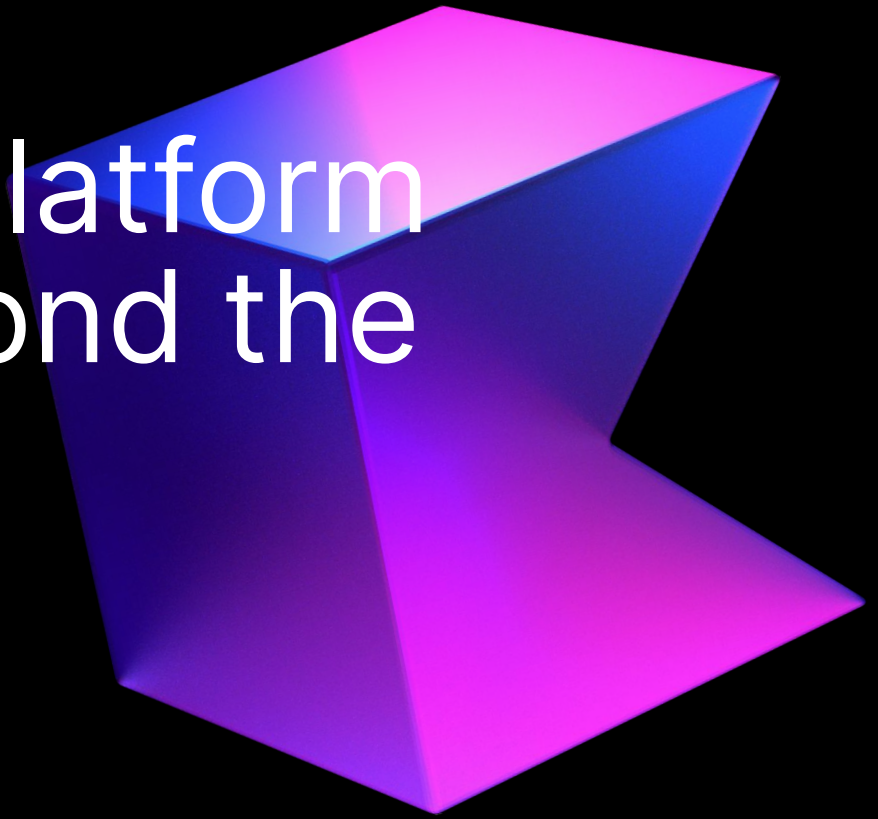
Kotlin

Kotlin Multiplatform Mobile: Beyond the Basics

Pamela Hill

@pamelaahill
pamelaahill@androiddev.social

February 21, 2023



Why should you learn about Kotlin Multiplatform Mobile?

Write shared code once and use on Android and iOS

- **Flexibility** to choose what is shared and what not
- Ability to **adopt gradually**, start simple
- Shared code helps you go **faster**
- Shared code ensures **consistency** amongst platforms

