



# Mobile Application Development

---

Produced by Dave Drohan ([david.drohan@setu.ie](mailto:david.drohan@setu.ie))  
Department of Computing & Mathematics  
South East Technological University  
Waterford, Ireland

Updated & Delivered by Gongzhe Qiao ([003969@nuist.edu.cn](mailto:003969@nuist.edu.cn))  
Department of Computer Science  
Nanjing University of Information Science and Technology  
Nanjing, China

[nuist.edu.cn](http://nuist.edu.cn)



[setu.ie](http://setu.ie)



# Introducing Kotlin

---

Overview - Part 3



# Agenda

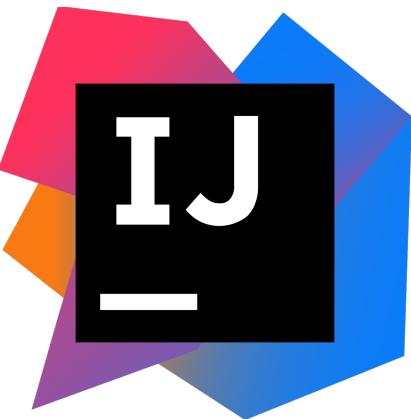
---

- ❑ What is Kotlin?
- ❑ Context - Java Vs JVM
- ❑ Kotlin History
- ❑ Milestones
- ❑ General Overview & Features
- ❑ Kotlin & IntelliJ IDEA
- ❑ Conclusion

# Agenda

---

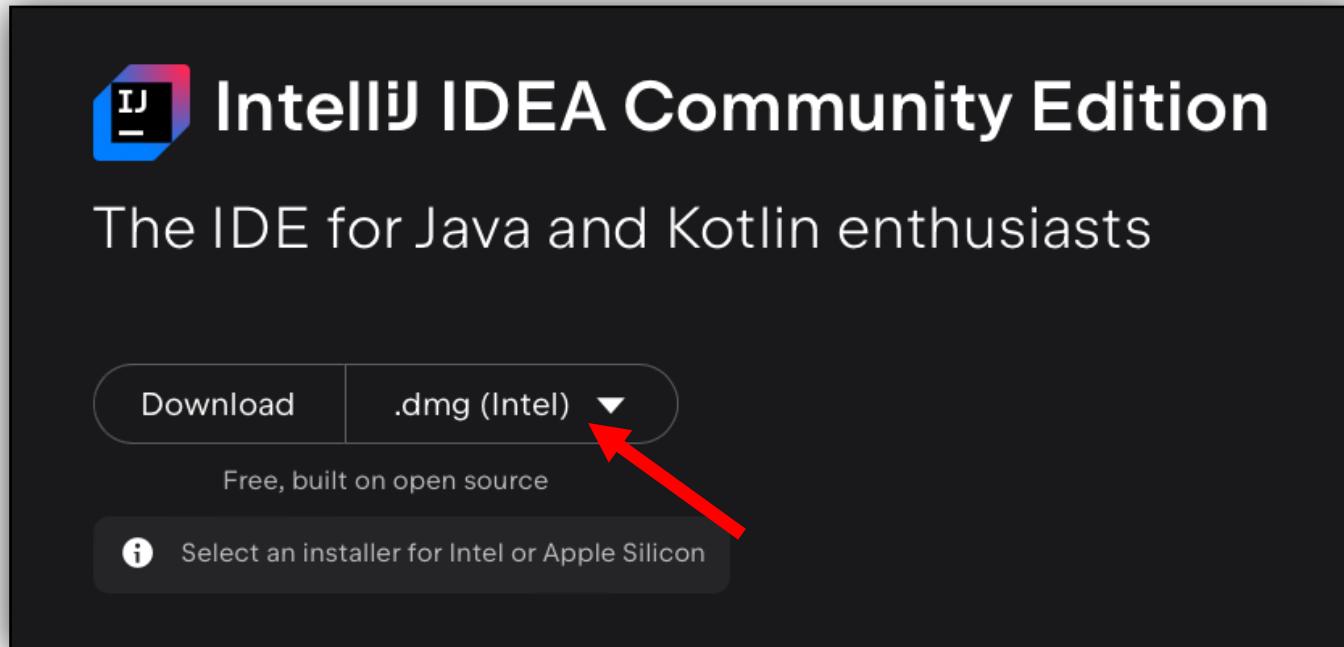
- ❑ What is Kotlin?
- ❑ Context - Java Vs JVM
- ❑ Kotlin History
- ❑ Milestones
- ❑ General Overview & Features
- ❑ **Kotlin & IntelliJ IDEA**
- ❑ Conclusion

