Breaking Down KMM: Exploring Kotlin Multiplatform Mobile for Beginners

Veronica Putri Anggraini

Software Engineer Android, eW+

bangkit

Tech Journey

Coding

- Start Coding in Android
- Got some achievement from several mobile app competition

GDSC

- Google I/O Mountain View, California
- IAK Facilitator

Career

- First Job as Trainer and Curriculum Developer
- Grace Hopper Scholarship, Bangalore

Dicoding

- External Code Reviewer Multiplatform Learning Path

Community

- GDG Lead

Career

- eW+ (LINE Bank)
- WTM Ambassador
- Google Dev Library Contributor

2016 | 2017 | 2018 | 2019 | 2020 | 2021 - Now

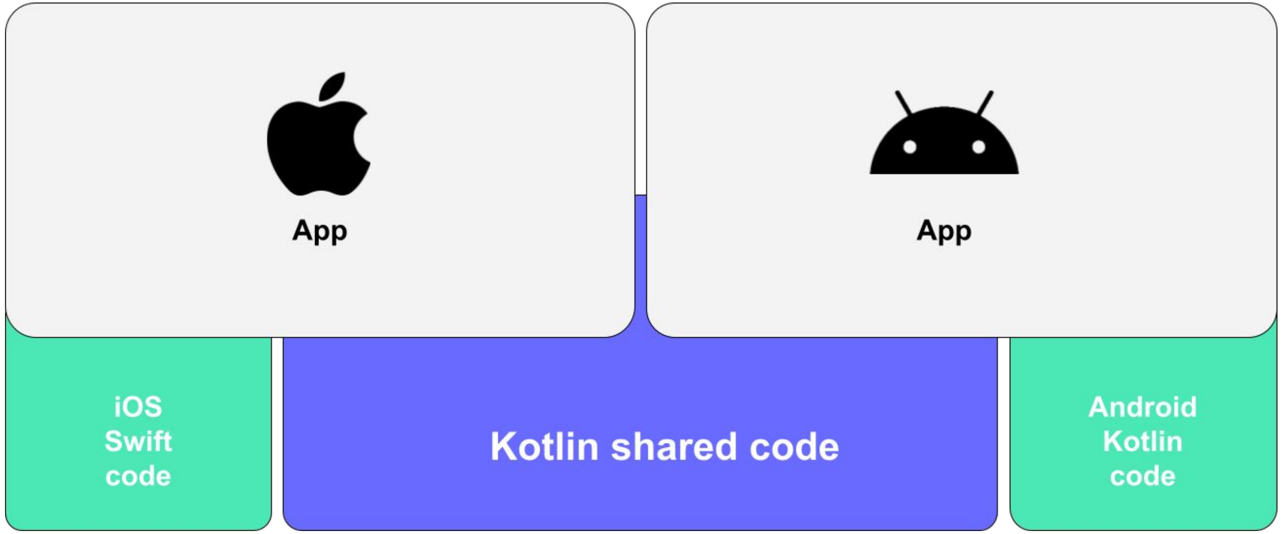
bangkit

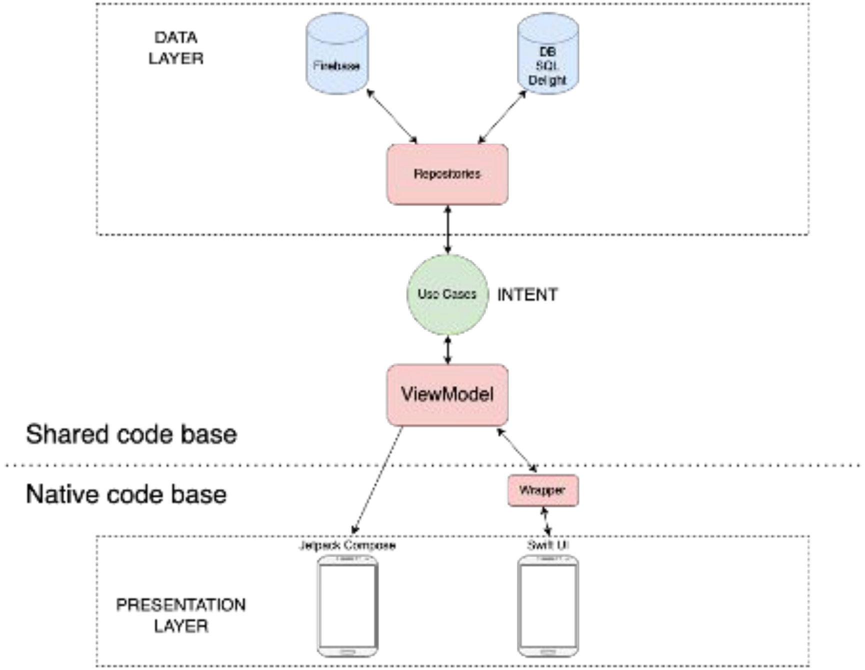
Some of Mobile Apps Development Method

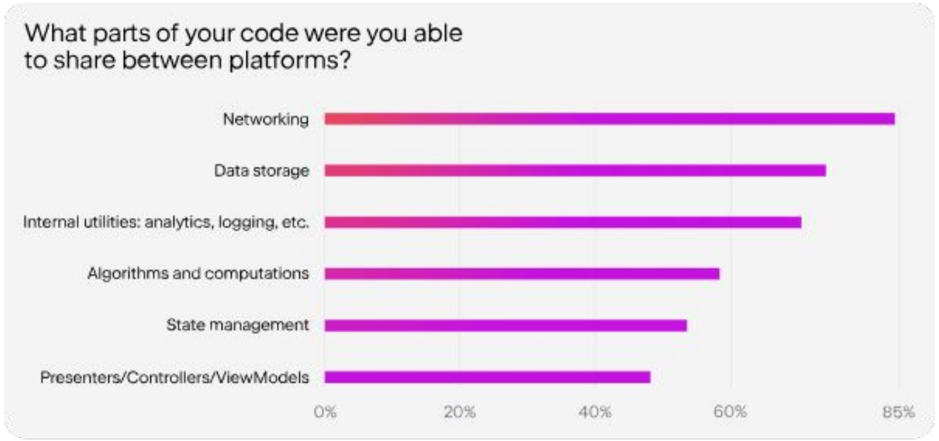
NATIVE

Cross Platform

Multiplatform


App
App
iOS Swift code
Kotlin shared code
Android Kotlin code




What parts of your code were you able to share between platforms?

Based on the results of the Kotlin Multiplatform Survey Q1-Q2 2021

bangkit

Why choose KMM ?

Pros

1. Android developers are already familiar
2. Shared business logic
3. Can share as much or as little as desired
4. Not sharing the UI can be a bane or a boon.
5. Interoperability
6. Ability to use platform-specific libraries
7. Has libraries for all major tasks

Cons

1. Limited platform support
2. Learning curve
3. Integration challenges
4. Third-party library support

How to start ?

bangkit

“As Developer”

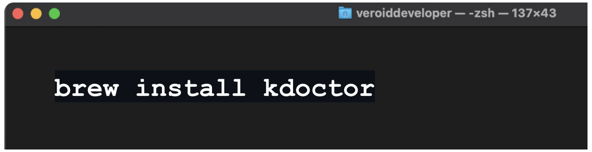
1. Knowledge of Kotlin
2. Understanding of multiplatform development
3. Familiarity with the platforms you want to target
4. Experience with build tool
5. Willingness to learn

“Requirement”

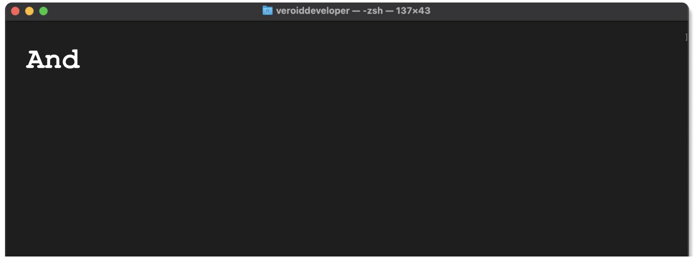
1. **Mac with macOS** 💻
2. **JDK**
3. **Android Studio** version 4.1 or above
4. **XCode** 11.3 or above

“KDoctor”

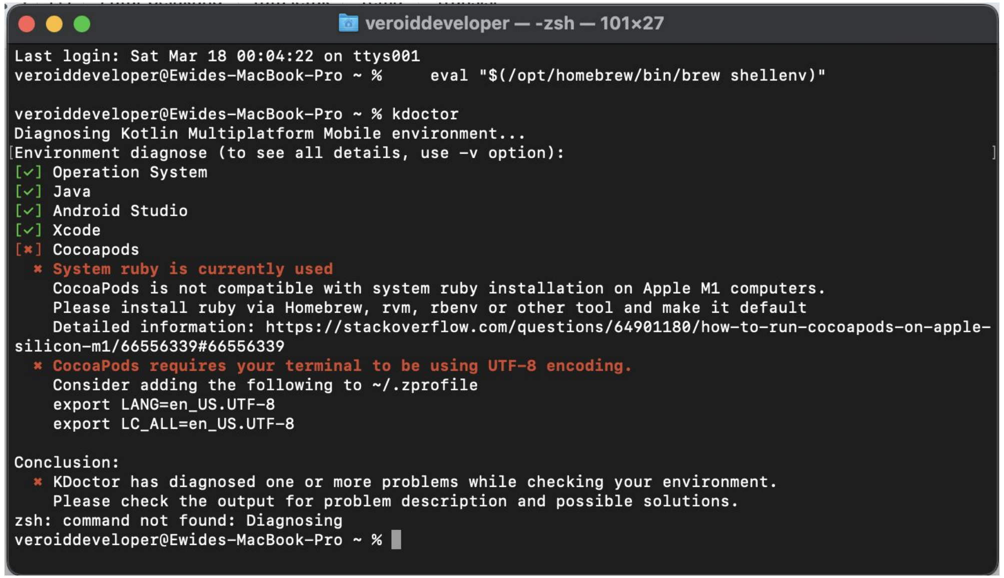
KDoctor is a command-line tool that helps to set up the environment for Kotlin Multiplatform Mobile app development.