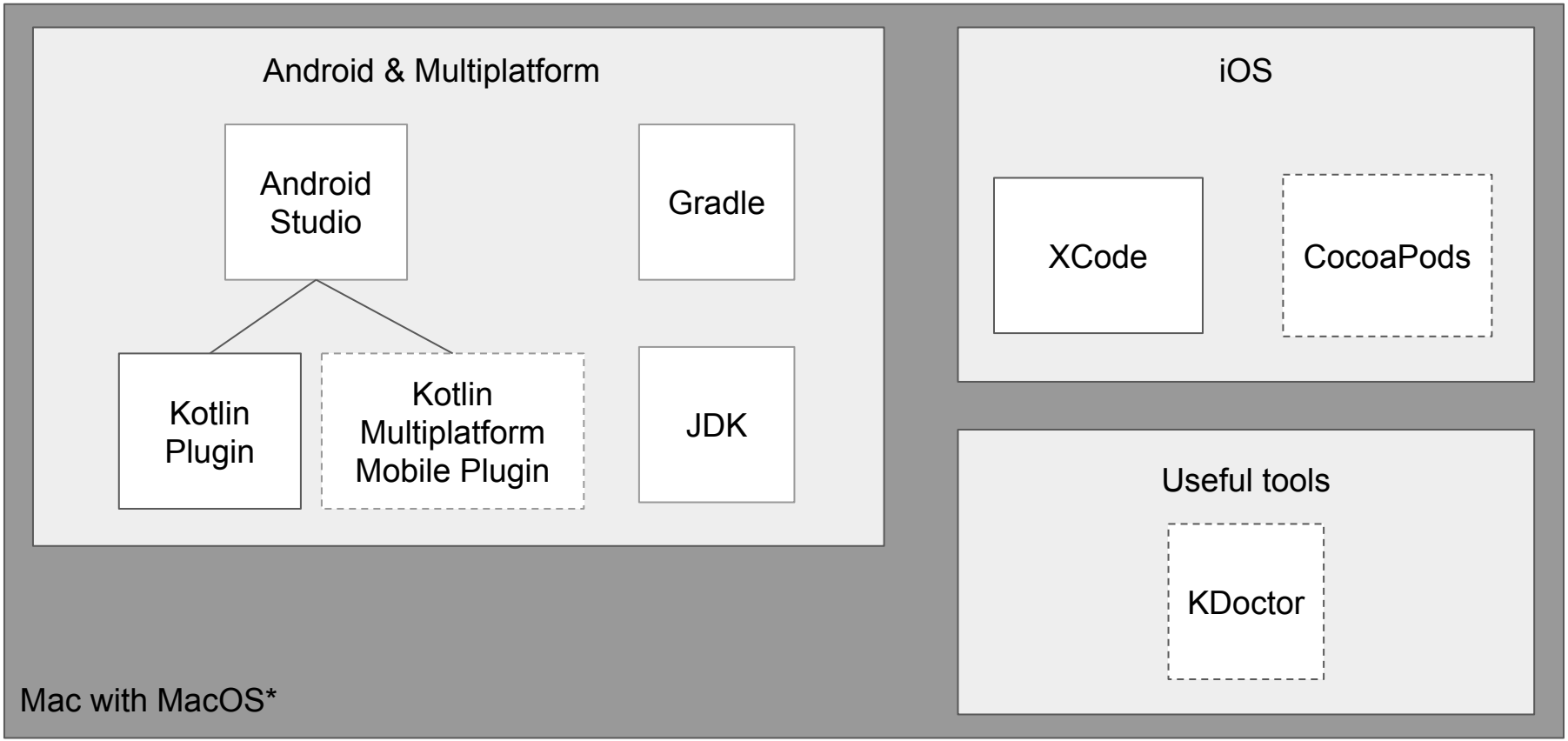
What you will learn?

- Review some Kotlin Multiplatform Mobile basics
- Use the multiplatform library Ktor to retrieve data across the network
- Transform the retrieved data from JSON to Kotlin objects using kotlinx.serialization
- Perform the network operation asynchronously using Kotlin coroutines
- Consume the retrieved data from Android and iOS for presentation on the screen

✨ Bonus: learn about Kotlin flows and how to consume them from iOS ✨

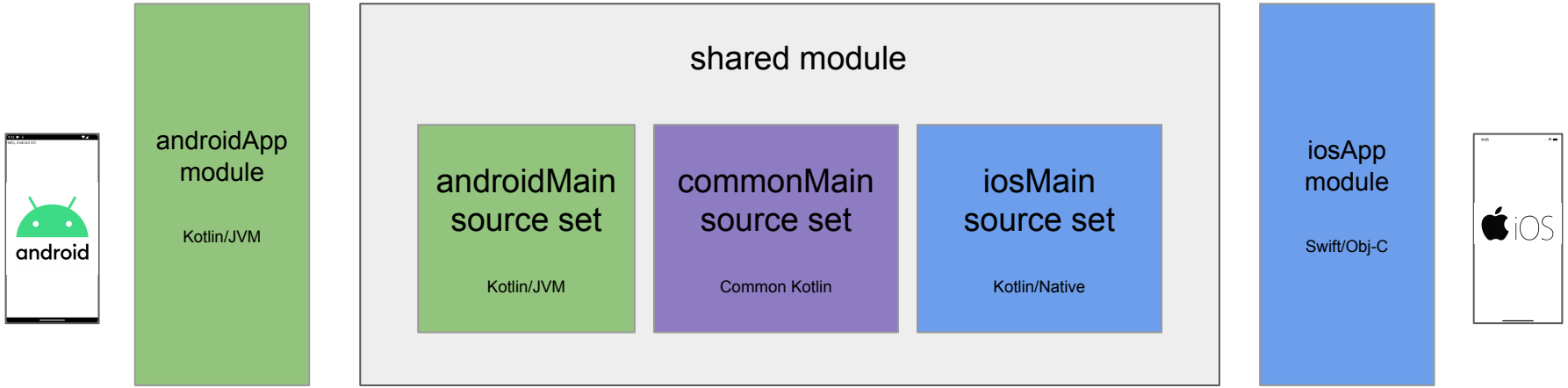
Review of Kotlin Multiplatform Mobile Basics