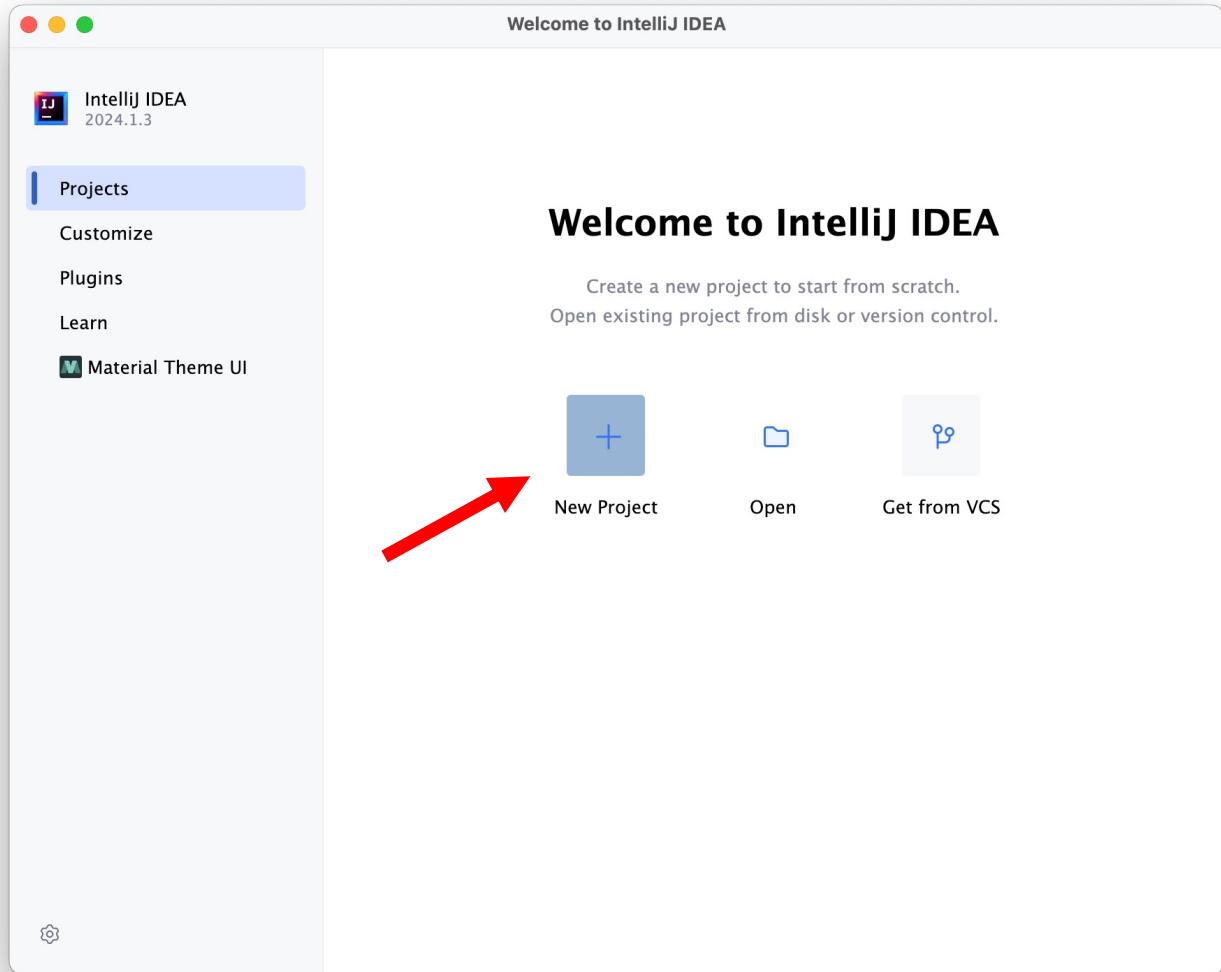


Refer to the Practical Labs for the  
current / latest version of the IDE  
we will be using

**Can be Downloaded From**  
**(<https://www.jetbrains.com/idea/download>)**

# IntelliJ Community Edition Version X.Y.Z





# New Project

# New Project

New Project

Name:  ←

Location:  █  
Project will be created in: ~/Library/CloudStorage/Dropbox/\_kotlinlabs/Types  
 Create Git repository

Build system: IntelliJ Maven Gradle ←

JDK: 21 Oracle OpenJDK 21.0.1 - aarch64 ←

Gradle DSL: Kotlin Groovy

Add sample code  
 Generate code with onboarding tips ←

To create a Kotlin Multiplatform project, [click here](#)

More via plugins... ←

? CANCEL CREATE

# New Project

New Project

Name:

Location:

Project will be created in: ~/Library/CloudStorage/Dropbox/\_kotlinlabs/Types

Create Git repository

Build system: IntelliJ  Maven  Gradle

JDK:

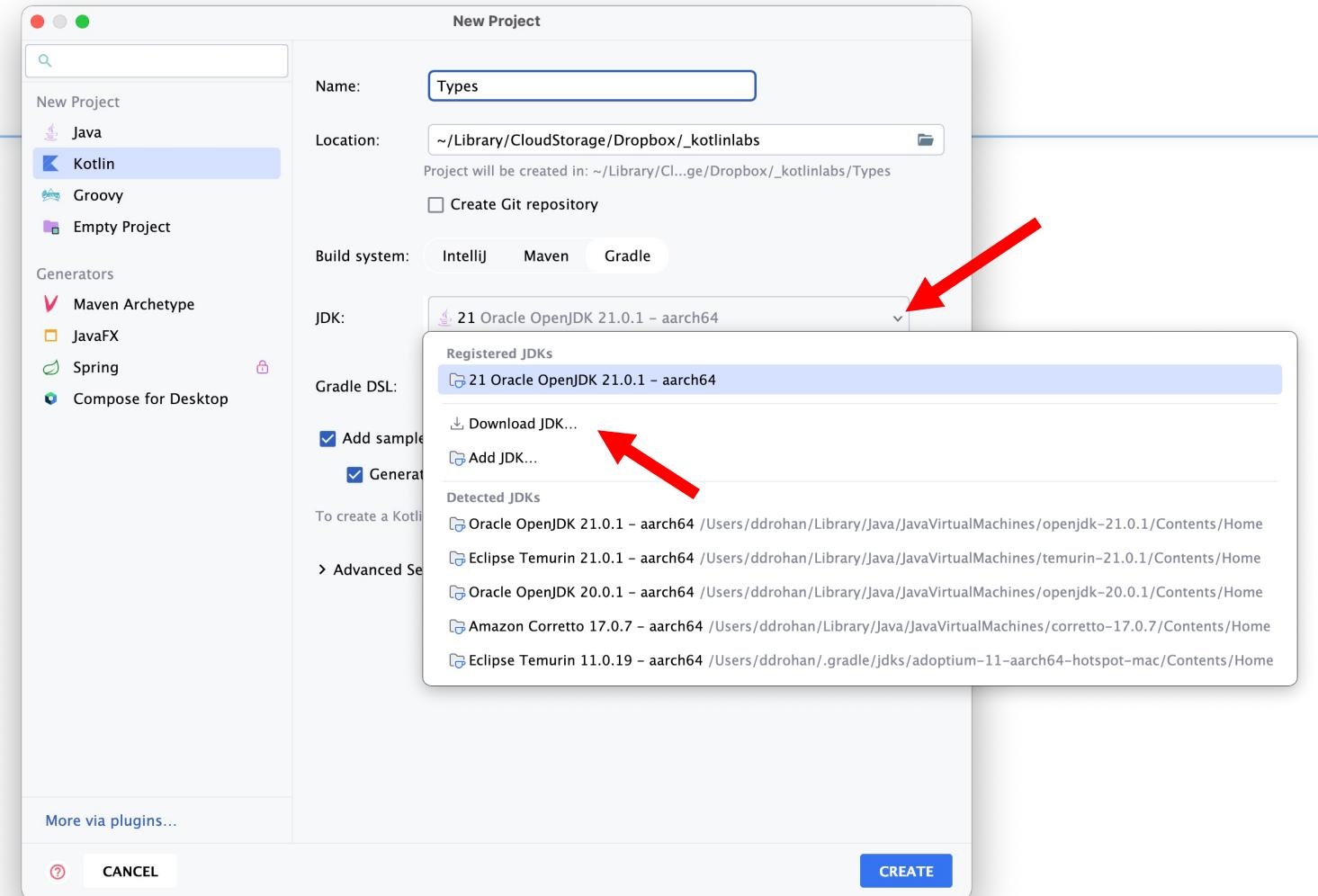
Registered JDks

Add samples  Generat

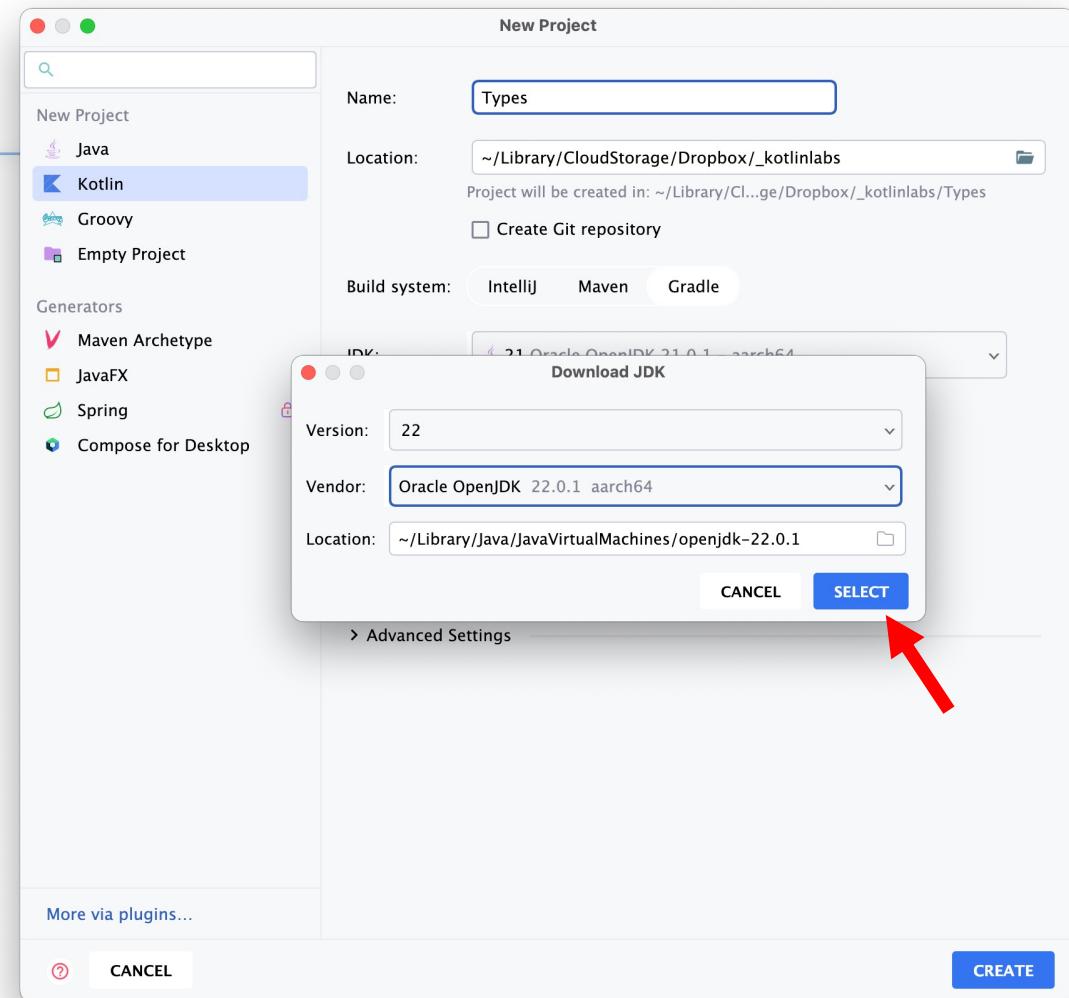
To create a Kotli

> Advanced Se

More via plugins...



# New Project



# New Project

New Project

Name:

Location:

Project will be created in: ~/Library/CloudStorage/Dropbox/\_kotlinlabs/Types

Create Git repository

Build system:  IntelliJ  Maven  Gradle

JDK:

Gradle DSL:  Kotlin  Groovy

Add sample code

Generate code with onboarding tips

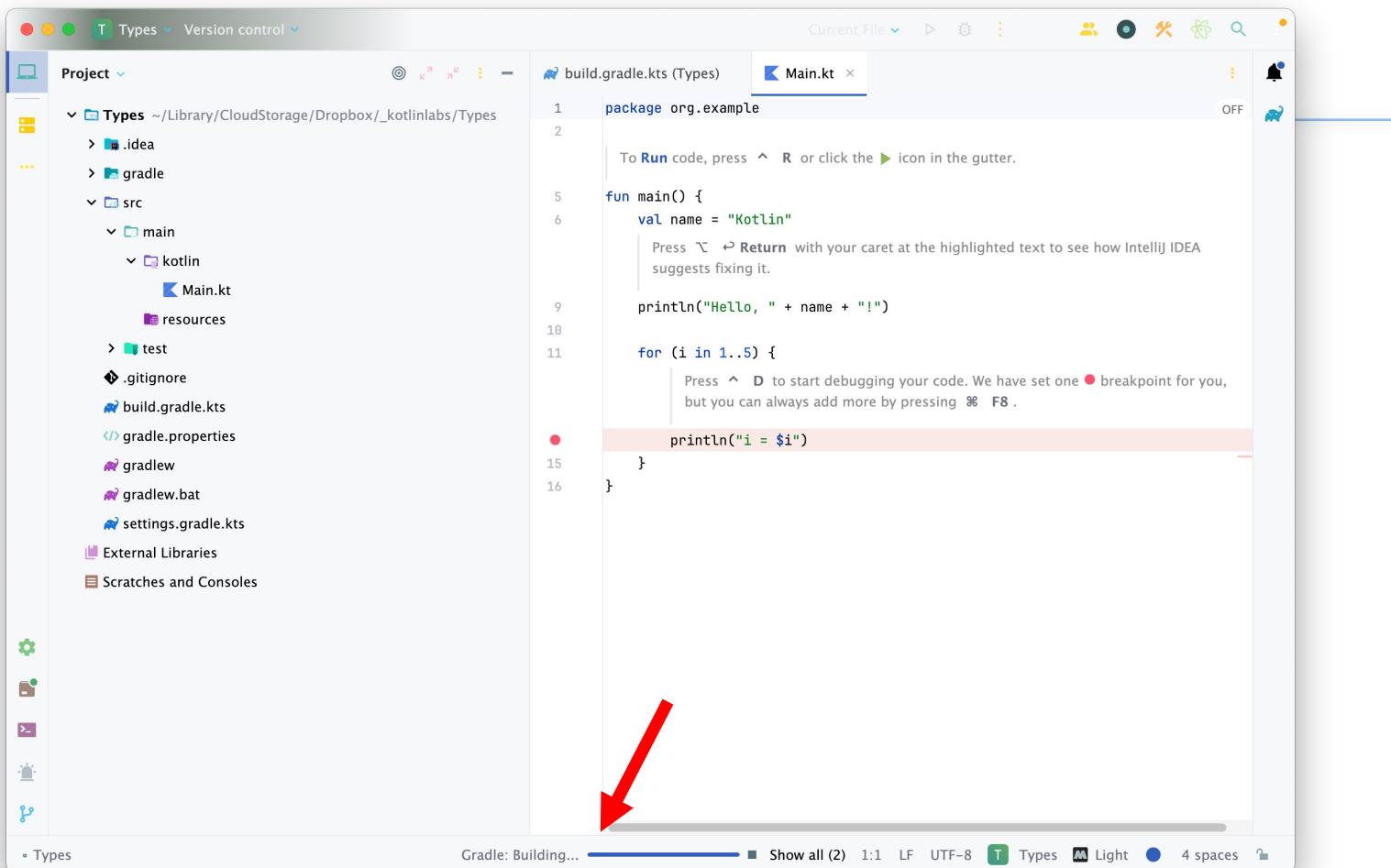
To create a Kotlin Multiplatform project, [click here](#)

> Advanced Settings

More via plugins...