veroiddeveloper — -zsh — 137×43

```
brew install kdoctor
```

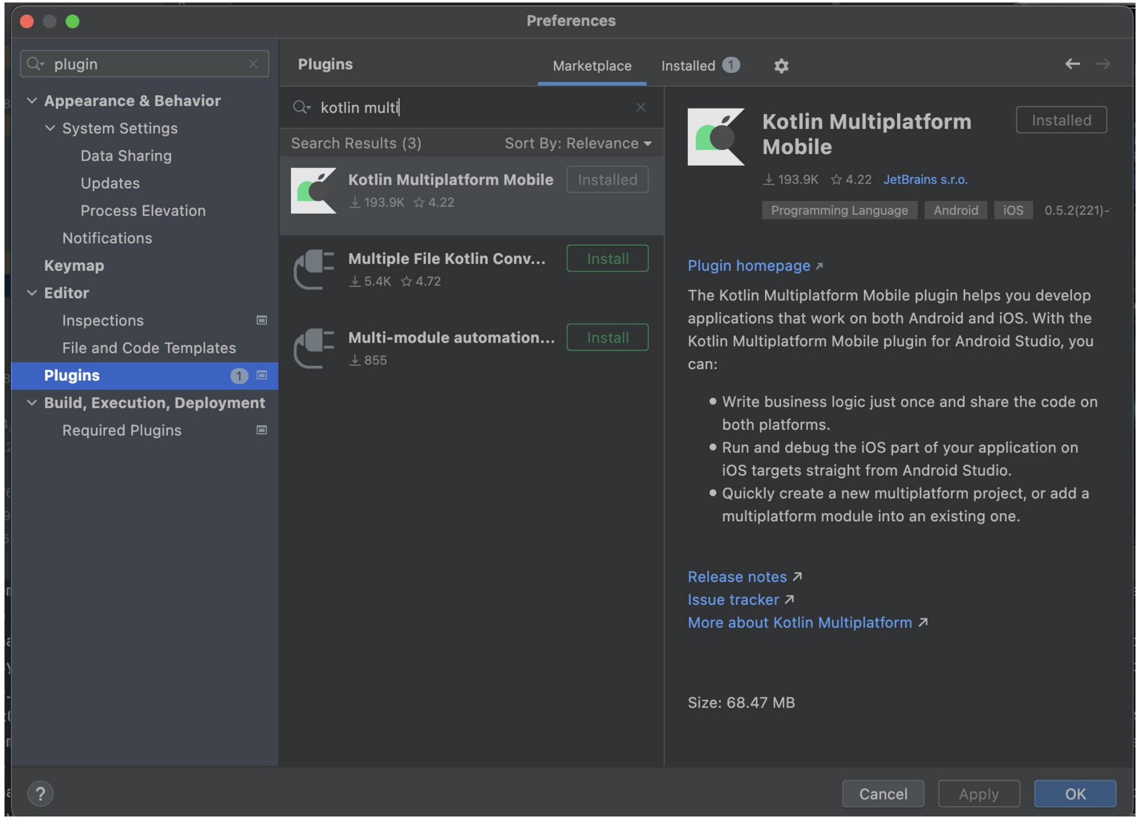

veroiddeveloper — -zsh — 137×43

```
And
```


veroiddeveloper — -zsh — 101×27

```
Last login: Sat Mar 18 00:04:22 on ttys001
veroiddeveloper@Ewides-MacBook-Pro ~ %     eval "$(/opt/homebrew/bin/brew shellenv)"

veroiddeveloper@Ewides-MacBook-Pro ~ % kdoctor
Diagnosing Kotlin Multiplatform Mobile environment...
Environment diagnose (to see all details, use -v option):
[✓] Operation System
[✓] Java
[✓] Android Studio
[✓] Xcode
[✖] Cocoapods
  ✖ System ruby is currently used
    CocoaPods is not compatible with system ruby installation on Apple M1 computers.
    Please install ruby via Homebrew, rvm, rbenv or other tool and make it default
    Detailed information: https://stackoverflow.com/questions/64901180/how-to-run-cocoapods-on-apple-
silicon-m1/66556339#66556339
  ✖ CocoaPods requires your terminal to be using UTF-8 encoding.
    Consider adding the following to ~/.zprofile
    export LANG=en_US.UTF-8
    export LC_ALL=en_US.UTF-8

Conclusion:
  ✖ KDoctor has diagnosed one or more problems while checking your environment.
    Please check the output for problem description and possible solutions.
zsh: command not found: Diagnosing
veroiddeveloper@Ewides-MacBook-Pro ~ %
```

“KMM Plugin”


Preferences

plugin

- Appearance & Behavior
  - System Settings
    - Data Sharing
    - Updates
    - Process Elevation
  - Notifications
- Keymap
- Editor
  - Inspections
  - File and Code Templates
- Plugins 1
- Build, Execution, Deployment
  - Required Plugins

Plugins | Marketplace | Installed 1

kotlin multi

Search Results (3) Sort By: Relevance

Kotlin Multiplatform Mobile — Installed — 193.9K ☆ 4.22

Multiple File Kotlin Conv... — Install — 5.4K ☆ 4.72

Multi-module automation... — Install — 855

Kotlin Multiplatform Mobile — Installed

193.9K ☆ 4.22 JetBrains s.r.o.

Programming Language | Android | iOS | 0.5.2(221)-

Plugin homepage ↗

The Kotlin Multiplatform Mobile plugin helps you develop applications that work on both Android and iOS. With the Kotlin Multiplatform Mobile plugin for Android Studio, you can:

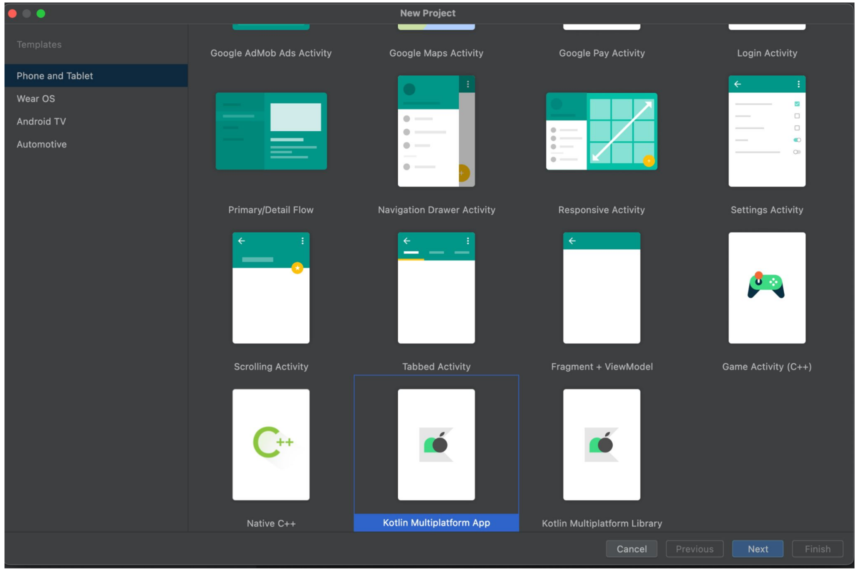
- Write business logic just once and share the code on both platforms.
- Run and debug the iOS part of your application on iOS targets straight from Android Studio.
- Quickly create a new multiplatform project, or add a multiplatform module into an existing one.

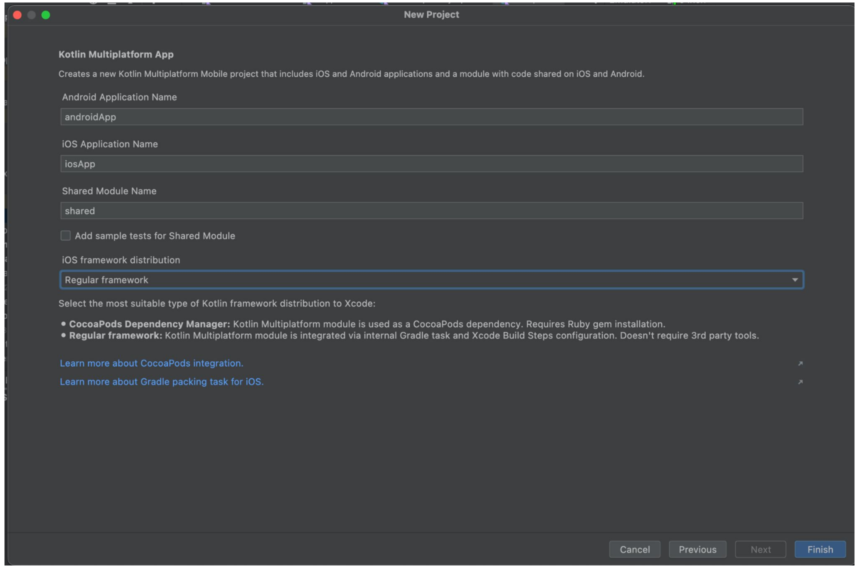
Release notes ↗
Issue tracker ↗
More about Kotlin Multiplatform ↗

Size: 68.47 MB

bangkit

“Create Project”





New Project

Kotlin Multiplatform App

Creates a new Kotlin Multiplatform Mobile project that includes iOS and Android applications and a module with code shared on iOS and Android.

