

Clara Eusebi

Technical Writing Portfolio

Certification: Technical Writer HQ

Link to Certificate: <https://technicalwriterhq.com/wp-content/uploads/2023/05/Clara-Eusebi.pdf>

Table of Contents

Executive Summary	3
Knowledge Base: Introduction to Spring Boot	3
Information Architecture	4
Product Description: Why Spring Initializr?	5
Audience Persona	6
Who is the Tutorial Geared Towards?	6
“The Mid-Level Developer”	6
About Sara	6
Goals and Needs	6
Motivations	6
Everyday Activities	6
Frustrations	6
Device Usage	6
Step-By-Step Tutorial	7
Purpose of Tutorial	7
Purpose of Spring Initializr website	7
How to Fill out the Spring Initializr Web Page	7
How to generate the project and download the resultant project	9
How to get started using your newly generated boilerplate	9
How to add custom code to your new boilerplate	9
FAQ: Questions and Answers	10

Executive Summary

This portfolio contains seven sections, each corresponding to a type of Technical Writing document. These documents describe the usage of Spring Initializr (available at <https://start.spring.io/>).

Spring Initializr is a web application. The end user enters information about their project into the Spring Initializr web page and then downloads a zip file from that page. This compressed folder has all of the files the end user will need to start a Java microservices application.

In a previous job, I was able to see a demo of Spring Initializr and we immediately adopted it for use, as it was so helpful for creating boilerplate for Spring Boot Java repositories. The effectiveness of the tool is the reason I have chosen it as the subject of this portfolio. This portfolio assumes no knowledge about Spring Boot or Spring Initializr, although the Tutorial assumes familiarity with Java and IntelliJ (an Integrated Development Environment, or IDE).

Knowledge Base: Introduction to Spring Boot

Spring Boot is an extension of the Spring Framework, which works with Java web applications.

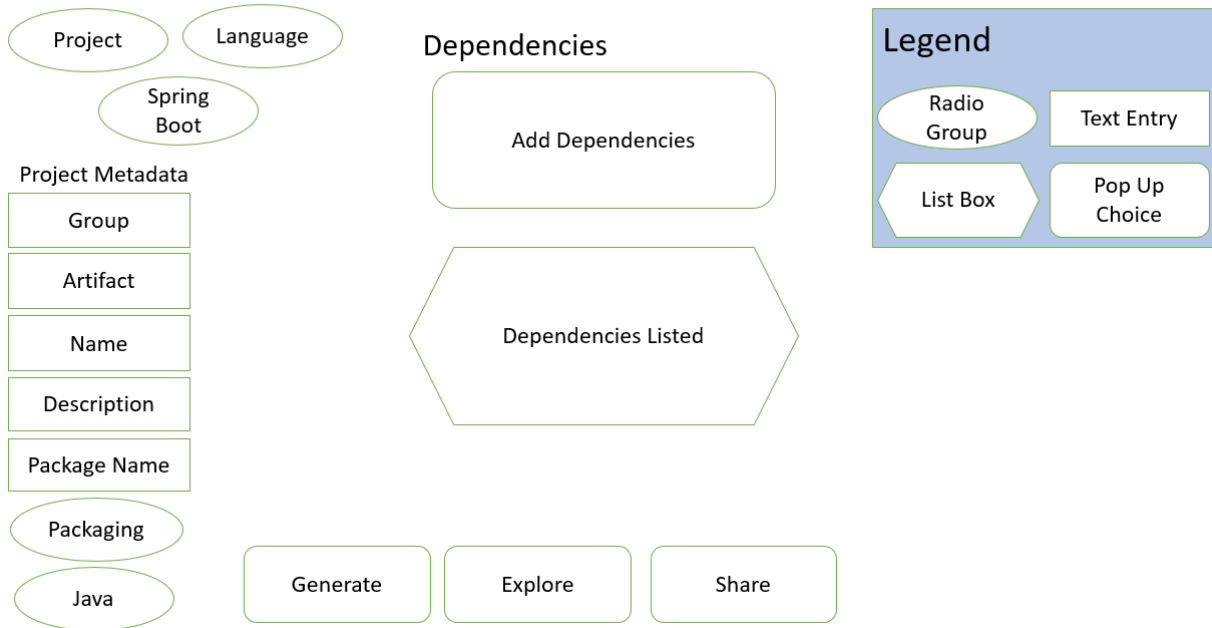
A major advantage to using Spring Boot is that it reduces the boilerplate code you need for Java projects. Spring Initializr offers a further advantage: the boilerplate you do need to generate is created automatically for you, based on your choices when using the web interface.