# New Project



The screenshot shows the IntelliJ IDEA interface with a new project named "Types". The project structure on the left includes files like .idea, gradle, build.gradle.kts, and Main.kt. The Main.kt file is open in the editor, displaying the following code:

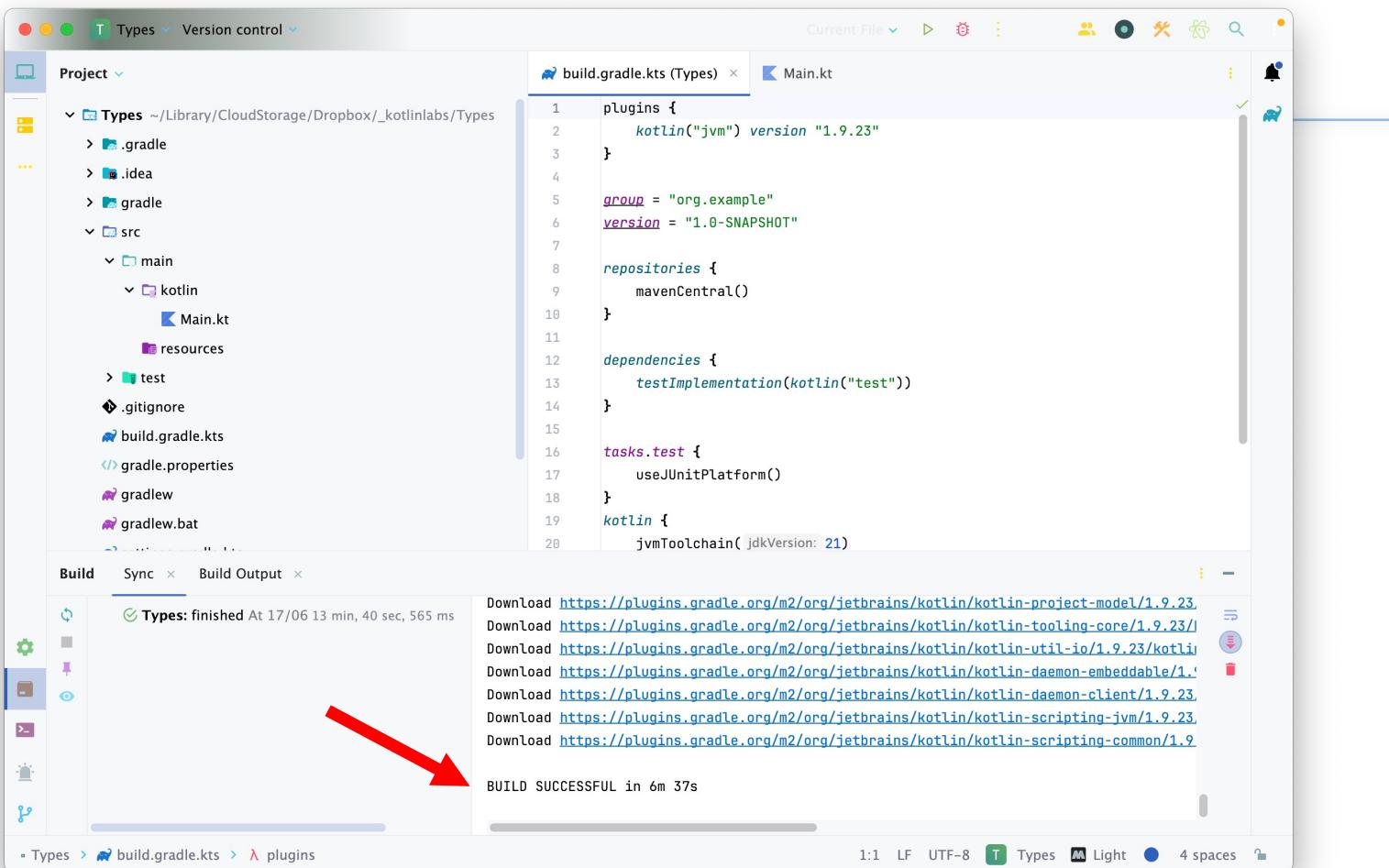
```

1 package org.example
2
3 To Run code, press ⌘ R or click the ▶ icon in the gutter.
4
5 fun main() {
6     val name = "Kotlin"
7     Press ⌘ ← Return with your caret at the highlighted text to see how IntelliJ IDEA
8     suggests fixing it.
9
10    println("Hello, " + name + "!")
11
12    for (i in 1..5) {
13        Press ⌘ D to start debugging your code. We have set one breakpoint for you,
14        but you can always add more by pressing ⌘ F8 .
15        println("i = $i")
16    }

```

The status bar at the bottom left shows "Gradle: Building..." with a progress bar. A red arrow points from the text above to this status bar.

# New Project



The screenshot shows the IntelliJ IDEA interface with a new project named "Types". The project structure on the left includes files like build.gradle.kts, Main.kt, and resources. The main editor window displays build.gradle.kts with code for a Kotlin JVM plugin. The bottom right pane shows the build output with successful plugin downloads and a build time of 6m 37s.

Project: Types

build.gradle.kts (Types) x Main.kt

```
plugins {
    kotlin("jvm") version "1.9.23"
}

group = "org.example"
version = "1.0-SNAPSHOT"

repositories {
    mavenCentral()
}

dependencies {
    testImplementation(kotlin("test"))
}

tasks.test {
    useJUnitPlatform()
}

kotlin {
    jvmToolchain( jdkVersion: 21)
```