Kotlin Multiplatform Mobile Ecosystem



Create your (first) cross-platform app

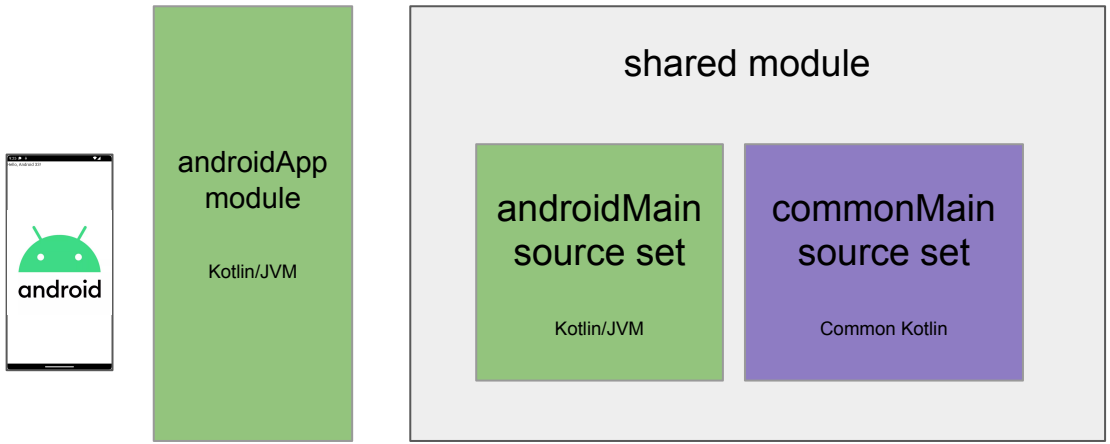
Project Structure



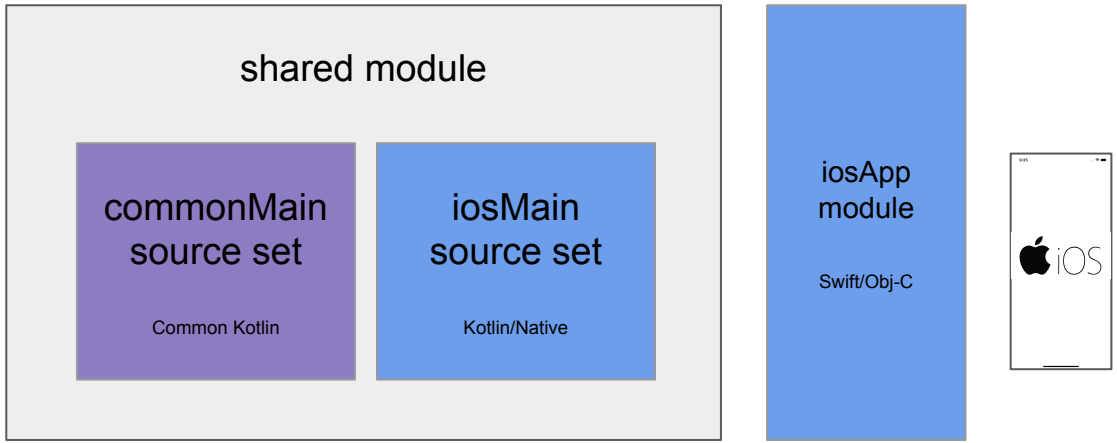
- Room
- Retrofit

- Kotlinx-serialization
- Kotlinx-datetime
- Koin

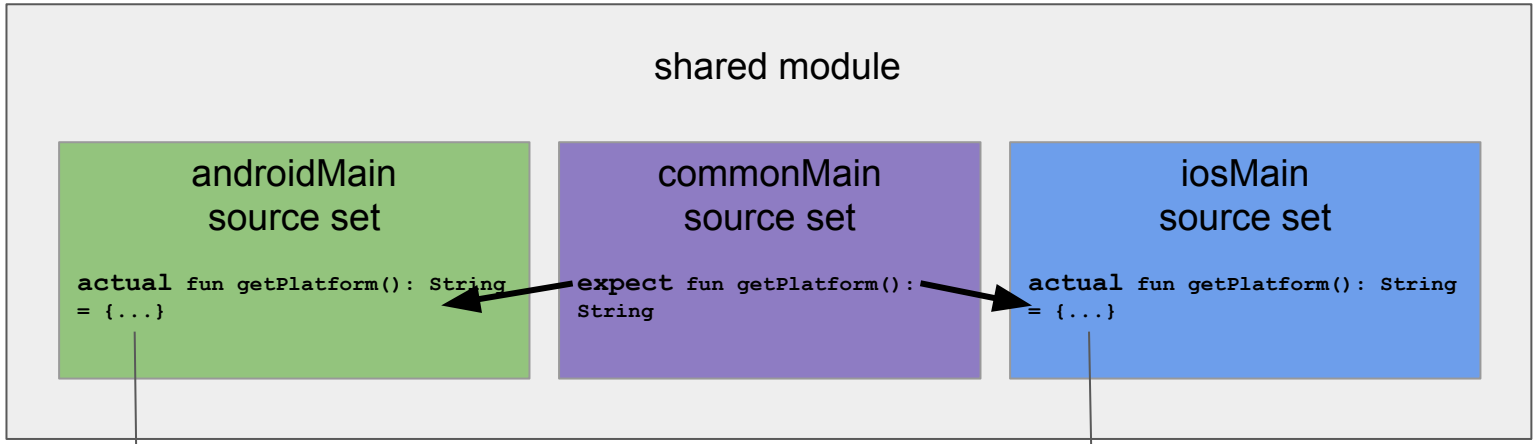
