Setting up a project (with Spring Boot)

Agenda

1. Spring vs. Spring Boot
2. Maven dependencies
3. The Spring Boot Initializr
4. Exercise

Spring vs. Spring Boot

What is Spring?

The Spring framework provides comprehensive infrastructure support for developing Java Applications.

Packed with some nice features like Dependency Injection and out-of-the-box modules:

Some modules

- Spring JDBC
- Spring Security
- Spring MVC
- Spring AOP
- Spring ORM
- Spring Test

What is Spring Boot?

Spring Boot is basically an extension of the Spring framework, eliminating the boilerplate configurations required for setting up a Spring application.

It takes an opinionated view of the Spring platform, giving preference to convention over extensive configuration.

Some features

- Opinionated starter dependencies to simplify build and application configuration.
- Embedded server to avoid complexity in application deployment.
- Metrics, health check and externalized configuration.
- Automatic config for Spring functionality - whenever possible

Maven dependencies

Minimum web app with Spring

Let’s look at the minimum dependencies required to create a web app using Spring.

Due to Spring’s nature, you need to add each dependency and configure them thoroughly. It requires some average knowledge of the framework and the ecosystem, with typical libraries that solve usual problems.
In this example, we can see the minimum required to setup a webapp are the 2 dependencies shown

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>5.3.5</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.5</version>
</dependency>
```

Minimum web app with Spring Boot

Unlike Spring, Spring Boot requires only one dependency to get a web app up and running.

All other dependencies are added automatically to the final archive during build time

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <version>2.4.4</version>
</dependency>
```

Spring boot starter dependencies

Spring Boot provides a number of starter dependencies for different Spring modules. Some of the most commonly used are the ones shown here on the side.

For a full list of starters, please check https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using-boot-starter

Some starters examples:

- spring-boot-starter-data-jpa
- spring-boot-starter-security
- spring-boot-starter-test
- spring-boot-starter-web
- spring-boot-starter-thymeleaf

Spring Boot Initializr

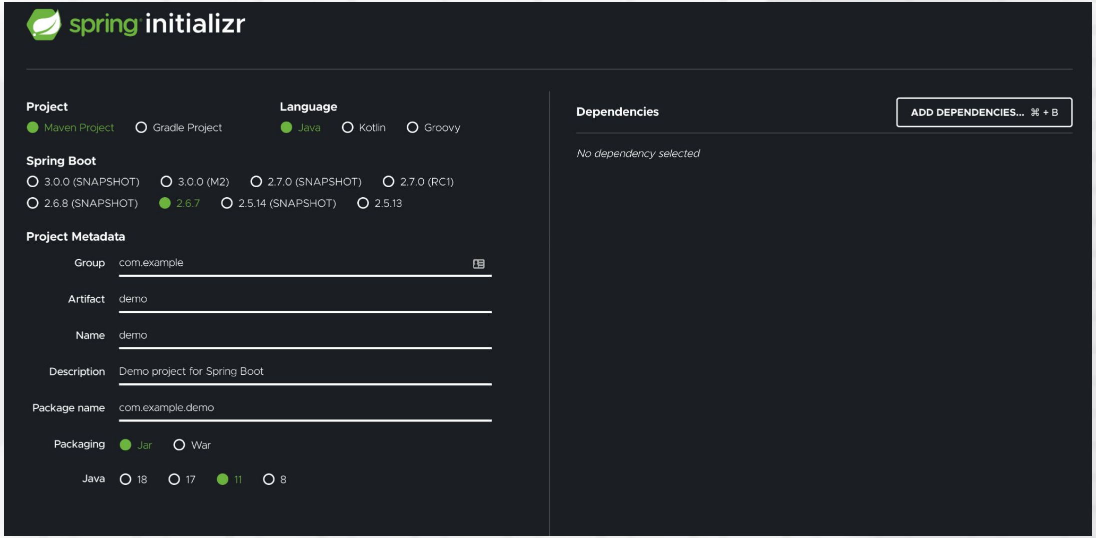
Spring Initializr

Super fast scaffolding

The Spring Initializr (**yes!, without the E, just like early 2010**) is a tool/website (found at https://start.spring.io) that can generate a base application for you.

You just need to specify your maven properties' groupId and artifactId, as well as a name.