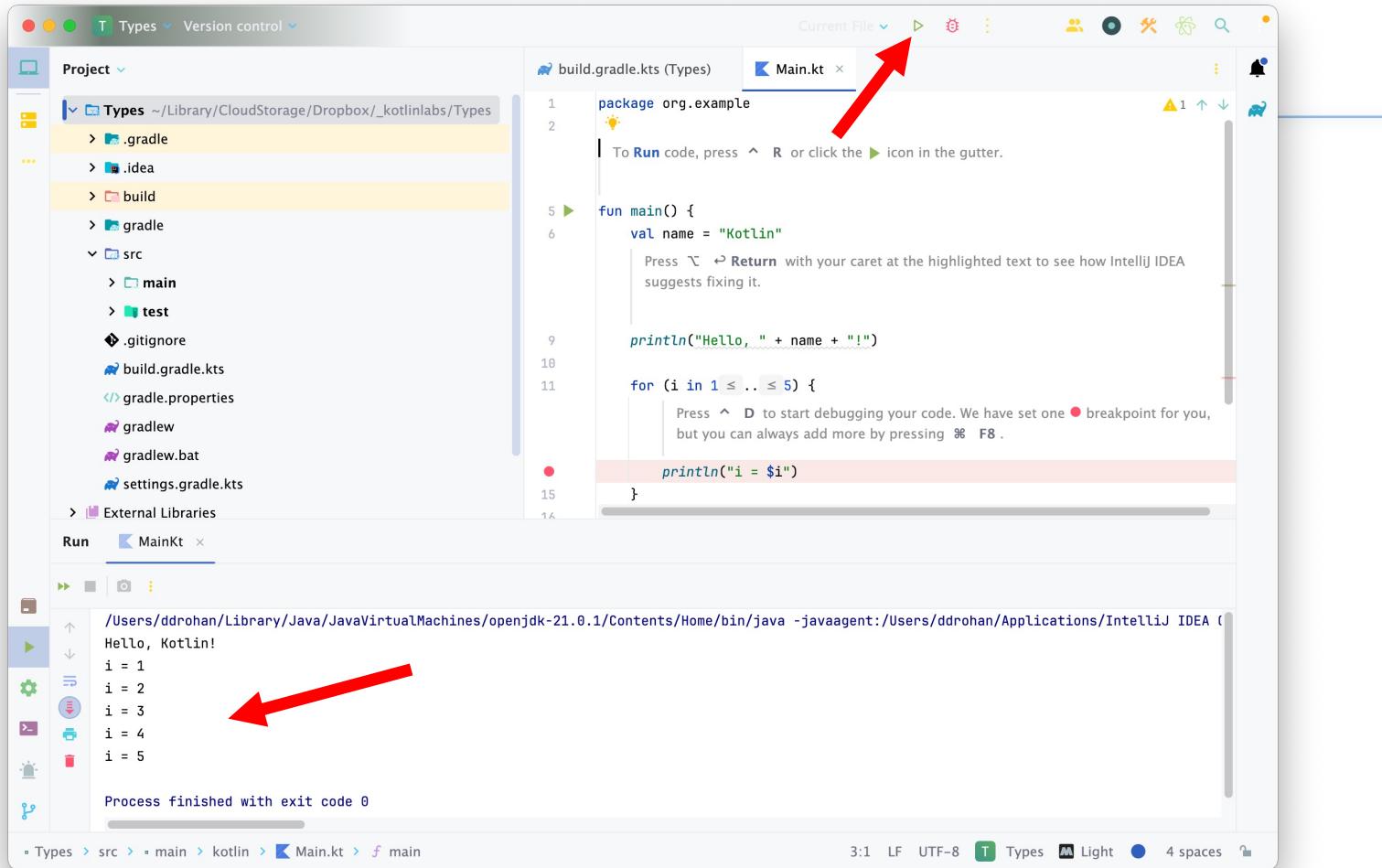
Build Sync Build Output

Types: finished At 17/06 13 min, 40 sec, 565 ms

```
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-project-model/1.9.23
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-tooling-core/1.9.23/
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-util-io/1.9.23/kotli
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-daemon-embeddable/1.
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-daemon-client/1.9.23
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-scripting-jvm/1.9.23
Download https://plugins.gradle.org/m2/org/jetbrains/kotlin/kotlin-scripting-common/1.9
```

BUILD SUCCESSFUL in 6m 37s

# New Project



The screenshot shows the IntelliJ IDEA interface with a new project named "Types". The code editor displays "Main.kt" with the following content:

```

package org.example

fun main() {
    val name = "Kotlin"
    println("Hello, " + name + "!")
    for (i in 1 .. 5) {
        println("i = $i")
    }
}

```

The terminal window at the bottom shows the output of the program:

```

Hello, Kotlin!
i = 1
i = 2
i = 3
i = 4
i = 5

```

At the bottom of the screen, the navigation bar shows the path: "Types > src > main > kotlin > Main.kt > main".

# New Project

**Settings**

**Build, Execution, Deployment > Build Tools > Gradle**

**General Settings**

Gradle user home: `/Users/ddrohan/.gradle` 

Override the default location where Gradle stores downloaded files, e.g. to tune anti-virus software on Windows

Generate \*.iml files for modules imported from Gradle  
Enable if you have a mixed project with IntelliJ IDEA modules and Gradle modules so that it could be shared via VCS

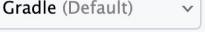
Enable parallel Gradle model fetching for Gradle 7.4+   
Allow collecting Gradle models in parallel during a project reload.

**Gradle Projects**

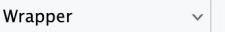
**Types**  Download external annotations for dependencies

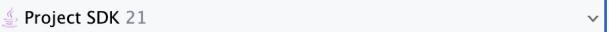
**Build and Run**  
By default IntelliJ IDEA uses Gradle to build the project and run the tasks.  
In a pure Java/Kotlin project, building and running by means of the IDE might be faster, thanks to optimizations. Note, that the IDE doesn't support all Gradle plugins and the project might not be built correctly with some of them.

Build and run using: 

Run tests using: 

**Gradle**

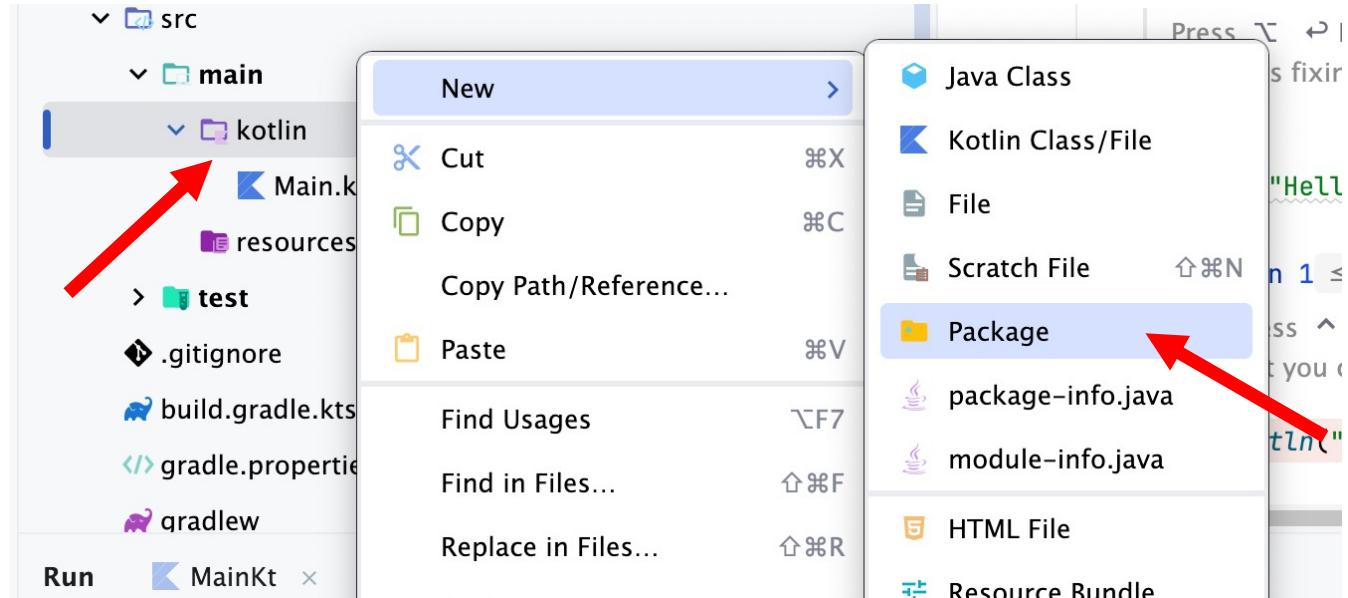
Distribution: 