Android Application Name: androidApp

iOS Application Name: iosApp

Shared Module Name: shared

☐ Add sample tests for Shared Module

iOS framework distribution: Regular framework

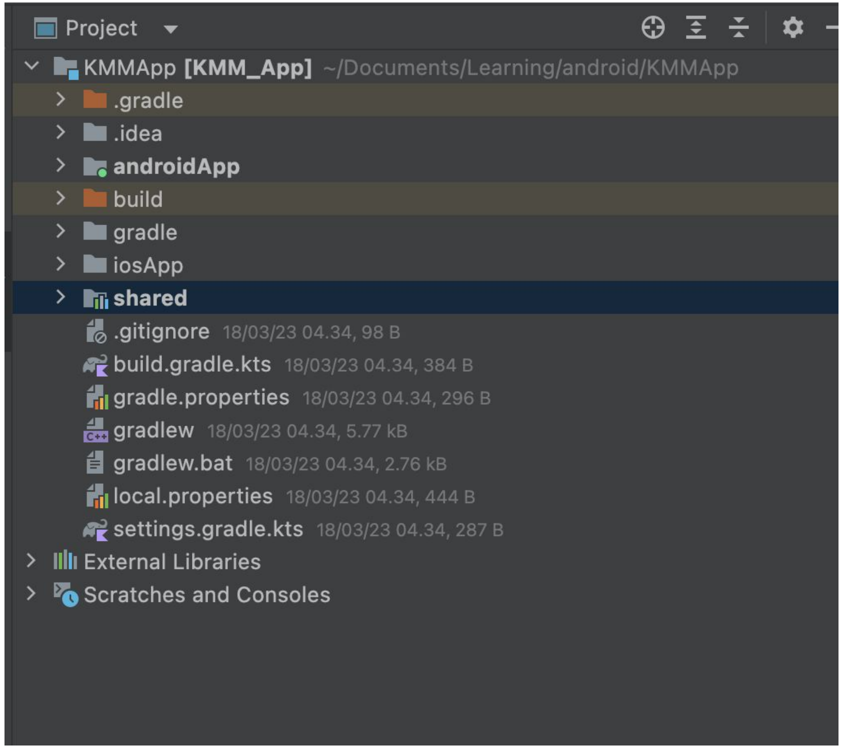
Select the most suitable type of Kotlin framework distribution to Xcode:

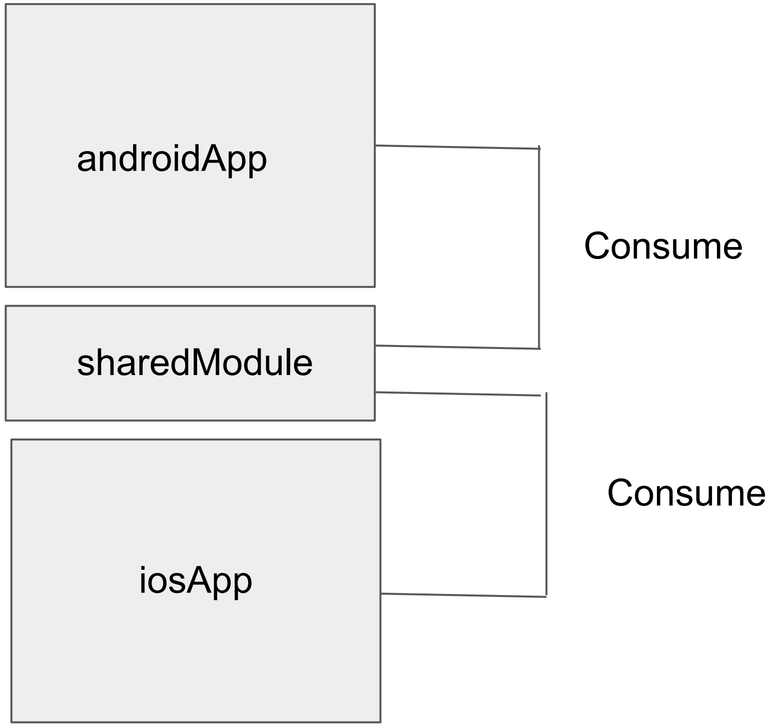
- **CocoaPods Dependency Manager:** Kotlin Multiplatform module is used as a CocoaPods dependency. Requires Ruby gem installation.
- **Regular framework:** Kotlin Multiplatform module is integrated via internal Gradle task and Xcode Build Steps configuration. Doesn't require 3rd party tools.

Learn more about CocoaPods integration.

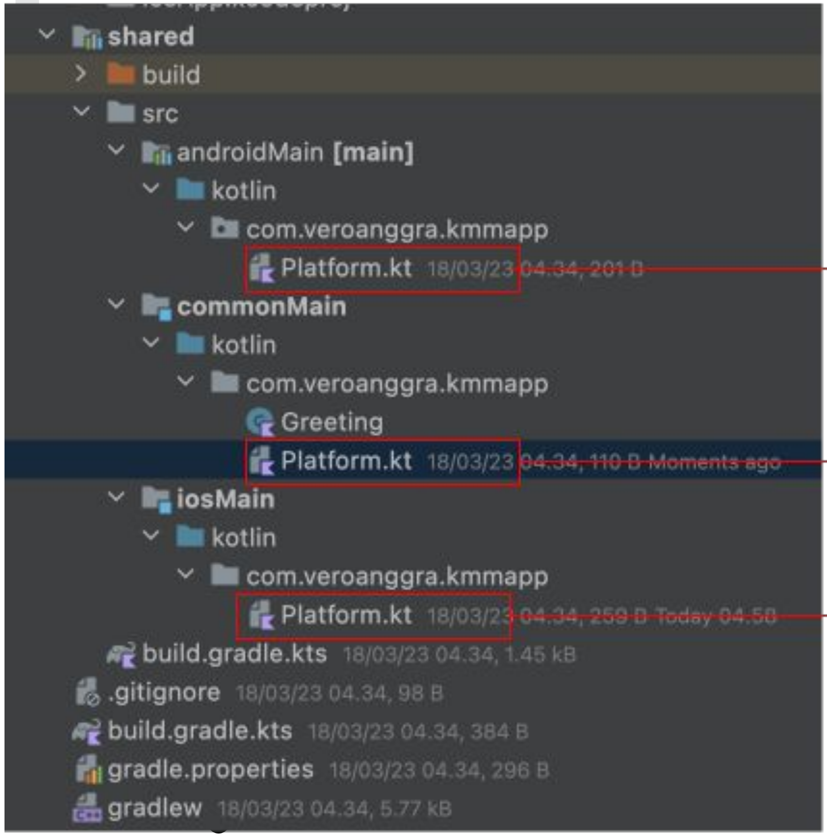
Learn more about Gradle packing task for iOS.

Project Structure


```
Project
KMMApp [KMM_App] ~/Documents/Learning/android/KMMApp
  .gradle
  .idea
  androidApp
  build
  gradle
  iosApp
  shared
  .gitignore  18/03/23 04.34, 98 B
  build.gradle.kts  18/03/23 04.34, 384 B
  gradle.properties  18/03/23 04.34, 296 B
  gradlew  18/03/23 04.34, 5.77 kB
  gradlew.bat  18/03/23 04.34, 2.76 kB
  local.properties  18/03/23 04.34, 444 B
  settings.gradle.kts  18/03/23 04.34, 287 B
External Libraries
Scratches and Consoles
```