You can also specify the packaging type (war or jar), supported java version, build tool (maven or gradle), language (Java, Kotlin or Groovy) and finally some dependencies.



Project
● Maven Project ○ Gradle Project

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 3.0.0 (SNAPSHOT) ○ 3.0.0 (M2) ○ 2.7.0 (SNAPSHOT) ○ 2.7.0 (RC1)
○ 2.6.8 (SNAPSHOT) ● 2.6.7 ○ 2.5.14 (SNAPSHOT) ○ 2.5.13

Project Metadata

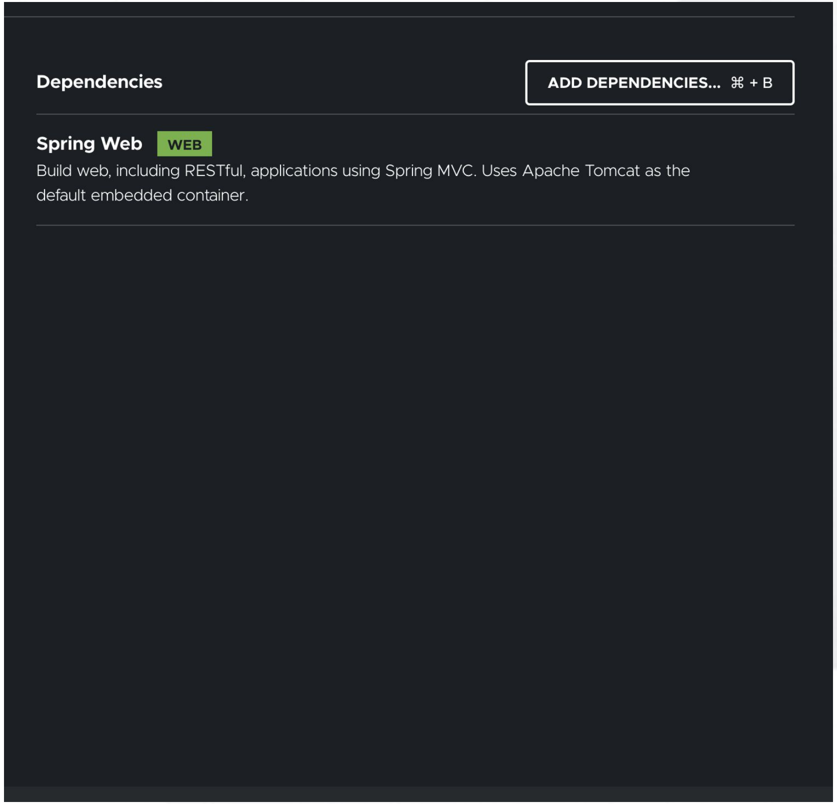
Group: com.example
Artifact: demo
Name: demo
Description: Demo project for Spring Boot
Package name: com.example.demo
Packaging: ● Jar ○ War
Java: ○ 18 ○ 17 ● 11 ○ 8

Dependencies

No dependency selected

Spring boot Initializr dependencies

Now, when you specify dependencies, you can explore the options, but what you'll end up with is simple one or many of the starters already mentioned. In the example shown, you can see we added simple the starter-web

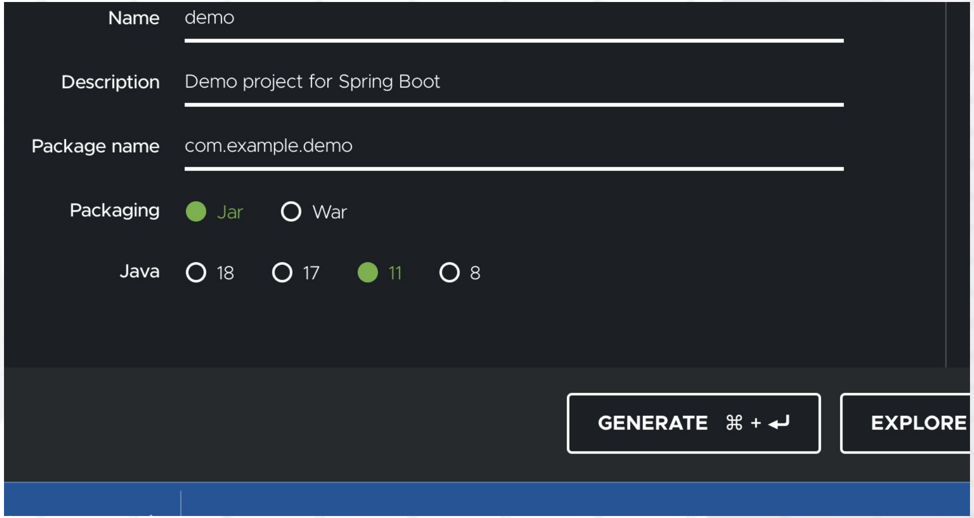


Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring boot Initializr generation

