1. Goal

The goal of this lesson is to show you how to use both Kotlin and Java to build a Spring application.

2. Lesson Notes

The relevant module you need to import when you're starting with this lesson is : m12-lesson2-start

If you want to skip and see the complete implementation, feel free to jump ahead and import: m12-lesson2

3. Maven Setup for Kotlin

The Maven configuration for a Kotlin project is a bit more complex, so let's go over this in detail.

First, we need the *kotlin-stdlib-jdk8* and *jackson-module-kotlin* dependencies, which are managed by Spring Boot.

```
HTML
<dependency>
    <groupId>org.jetbrains.kotlin</groupId>
    <artifactId>kotlin-stdlib-jdk8</artifactId>
</dependency>

<dependency>
    <groupId>com.fasterxml.jackson.module</groupId>
    <artifactId>jackson-module-kotlin</artifactId>
</dependency>
```

Next, let’s create the source directory for kotlin as *src\main\kotlin* in the *um-webapp*module.

Then, we'll configure the *build-helper-maven-plugin* plugin. This is required because, by default, Maven supports one source directory. We're going to use this plugin for the *Kotlin* source.

Let's define the plugin in the *pluginManagement* section, then enable it in the parent POM:

```
HTML
<pluginManagement>
    <plugins>
        <plugin>
```

```
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>build-helper-maven-plugin</artifactId>
            <executions>
                <execution>
                    <id>add-source</id>
                    <phase>generate-sources</phase>
                    <goals>
                        <goal>add-source</goal>
                    </goals>
                    <configuration>
                        <sources>
                            <source>src/main/kotlin</source>
                        </sources>
                    </configuration>
                </execution>
                <execution>
                    <id>add-test-source</id>
                    <phase>generate-test-sources</phase>
                    <goals>
                        <goal>add-test-source</goal>
                    </goals>
                    <configuration>
                        <sources>
                            <source>src/test/kotlin</source>
                        </sources>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</pluginManagement>
```

HTML

```
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>build-helper-maven-plugin</artifactId>
</plugin>
```

We also need to configure the Kotlin Maven plugin which will be used by Maven to compile Kotlin code.

The easiest way to define this configuration is to take this from the Kotlin Reference website. So let's copy the *kotlin-maven-plugin* and *maven-compiler-plugin* configuration code from the Kotlin Reference website to the *pluginManagement* section of the parent POM, then enable these in the *plugins* list.

Note that the Kotlin *src* directory should be listed before the Java *src* directory in the *kotlin-maven-plugin* configuration because the Kotlin compiler should be invoked before the Java compiler.

To enable Spring support, we also need to add the *spring* compiler plugin to the *kotlin-maven-plugin* configuration.

We'll also use the *kotlin-maven-allopen* dependency with *kotlin-maven-plugin.* This is required because in Kotlin everything is *final* by default. An IoC container like Spring will not work if the classes are *final*.

Let's have a look at the full Kotlin configuration for these 2 plugins:

HTML

```
<pluginManagement>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>${maven-compiler-plugin.version}</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
            <executions>
                <!-- Replacing default-compile as it is treated specially by maven -->
                <execution>
                    <id>default-compile</id>
                    <phase>none</phase>
                </execution>
                <!-- Replacing default-testCompile as it is treated specially by maven -->
                <execution>
                    <id>default-testCompile</id>
                    <phase>none</phase>
                </execution>
                <execution>
                    <id>java-compile</id>
                    <phase>compile</phase>
                    <goals>
                        <goal>compile</goal>
                    </goals>
                </execution>
                <execution>
                    <id>java-test-compile</id>
                    <phase>test-compile</phase>
                    <goals>
                        <goal>testCompile</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
        <plugin>
            <groupId>org.jetbrains.kotlin</groupId>
            <artifactId>kotlin-maven-plugin</artifactId>
            <configuration>
                <compilerPlugins>
                    <plugin>spring</plugin>
                </compilerPlugins>
            </configuration>
            <dependencies>
                <dependency>
                    <groupId>org.jetbrains.kotlin</groupId>
                    <artifactId>kotlin-maven-allopen</artifactId>
                    <version>${kotlin.version}</version>
                </dependency>

            </dependencies>
```

```
            <executions>
                <execution>
                    <id>compile</id>
                    <goals>
                        <goal>compile</goal>
                    </goals>
                    <configuration>
                        <sourceDirs>
                            <sourceDir>${project.basedir}/src/main/kotlin</sourceDir>
                            <sourceDir>${project.basedir}/src/main/java</sourceDir>
                        </sourceDirs>
                    </configuration>
                </execution>
                <execution>
                    <id>test-compile</id>
                    <goals>
                        <goal>test-compile</goal>
                    </goals>
                    <configuration>
                        <sourceDirs>
                            <sourceDir>${project.basedir}/src/test/kotlin</sourceDir>
                            <sourceDir>${project.basedir}/src/test/java</sourceDir>
                        </sourceDirs>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</pluginManagement>
```

Before moving code from Java to Kotlin, let’s run the code and see that everything works fine.