Please note that although Spring Boot is a Java based tool it can also work with Kotlin and Groovy. The Spring Initializr allows you to choose your programming language from amongst Java, Kotlin, and Groovy. Kotlin and Groovy are newer languages based on Java.

Information Architecture

The following diagram explores the Information Architecture of the Spring Initializr website. Please refer to the Legend, in blue on the right, to learn what the included shapes represent.

Spring Initializr



Product Description: Why Spring Initializr?

Spring Initializr is a website that allows you to generate the basic code needed for a Spring Boot project. The basic code here is referred to as boilerplate. The site streamlines the generation of fully configured Maven and Spring Boot code for use in your project.

There are multiple options available on the Spring Initializr website. All combinations of options will generate valid code. Choose the options that best suit your project and system.

The tutorial below steps through using the Spring Initializr website to create a basic Java project with a Lombok dependency. Lombok is a library that makes it easy to create getters and setters for your POJOs (Plain Old Java Objects) using annotations.

Audience Persona

Who is the Tutorial Geared Towards?

The tutorial is geared towards a Software Engineer with several years in the business and experience using Java software, with a need to run more efficient project setup. They have some familiarity with Spring Boot and with IntelliJ, a Java IDE (Integrated Development Environment). The purpose of the documentation is to give a demonstration of how to use Spring Initializr (<https://start.spring.io/>) to generate boilerplate for new Java Spring Boot projects and to explain why this is a worthwhile endeavor. The tutorial is geared towards the following persona:

“The Mid-Level Developer”

Name: Sara Journey

Age: 35

Location: Boston, MA

Education: BS in Computer Science

About Sara

Sara is working on a team at a fictional company (Microservices-R-U) that needs to streamline getting started with Spring Boot for their Java applications. She is used to working with Java programs but may have struggled to create the initial boilerplate for a new Java Spring Boot repository.

Goals and Needs

Sara wants a fast, easy to use solution to generate a spring boot repository with a full pom file that will include all the dependencies she needs without her having to directly add dependencies to that pom. The pom file is an XML file that serves as the source of truth for the versions of the application, build, and dependencies. Sara would like to be sure the dependencies she does include are compatible with one another.

Motivations

Sara wants to impress her boss and improve her entire team’s process for creating new repositories. She would like to become familiar with the Spring Initializr software so that she can disseminate that knowledge throughout the development team at Microservices-R-U.

Everyday Activities

Sara loves to read Wired and stay on top of news about new technologies. She aspires to become a Senior Software Engineer.