Finally, you click on the Generate button and what you’ll get is a zip file with a project inside. Import this project into your IDE and have fun



Name: demo

Description: Demo project for Spring Boot

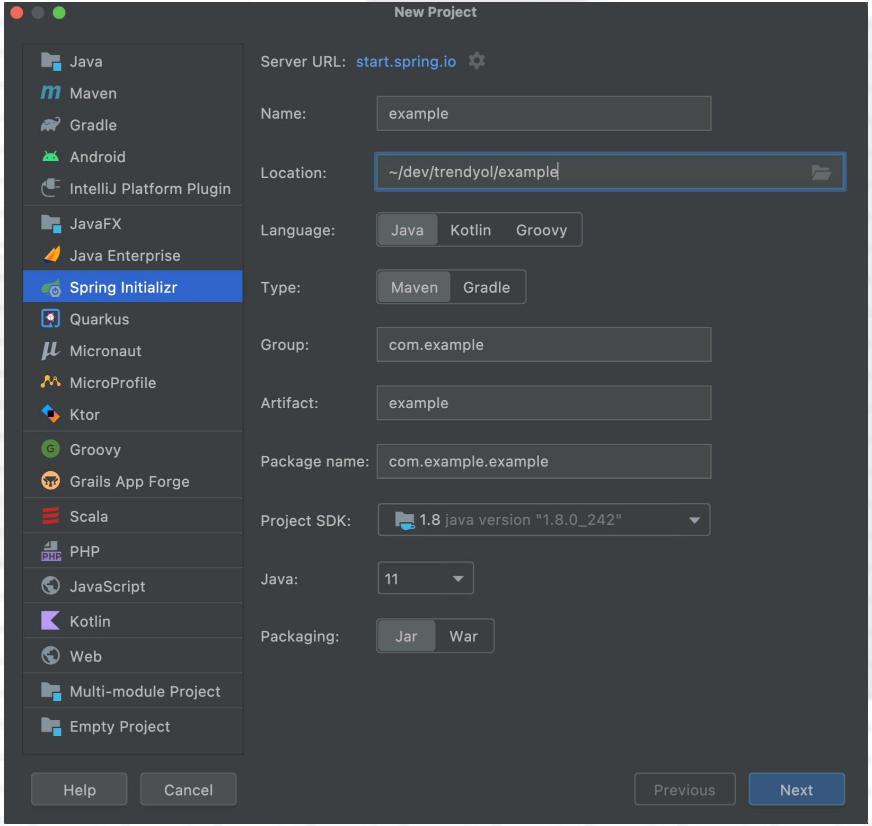
Package name: com.example.demo

Packaging: ● Jar ○ War

Java: ○ 18 ○ 17 ● 11 ○ 8

Spring boot Initializr from IDE

IDEs like IntelliJ IDEA comes with a Spring Initializr wrapper ready to be used without the need of downloading a zip file and import it. It will do everything directly for you.



Server URL: start.spring.io

Name: example

Location: ~/dev/trendyol/example

Language: [✓ Java] Kotlin Groovy

Type: [✓ Maven] Gradle

Group: com.example

Artifact: example

Package name: com.example.example

Project SDK: 1.8 java version "1.8.0_242"

Java: 11

Packaging: [✓ Jar] War

Exercise

Create a new project

- Start a new Spring Boot project from scratch by going to start.spring.io website
- Add the Spring Web starter and generate a new project
- Download it and import it into your IDE
- Create a RestController under the “/hello” path
- Allow it to receive an optional parameter “name”
- Make it return a “Hello World” message or a “Hello ” and the “name” passed as parameter
- Explore the options to build, test and run in your IDE

Thank you

If you have any questions,
please get in touch.

✉

in