Gradle JVM: 

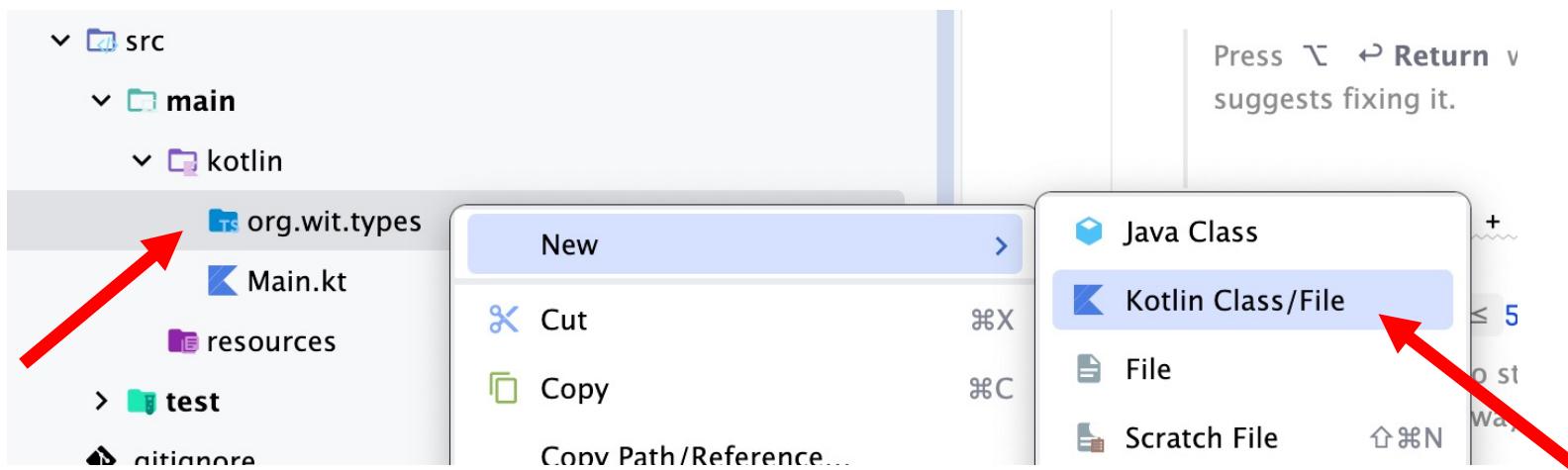
**CANCEL** **APPLY** **OK**



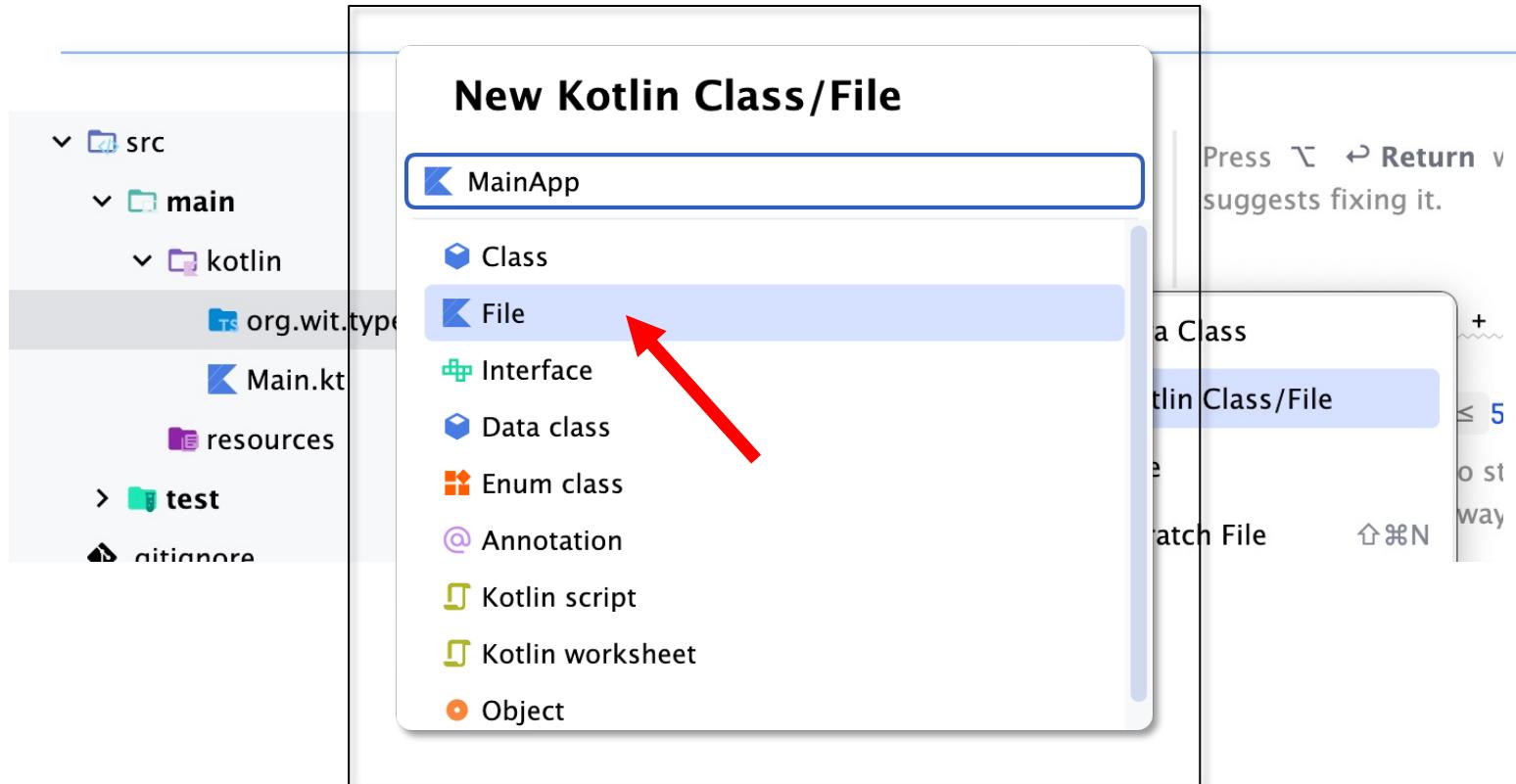

# New Project



# New Project



# New Project

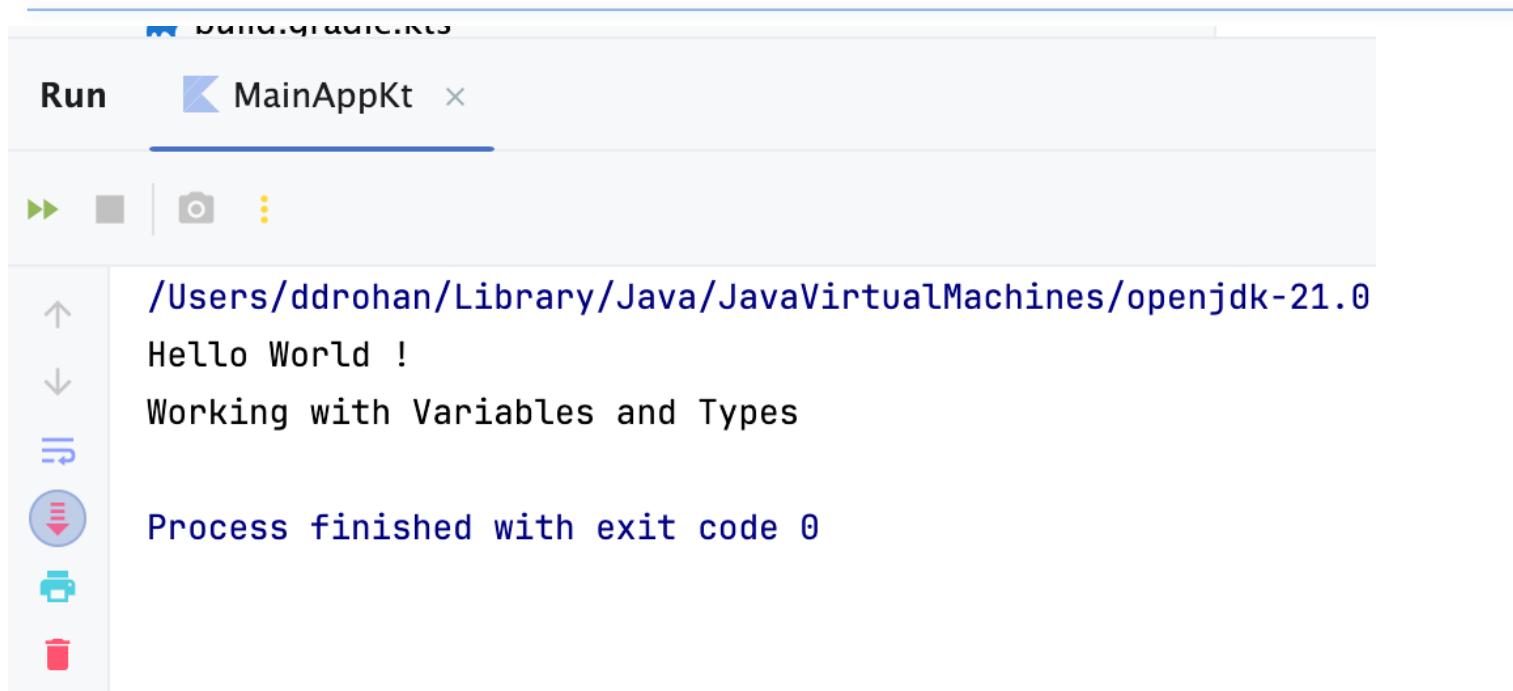