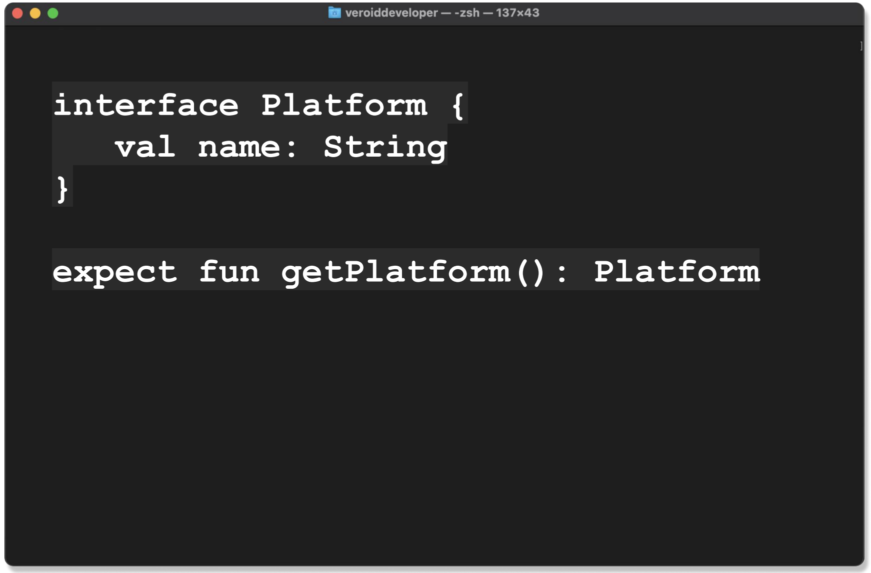
“Shared Package”



```
shared
  build
  src
    androidMain [main]
      kotlin
        com.veroanggra.kmmapp
          Platform.kt   18/03/23 04.34, 201 B
    commonMain
      kotlin
        com.veroanggra.kmmapp
          Greeting
          Platform.kt   18/03/23 04.34, 110 B Moments ago
    iosMain
      kotlin
        com.veroanggra.kmmapp
          Platform.kt   18/03/23 04.34, 259 B Today 04.58
  build.gradle.kts   18/03/23 04.34, 1.45 kB
.gitignore   18/03/23 04.34, 98 B
build.gradle.kts   18/03/23 04.34, 384 B
gradle.properties   18/03/23 04.34, 296 B
gradlew   18/03/23 04.34, 5.77 kB
```

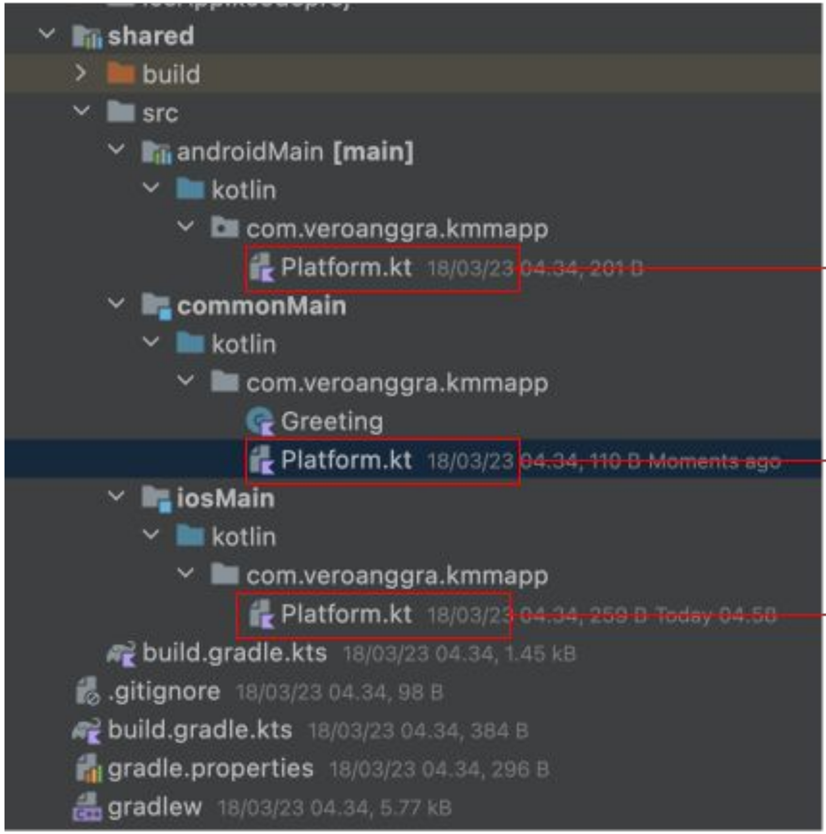
Anatomy

Common - (expect)



```kotlin
interface Platform {
    val name: String
}

expect fun getPlatform(): Platform
```

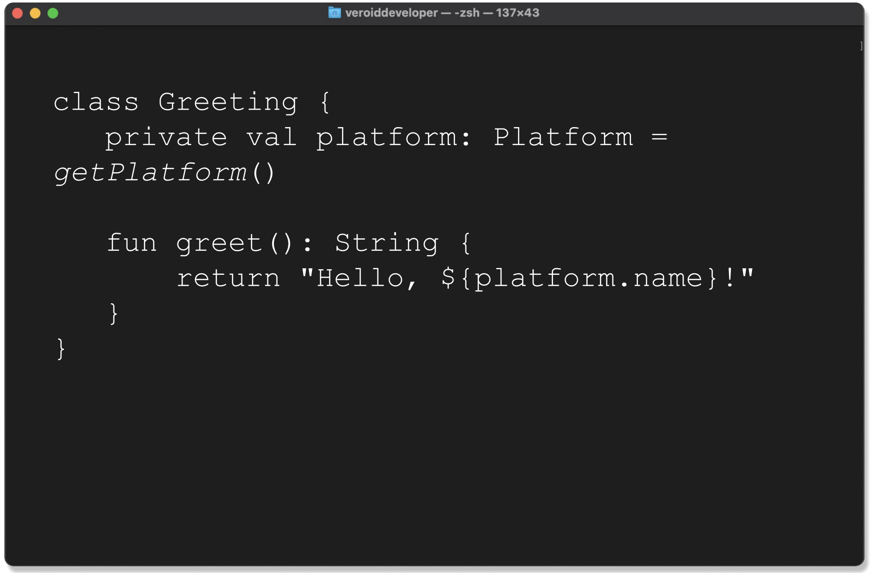
“Shared Package”



```
shared
  build
  src
    androidMain [main]
      kotlin
        com.veroanggra.kmmapp
          Platform.kt 18/03/23 04.34, 201 B
    commonMain
      kotlin
        com.veroanggra.kmmapp
          Greeting
          Platform.kt 18/03/23 04.34, 110 B Moments ago
    iosMain
      kotlin
        com.veroanggra.kmmapp
          Platform.kt 18/03/23 04.34, 259 B Today 04.58
  build.gradle.kts 18/03/23 04.34, 1.45 kB
.gitignore 18/03/23 04.34, 98 B
build.gradle.kts 18/03/23 04.34, 384 B
gradle.properties 18/03/23 04.34, 296 B
gradlew 18/03/23 04.34, 5.77 kB
```

Anatomy

Greeting Function

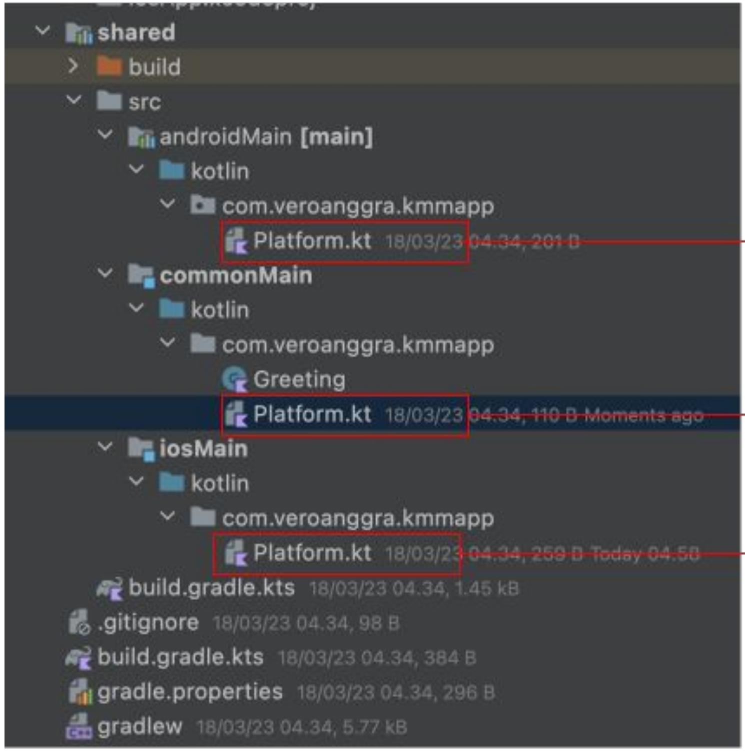


veroiddeveloper — -zsh — 137×43

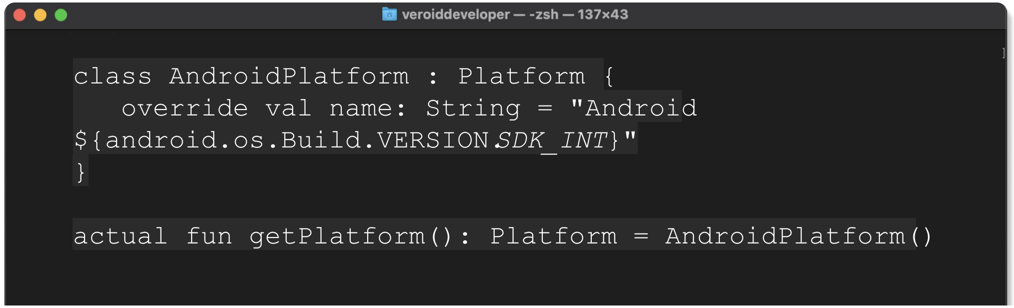
```
class Greeting {
   private val platform: Platform =
getPlatform()

   fun greet(): String {
       return "Hello, ${platform.name}!"
   }
}
```

“Shared Package”

