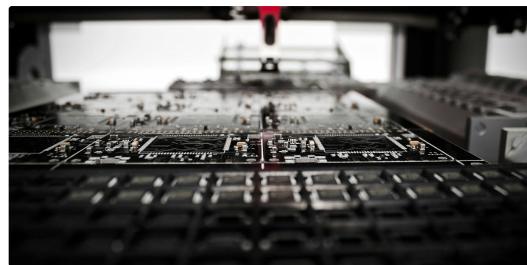
### Medium's Huge List of Publications Accepting...

262 stories · 1965 saves



### Staff Picks

584 stories · 765 saves



Eric Ampire in ProAndroidDev

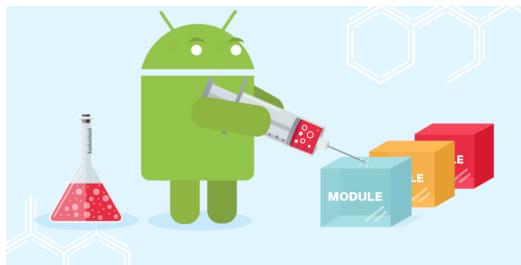
## Part 1 — Compose Multiplatform Made easy with Circuit

1. Introduction

💡 · 11 min read · Feb 12, 2024

👏 407    🎧 2

Bookmark ...



Reena Rote in Towards Dev

## Dependency Inversion Vs Dependency Injection

Dependency inversion and dependency injection are two closely related concepts in...

💡 · 3 min read · Jan 8, 2024

👏 155    🎧 1

Bookmark ...



## Retrofit vs. Ktor Client

Navigating the Old Guard and the Newcomer in Android HTTP Requests?

6 min read · Feb 5, 2024



81



...



81



1



...

---

[See more recommendations](#)

## What Every Android Developer Should Have in Their GitHub?

In the competitive world of Android development, having a strong GitHub...

3 min read · Feb 5, 2024



81



...



81



1



...