# New Project

MainApp.kt ×

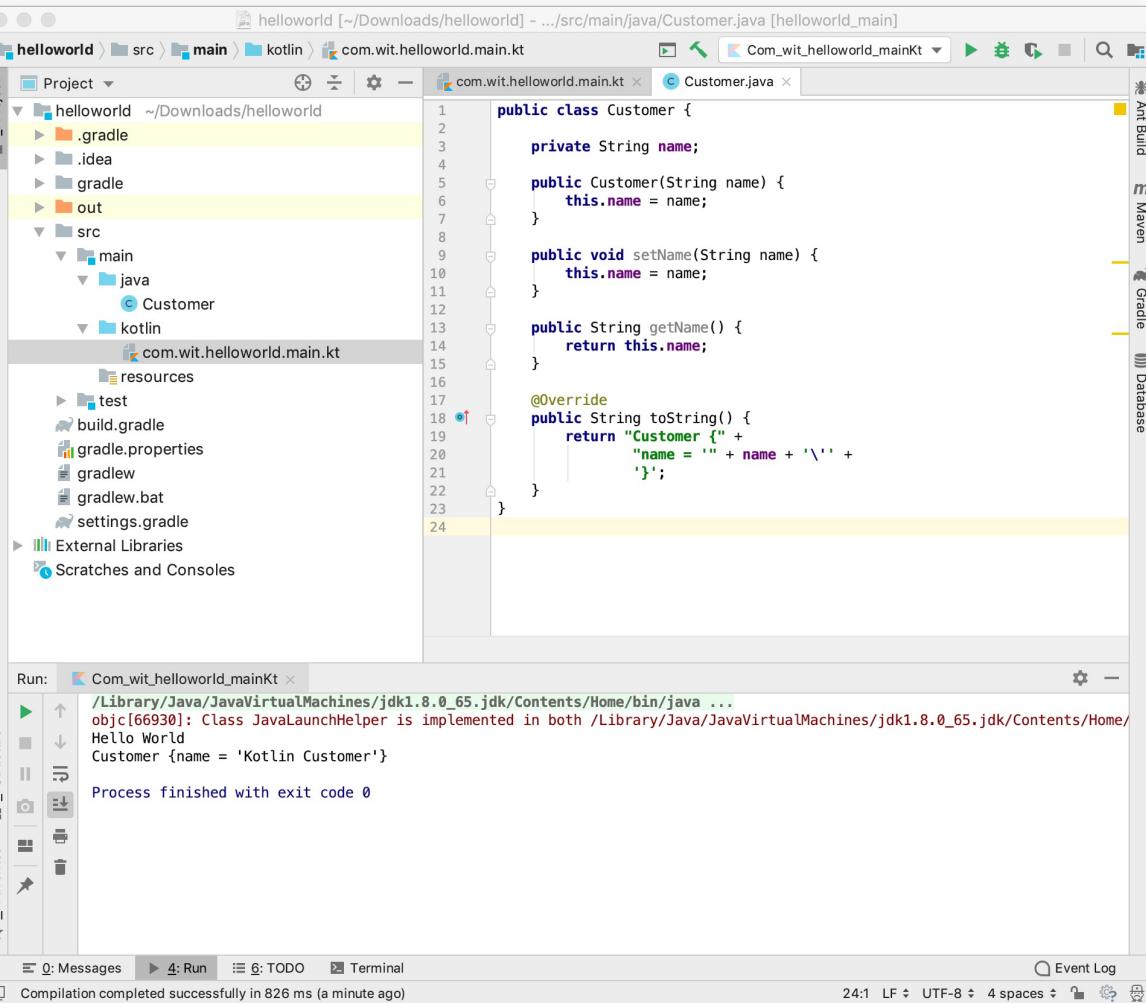
```
1 package org.wit.types
2
3 ➤ fun main() {
4     println("Hello World !")
5     println("Working with Variables and Types")
6 }
7
```