Anatomy

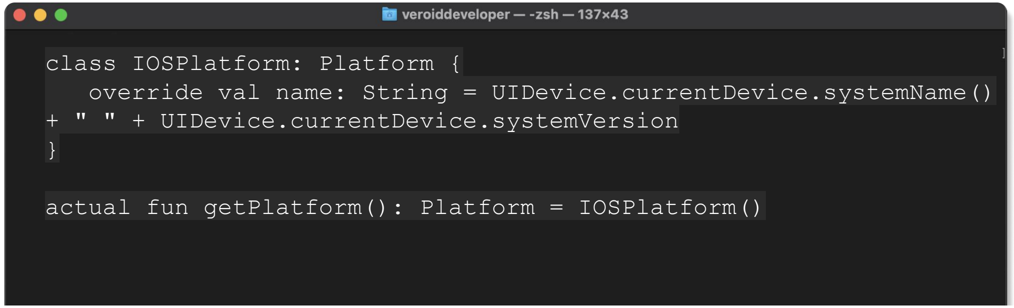

shared
build
src
androidMain [main]
kotlin
com.veroanggra.kmmapp
Platform.kt 18/03/23 04.34, 201 B
commonMain
kotlin
com.veroanggra.kmmapp
Greeting
Platform.kt 18/03/23 04.34, 110 B Moments ago
iosMain
kotlin
com.veroanggra.kmmapp
Platform.kt 18/03/23 04.34, 259 B Today 04.58
build.gradle.kts 18/03/23 04.34, 1.45 kB
.gitignore 18/03/23 04.34, 98 B
build.gradle.kts 18/03/23 04.34, 384 B
gradle.properties 18/03/23 04.34, 296 B
gradlew 18/03/23 04.34, 5.77 kB

Android - (actual)



```
class AndroidPlatform : Platform {
   override val name: String = "Android
${android.os.Build.VERSION.SDK_INT}"
}

actual fun getPlatform(): Platform = AndroidPlatform()
```

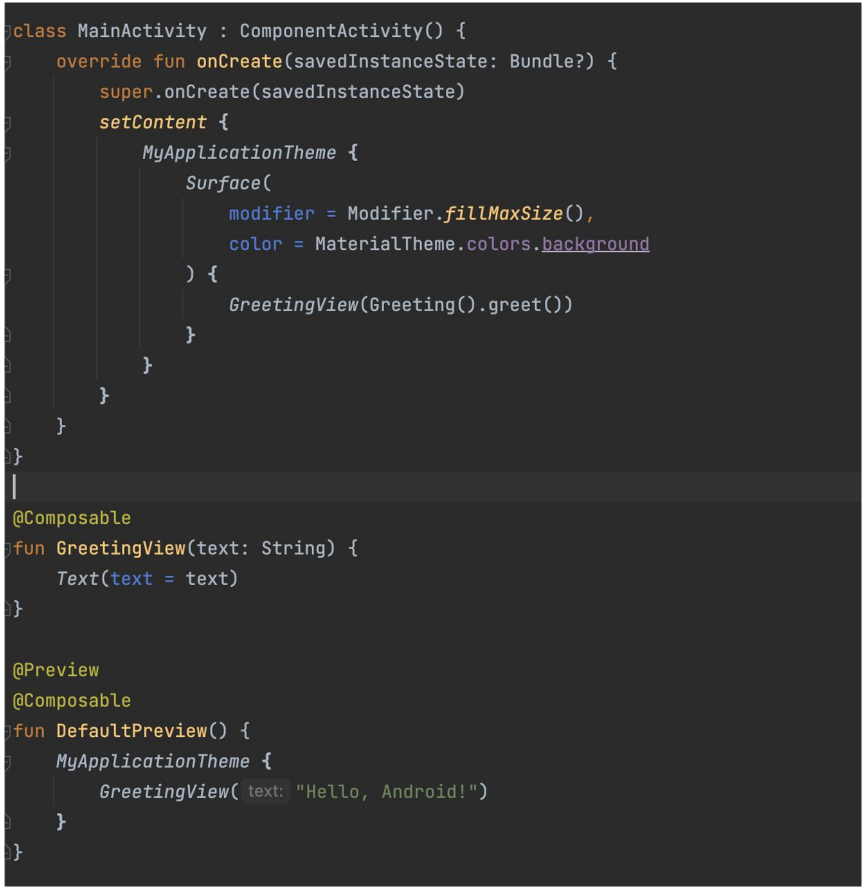
IOS - (actual)



```
class IOSPlatform: Platform {
   override val name: String = UIDevice.currentDevice.systemName()
+ " " + UIDevice.currentDevice.systemVersion
}

actual fun getPlatform(): Platform = IOSPlatform()
```

bangkit

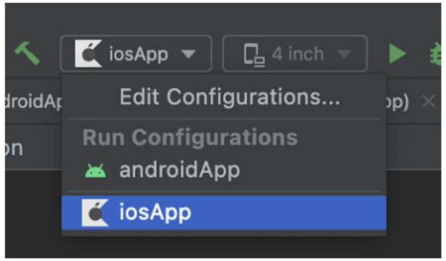
“Android Package”

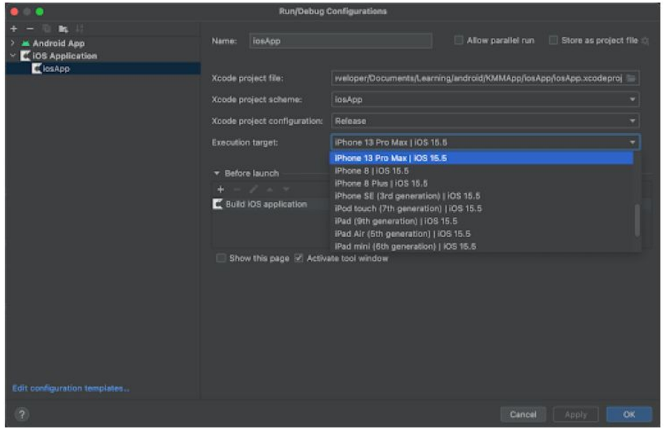


```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MyApplicationTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    GreetingView(Greeting().greet())
                }
            }
        }
    }
}

@Composable
fun GreetingView(text: String) {
    Text(text = text)
}

@Preview
@Composable
fun DefaultPreview() {
    MyApplicationTheme {
        GreetingView( text: "Hello, Android!")
    }
}
```

bangk!t

“Compile”





Name: iosApp
[ ] Allow parallel run
[ ] Store as project file

Xcode project file: veloper/Documents/Learning/android/KMMApp/iosApp/iosApp.xcodeproj
Xcode project scheme: iosApp
Xcode project configuration: Release
Execution target: iPhone 13 Pro Max | iOS 15.5

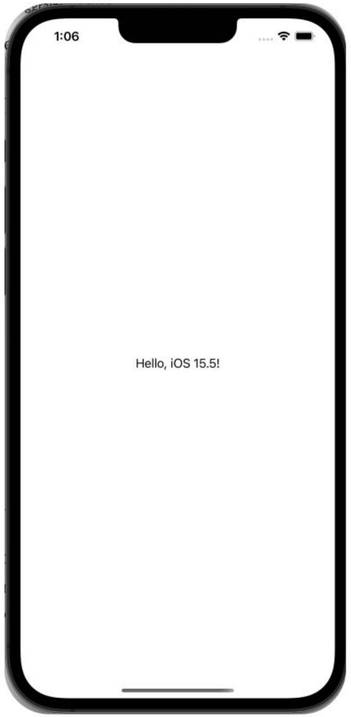
- iPhone 13 Pro Max | iOS 15.5
- iPhone 8 | iOS 15.5
- iPhone 8 Plus | iOS 15.5
- iPhone SE (3rd generation) | iOS 15.5
- iPod touch (7th generation) | iOS 15.5
- iPad (9th generation) | iOS 15.5
- iPad Air (5th generation) | iOS 15.5
- iPad mini (6th generation) | iOS 15.5

Before launch
Build iOS application

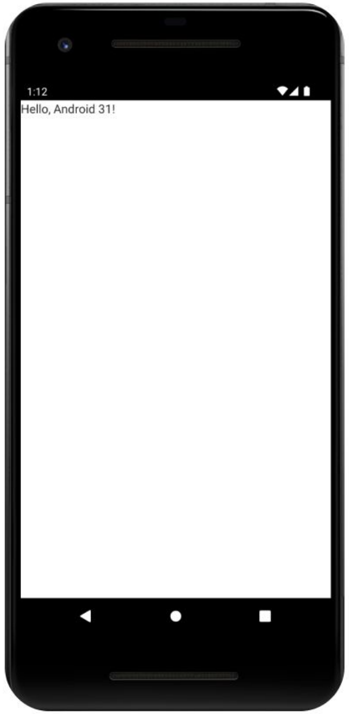
[ ] Show this page [✓] Activate tool window

bangkit

Result


1:06

Hello, iOS 15.5!


1:12

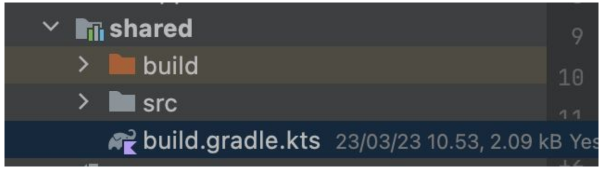
Hello, Android 31!