Android app contains:



iOS app contains:



expect/actual mechanism

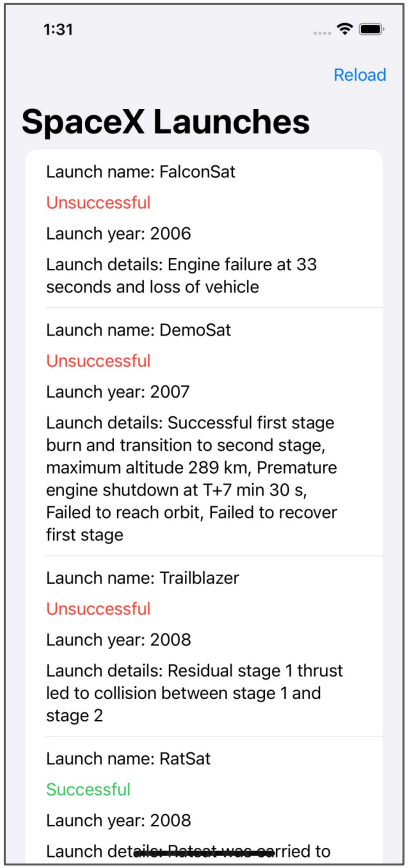


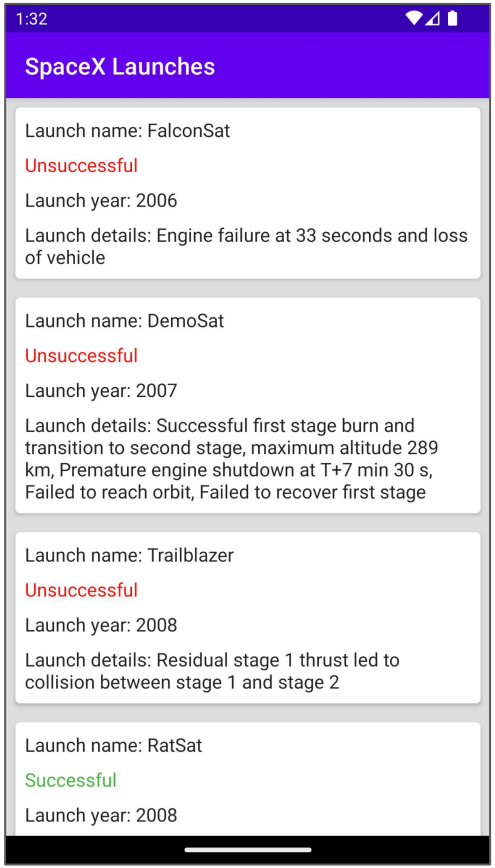

```
"Android ${android.os.Build.VERSION.SDK_INT}"
```

```
UIDevice.currentDevice.systemName() + " "
+ UIDevice.currentDevice.systemVersion
```

