Your First Multiplatform Project

Madrigal has safely landed on a tropical island after her long flight and is ready to relax and enjoy her vacation. But there is one small problem she did not plan for: No one will accept her Kronstadt gold! She will need to convert her gold into doubloons before she can spend any cash.

Exchange rates fluctuate constantly and unpredictably. To manage currency conversion, you will build a multiplatform application that will display the current exchange rate, ask the user how many doubloons they want, and then return the price in gold.

Begin a new project with the New Project wizard, as you have before. But this time, when choosing a template, select Native Application from the Multiplatform section, as shown in Figure 24.2. As usual, set the Build System to Gradle Groovy and choose a project JDK. Name your new project Doubloons4Gold.

Figure 24.2 Creating a Kotlin Multiplatform project

• • •		New Project	
Maven	Name: ?	Doubloons4Gold	Project structure Project Project F ann MPP Module
 ➡ Android ● IntelliJ Platform Plugin 	Location: ⑦ Project Template: ⑦	JVM	ative Your system
📑 JavaFX		Multiplatform	
Groovy		Gin Library ≢ Native Application	
Kotlin		S Full-Stack Web Application	
Multi-module Project		Browser Application React Application Node.JS Application Compose Multiplatform Native application for a specific platform or several platforms that use Kotlin/Native	
	Build System: 🕐	R Gradle Kotlin R Gradle Groovy M Maven	
	Project JDK:	📜 15 java version "15.0.2" 🔹	
	Artifact Coordinates	S	
	If anything is missing or see Kotlin issue tracker.	ms confusing in the Kotlin New Project wizard, please create an issue or feature request in the	
Cancel			Previous Next

In the wizard's second window, notice the project hierarchy on the lefthand side of the screen, as shown in Figure 24.3. This is where you can define targets for your multiplatform application and customize the hierarchy for your project. You should have one entry under app called native (Figure 24.3).

Figure 24.3 Setting the project template

• • •	New Project	
+ - > Project > app MPP Module # native Your system Target	Use kotlin.test Adds kotlin.test annotations and assert functions for unit testing	
? Cancel		Previous Finish

This page lets you define which targets you want to add in your new project. The Native Application template includes one target, which will match your computer's operating system and CPU architecture. Because you will be compiling your code for multiple targets, you also need to create a 'common" target, which will be for code that runs on every platform. To add a new target, select your app module. Then, click the plus button and choose Common to create a common module (Figure 24.4).

Figure 24.4 Adding a common module

• • •	New Project		
+ - Target Common JVM Kotlin/Native > Kotlin/JS > Android	View region View		
? Cancel		Previous	Finish

Although your template includes a native target, you are not quite ready to build this target. For now, remove the native target by selecting it in the hierarchy and clicking the minus button. You will add this target back to your project manually in Chapter 25. When you finish, your configuration should match Figure 24.5.





Click Finish to finalize these options and create your new project.

IntelliJ will perform an initial build of your project, and you will immediately be greeted with a compiler error:

Please initialize at least one Kotlin target in 'Doubloons4Gold (:)'.

As the error indicates, you need to define a target that your application can compile for.