How to setup the gradle file?

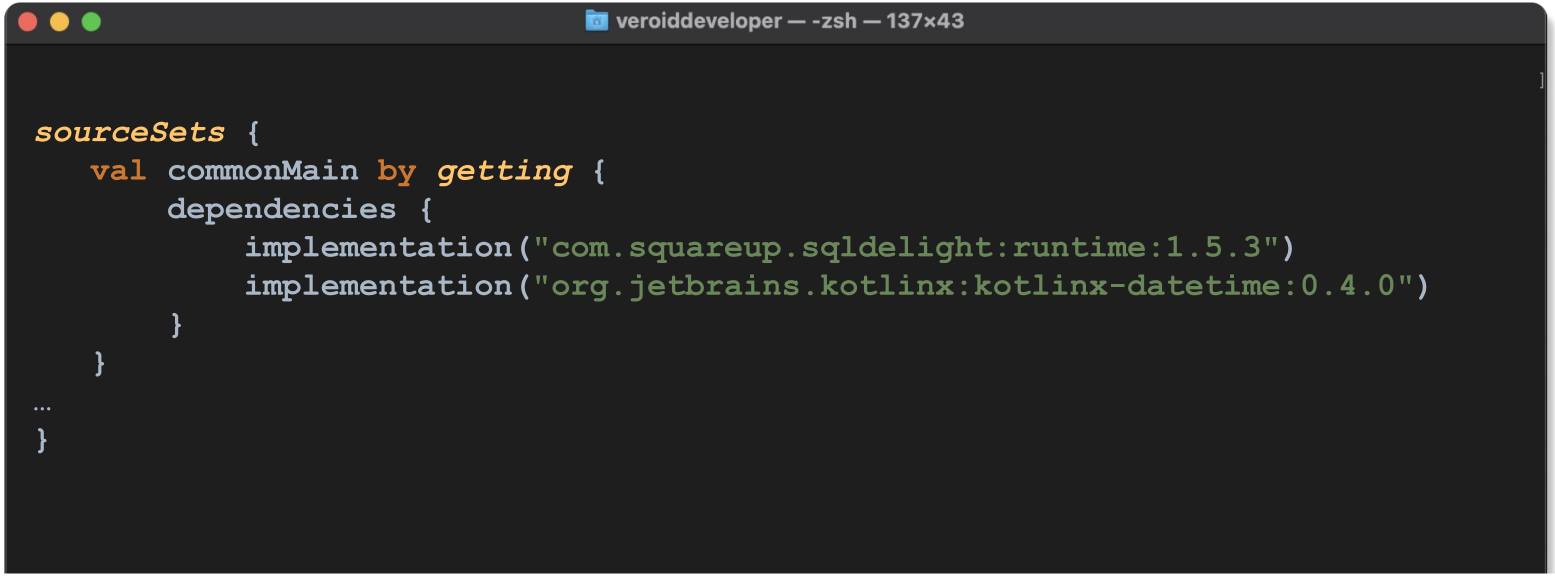
Simple Case

Build Simple Note App Using SQL Delight

Add the Library for Common



shared
build
src
build.gradle.kts 23/03/23 10.53, 2.09 kB Yes

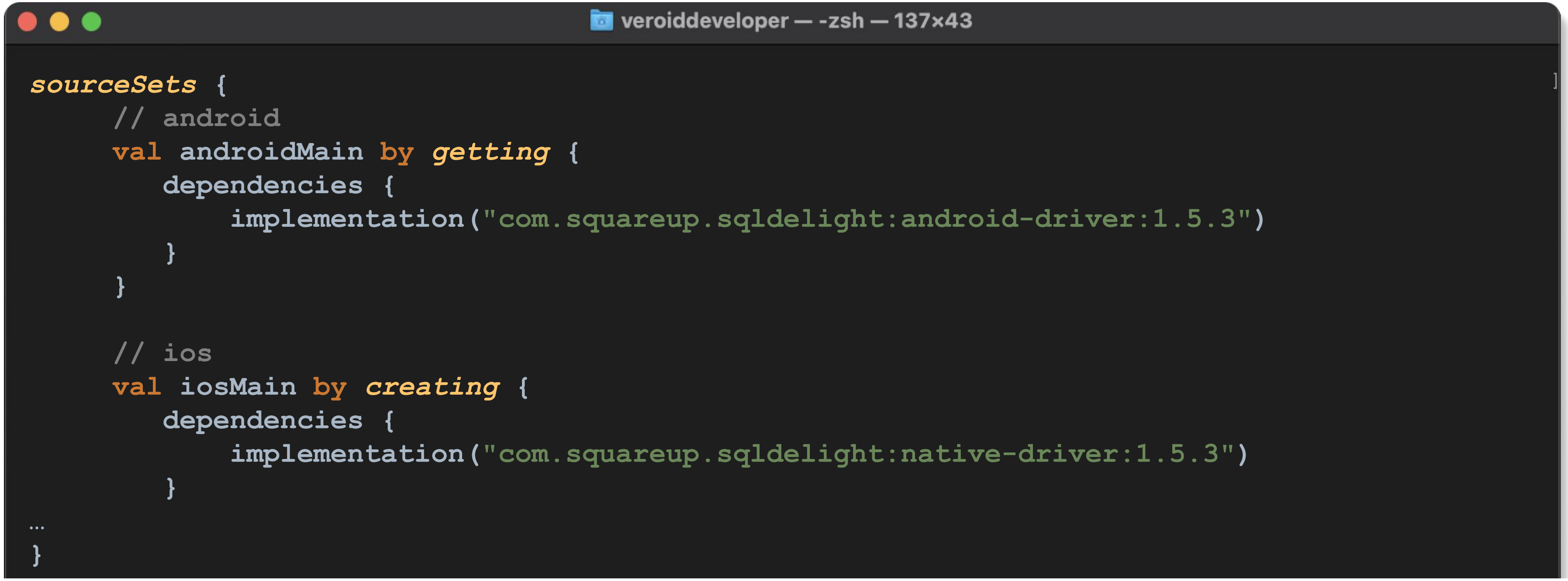


veroiddeveloper — -zsh — 137×43

```
sourceSets {
    val commonMain by getting {
        dependencies {
            implementation("com.squareup.sqldelight:runtime:1.5.3")
            implementation("org.jetbrains.kotlinx:kotlinx-datetime:0.4.0")
        }
    }
...
}
```

bangkit

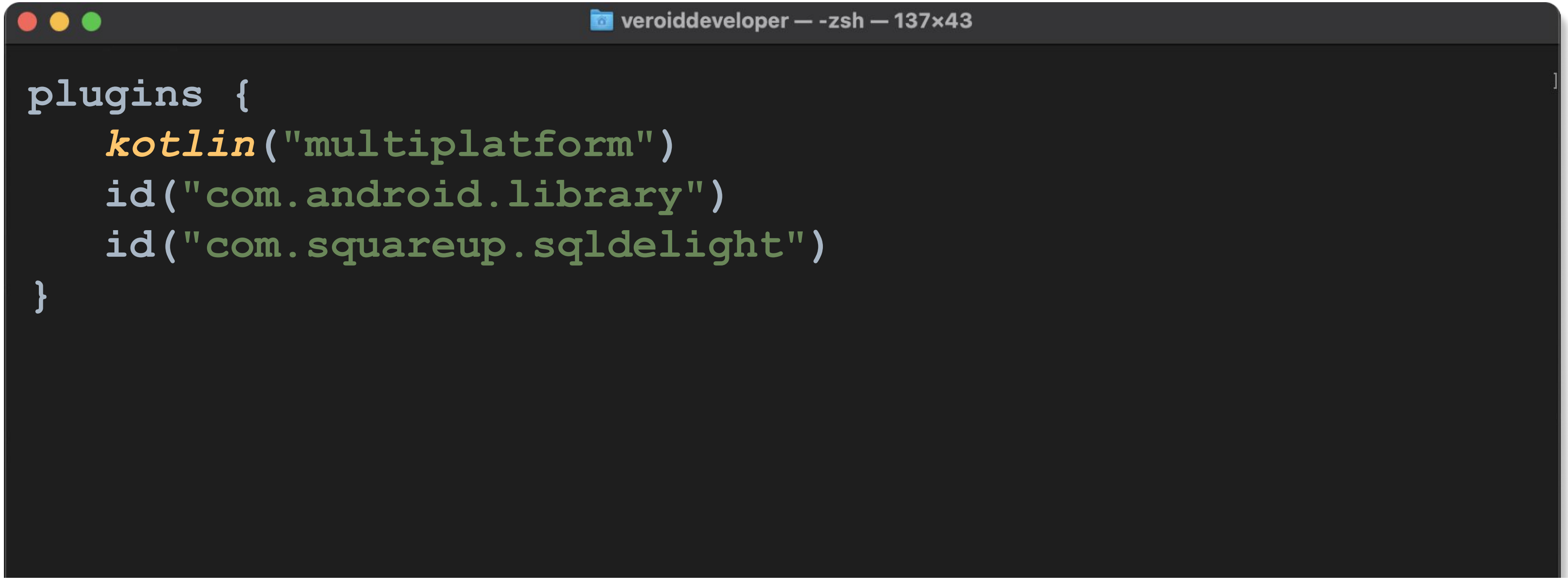
“Add the on Specific Platform”


veroiddeveloper — -zsh — 137×43

```
sourceSets {
        // android
        val androidMain by getting {
            dependencies {
                implementation("com.squareup.sqldelight:android-driver:1.5.3")
            }
        }

        // ios
        val iosMain by creating {
            dependencies {
                implementation("com.squareup.sqldelight:native-driver:1.5.3")
            }
…
}
```