Create a multiplatform app using Ktor

What we will build





Steps

1. Add dependencies to the shared module
2. Create a shared data model
3. Implement the API service using Ktor
4. Build an SDK
5. Use the SDK from Android
6. Use the SDK from iOS

Bonus section: Kotlin Flows on iOS

What is a Kotlin flow?

A flow is a type of asynchronous operation that can emit **multiple values sequentially**, as opposed to suspend functions that can only return one value.

For Android, it is recommended that you replace LiveData with flows.

However, flows aren’t supported so well in iOS, as:

- There’s no cancellation support
- The generic type of the flow is lost in Objective-C, making them hard to use.

Kotlin Community to the Rescue! 🦸

Rick Clephas wrote an awesome library called KMP-Native coroutines which you can use to consume suspend functions and flows much better than completion handlers.

https://github.com/rickclephas/KMP-NativeCoroutines

Resources for the curious

Handy resources

JetBrains tutorials: https://kotl.in/kmm-kodeco

Directory of multiplatform libraries: https://github.com/terrakok/kmm-awesome

Curated list of code samples: https://kotl.in/kmm-samples

ATOM podcast on Kotlin Youtube channel: https://www.youtube.com/c/kotlin

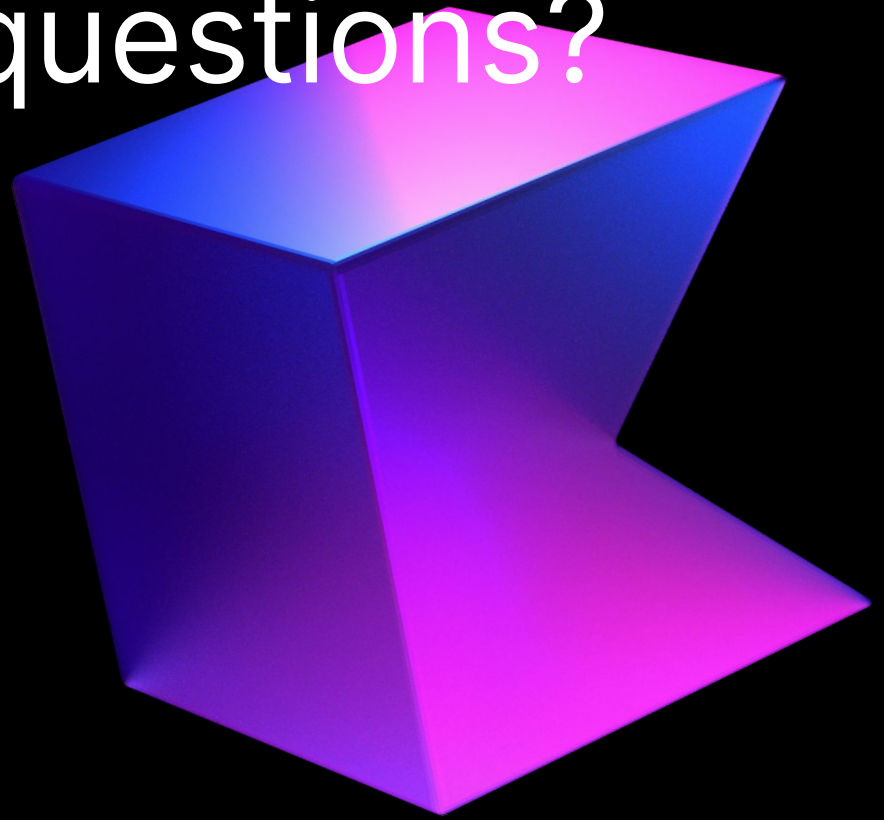
#multiplatform channel on kotlinlang Slack workspace: https://kotl.in/kotlinlang-signup

My code for the webinar

https://github.com/pahill/kmm-networking-and-data-storage

- Start branch: start_webinar
- Ktor and coroutines: webinar_step1
- Flows: final3

Thanks, any questions?



@pamelaahill
pamelaahill@androiddev.social