# New Project



The screenshot shows the IntelliJ IDEA interface with the 'Run' tool window open. The title bar says 'Run' and 'MainAppKt'. The main area displays the following text:  
`/Users/ddrohan/Library/Java/JavaVirtualMachines/openjdk-21.0  
Hello World !  
Working with Variables and Types  
Process finished with exit code 0`

# Interoperability



The screenshot shows the IntelliJ IDEA interface with a project named "helloworld". The project structure on the left includes a Java file "Customer.java" and a Kotlin file "com.wit.helloworld.main.kt". The Kotlin file contains the following code:

```

public class Customer {
    private String name;

    public Customer(String name) {
        this.name = name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    @Override
    public String toString() {
        return "Customer {" +
            "name = '" + name + '\'' +
            '}';
    }
}

```

The "Run" tab at the bottom shows the output of running the Kotlin main function:

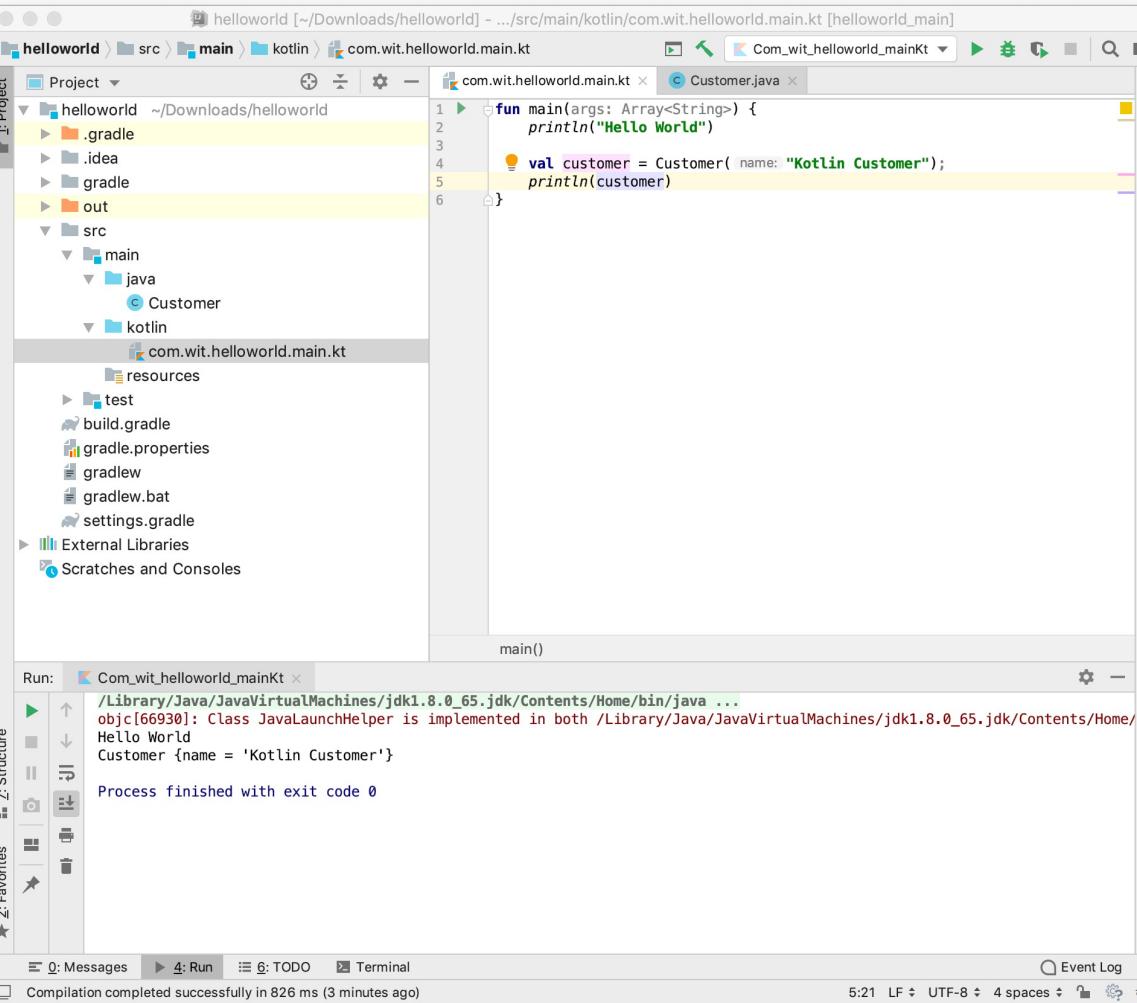
```

/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
objc[66930]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/
Hello World
Customer {name = 'Kotlin Customer'}

Process finished with exit code 0

```

# Interoperability



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure for "helloworld". It includes a Java file named "Customer.java" and a Kotlin file named "com.wit.helloworld.main.kt".
- Code Editor:** Displays the Kotlin code for "com.wit.helloworld.main.kt":
 

```
1 fun main(args: Array<String>) {
2     println("Hello World")
3
4     val customer = Customer( name: "Kotlin Customer");
5     println(customer)
6 }
```
- Run Tab:** Shows the output of running the application:
 

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java ...
objc[6930]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/
Hello World
Customer {name = 'Kotlin Customer'}
```
- Bottom Status Bar:** Shows the message "Compilation completed successfully in 826 ms (3 minutes ago)".

# Interoperability – Java Vs Kotlin

Customer.java

```

1  public class Customer {
2
3      private String name;
4
5      public Customer(String name) { this.name = name; }
6
7      public void setName(String name) { this.name = name; }
8
9      public String getName() { return this.name; }
10
11     @Override
12     public String toString() {
13         return "Customer {" +
14             "name = '" + name + '\'' +
15         '}';
16     }
17 }

```

Customer.kt

```

1  class Customer(name:String?) {
2
3      var name: String? = null
4
5      init { this.name = name }
6
7      override fun toString(): String {
8          return "Customer(name=$name)"
9      }

```

# Conclusion

---

- ❑ Kotlin is low-cost: can be developed in the same IDE, in the same project as Java code
- ❑ It is becoming one of the leading JVM languages
- ❑ It is best suited to work with the IntelliJ IDE
- ❑ And it's currently the 1st choice language for Android Development – so that's why we're using it ☺

# Agenda Recap

---

- ❑ What is Kotlin?
- ❑ Context - Java Vs JVM
- ❑ Kotlin History
- ❑ Milestones
- ❑ General Overview & Features
- ❑ **Kotlin & IntelliJ IDEA**
- ❑ Conclusion

Thanks.