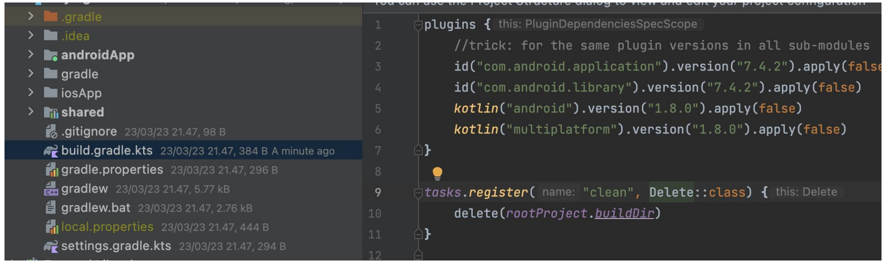
bangkit

Add the SQLDelight Plugin



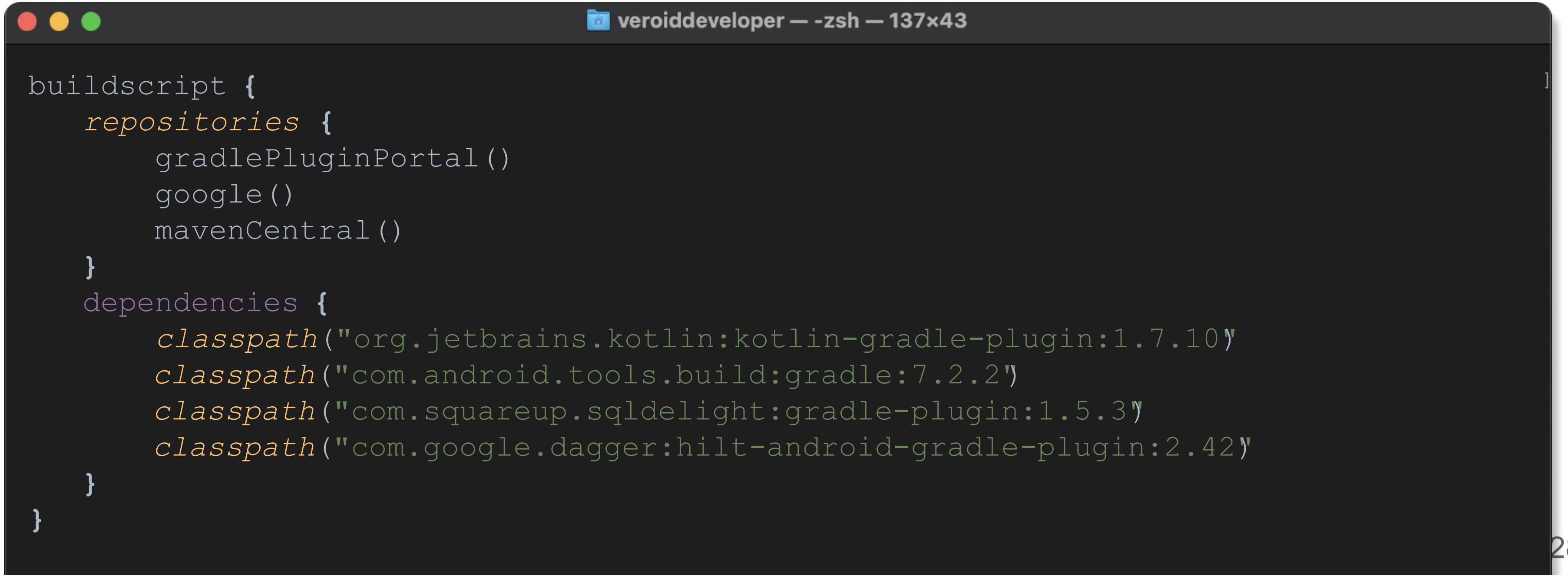
```
plugins {
    kotlin("multiplatform")
    id("com.android.library")
    id("com.squareup.sqldelight")
}
```

Setup Root Gradle File



```
.gradle
.idea
androidApp
gradle
iosApp
shared
.gitignore  23/03/23 21.47, 98 B
build.gradle.kts  23/03/23 21.47, 384 B A minute ago
gradle.properties  23/03/23 21.47, 296 B
gradlew  23/03/23 21.47, 5.77 kB
gradlew.bat  23/03/23 21.47, 2.76 kB
local.properties  23/03/23 21.47, 444 B
settings.gradle.kts  23/03/23 21.47, 294 B
```

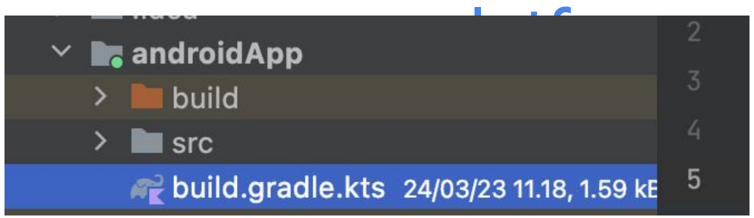
```
plugins { this: PluginDependenciesSpecScope
    //trick: for the same plugin versions in all sub-modules
    id("com.android.application").version("7.4.2").apply(false
    id("com.android.library").version("7.4.2").apply(false)
    kotlin("android").version("1.8.0").apply(false)
    kotlin("multiplatform").version("1.8.0").apply(false)
}

tasks.register( name: "clean", Delete::class) { this: Delete
    delete(rootProject.buildDir)
}
```



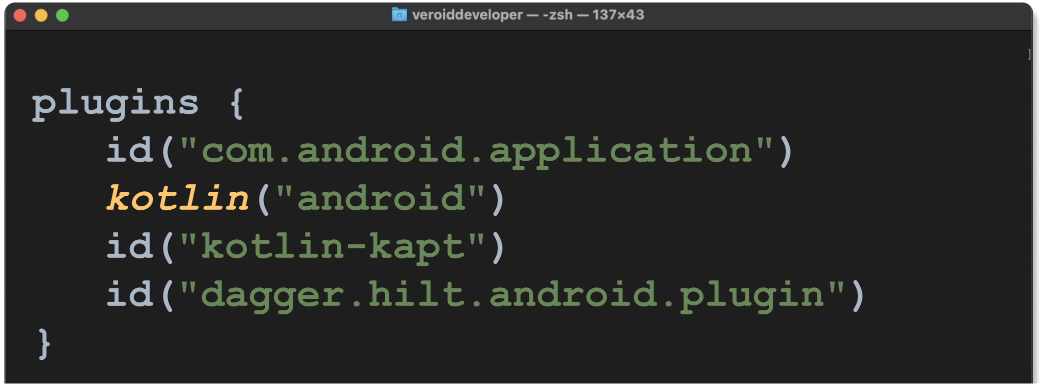
veroiddeveloper — -zsh — 137×43

```
buildscript {
    repositories {
        gradlePluginPortal()
        google()
        mavenCentral()
    }
    dependencies {
        classpath("org.jetbrains.kotlin:kotlin-gradle-plugin:1.7.10")
        classpath("com.android.tools.build:gradle:7.2.2")
        classpath("com.squareup.sqldelight:gradle-plugin:1.5.3")
        classpath("com.google.dagger:hilt-android-gradle-plugin:2.42")
    }
}
```

“Setup plugin on gradle android ”



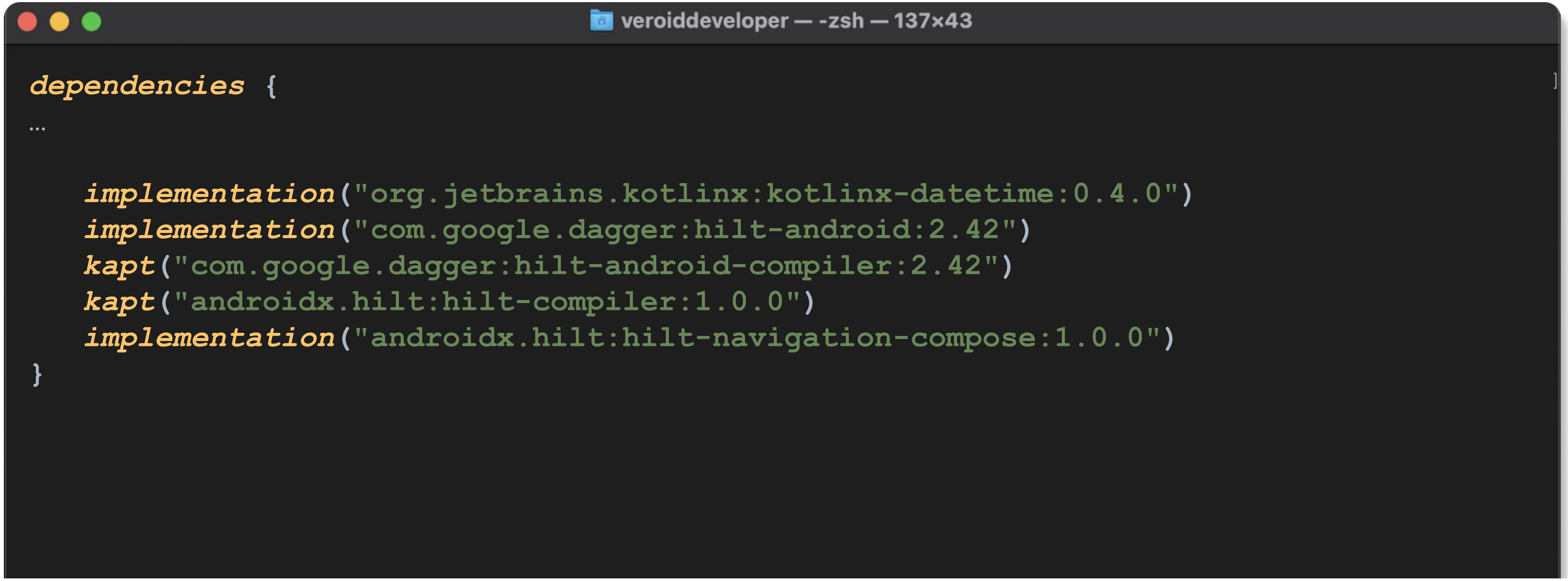
androidApp
build
src
build.gradle.kts 24/03/23 11.18, 1.59 kB



veroiddeveloper — -zsh — 137×43

```
plugins {
    id("com.android.application")
    kotlin("android")
    id("kotlin-kapt")
    id("dagger.hilt.android.plugin")
}
```

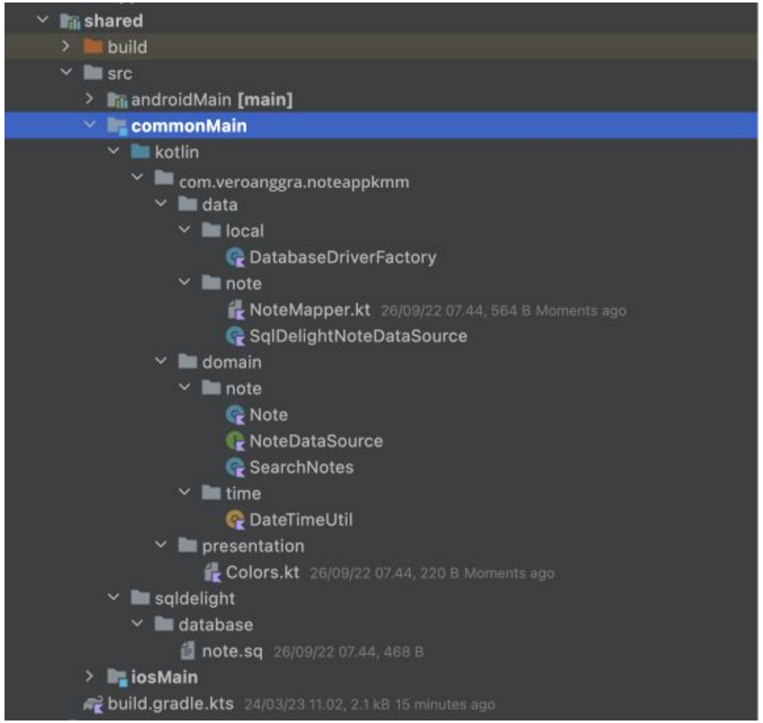
“Setup dependencies on gradle android platform”



```
dependencies {
...

    implementation("org.jetbrains.kotlinx:kotlinx-datetime:0.4.0")
    implementation("com.google.dagger:hilt-android:2.42")
    kapt("com.google.dagger:hilt-android-compiler:2.42")
    kapt("androidx.hilt:hilt-compiler:1.0.0")
    implementation("androidx.hilt:hilt-navigation-compose:1.0.0")
}
```

bangkit

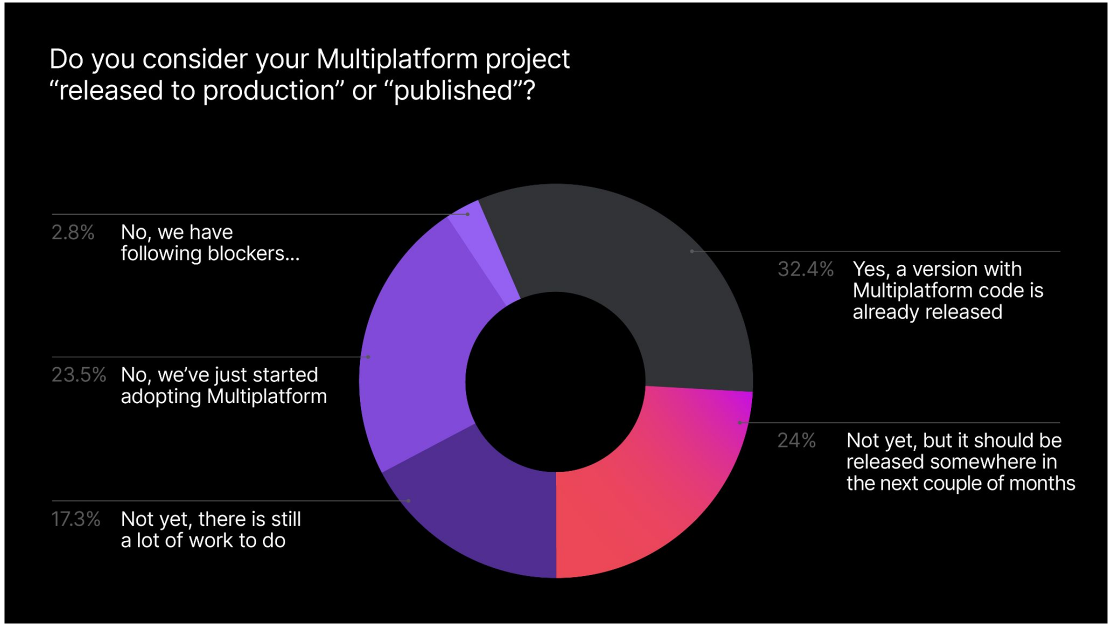
Now Your project ready to use



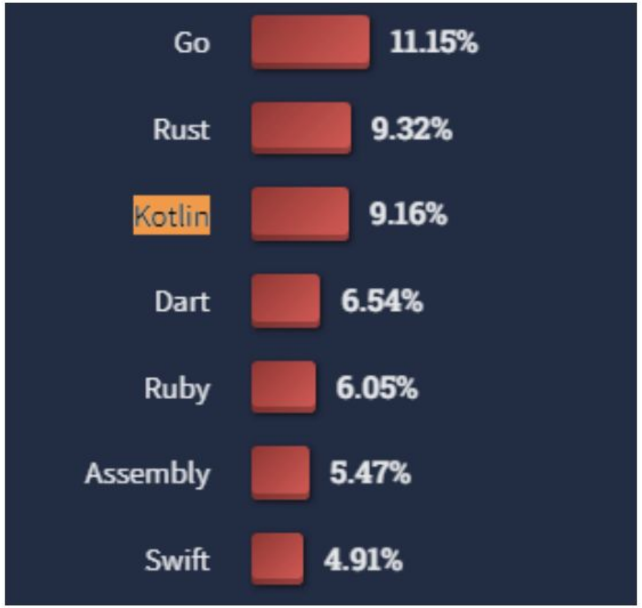
```
shared
  build
  src
    androidMain [main]
    commonMain
      kotlin
        com.veroanggra.noteappkmm
          data
            local
              DatabaseDriverFactory
            note
              NoteMapper.kt  26/09/22 07.44, 564 B Moments ago
              SqlDelightNoteDataSource
          domain
            note
              Note
              NoteDataSource
              SearchNotes
            time
              DateTimeUtil
          presentation
            Colors.kt  26/09/22 07.44, 220 B Moments ago
      sqldelight
        database
          note.sq  26/09/22 07.44, 468 B
    iosMain
  build.gradle.kts  24/03/23 11.02, 2.1 kB 15 minutes ago
```

bangkit

Now in Beta



Do you consider your Multiplatform project "released to production" or "published"?
2.8% No, we have following blockers...
23.5% No, we've just started adopting Multiplatform
17.3% Not yet, there is still a lot of work to do
32.4% Yes, a version with Multiplatform code is already released
24% Not yet, but it should be released somewhere in the next couple of months



Go 11.15%
Rust 9.32%
Kotlin 9.16%
Dart 6.54%
Ruby 6.05%
Assembly 5.47%
Swift 4.91%

Who implemented KMM

https://kotlinlang.org/lp/mobile/case-studies/

NETFLIX

Quizlet

Bai du 百度

TOUCHLAB

LEROY MERLIN

vmware®

ICEROCK

chalk

9GAG

Cash App

PHILIPS

careem

fastwork

AUTODESK.

Magic Technology

MIREGO

Unflow

netvirta

bangk!t

down dog

“Kotlin Multiplatform Mobile (KMM) is a promising technology that is changing the way mobile apps are developed. With KMM, developers can share business logic across multiple platforms, resulting in faster development times, increased code consistency, and reduced duplication of efforts.”

Veronica Anggraini

veroanggra

veroanggra

veroanggra

Thank You