Frustrations

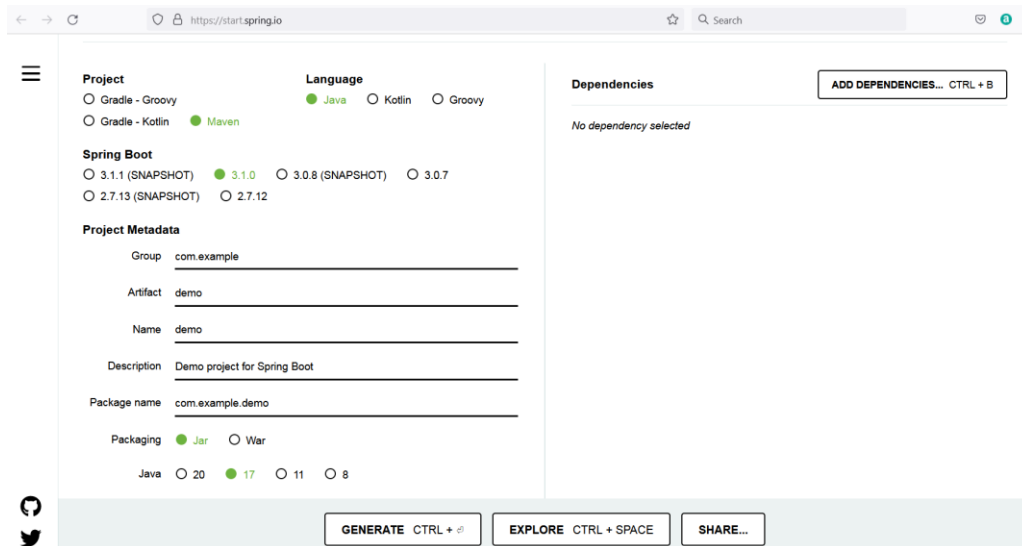
When Sara has tried to create new spring boot projects in the past, she has gotten bogged down with generating the initial repository and trying to create a pom with compatible dependencies. She would love a quick and easy solution to this problem (enter Spring Initializr).

Device Usage

Sara has IntelliJ, an Integrated Development Environment, installed and access to a PC running Windows. She has installed Java on her system and has access to the internet and an internet browser.

Step-By-Step Tutorial

The screen shot below shows the layout of the Spring Initializr website, which you will use in this tutorial.



Purpose of Tutorial

The purpose of the tutorial is to show our user, Sara, how to leverage the Spring Initializr website to generate boilerplate code for her next Java project.

Purpose of Spring Initializr website

The Spring Initializr website helps developers create basic repositories for Spring Boot projects with all of their needed dependencies and metadata already configured.

- **Dependencies** are external libraries you need to include in your application.
- **Metadata** is information about your project. A developer can use the site to choose their configuration and dependencies.
- **Configuration** represents a custom combination of application attributes.

The result of using the website is a downloadable file that contains all of the code needed for getting started with a Java Spring Boot project. The site can also be used for projects in Kotlin and Groovy languages, but these are not covered here to avoid redundancy.

How to Fill out the Spring Initializr Web Page

This tutorial can be used to generate a Java based Spring Boot project. Choosing options other than those specified here will result in a different, but no less valid, boilerplate.

Step 1. High Level Configurations

1. Choose your project. Options include:
 - a. Gradle-Groovy
 - b. Gradle-Kotlin
 - c. Maven

Choose Maven for a Java based project.

2. Choose your language. Options include:
 - a. Java
 - b. Kotlin
 - c. Groovy

Choose Java.

3. Choose your version of Spring Boot. Options include:
 - a. 3.1.1 (SNAPSHOT)
 - b. 3.1.0
 - c. 3.0.8 (SNAPSHOT)
 - d. 3.0.7
 - e. 2.7.13 (SNAPSHOT)
 - f. 2.7.12

Choose 3.1.0. It is the latest supported major build. SNAPSHOTs are pre-release versions.

Step 2. Project Metadata

1. Fill in your metadata. This is unique to your project.
 - a. Enter your Group. The group should correspond to the package name, which you will specify below. It begins with the domain, such as "com.example."
 - b. Enter your Artifact. You can call this "demo."
 - c. Enter a Name. This will become the name of the program, again you can use "demo."
 - d. Enter a Description. This can be anything relevant. An example is, "Demo project for Spring Boot."
 - e. Enter a Package Name, such as "com.example.demo." The package name will correspond with your group name.
 - f. Choose a packaging. Options are:
 - i. Jar
 - ii. War

For the tutorial, use "Jar." "War" files are only used for web applications.

- g. Choose a version of Java. Options are:
 - i. 20
 - ii. 17
 - iii. 11
 - iv. 8

For the tutorial, use "8." The versions are listed most recent to least recent.

Step 3. Dependencies

1. Choose your dependencies. Don't worry about the version numbers of the dependencies, Spring Initializr will coordinate these for you.
 - a. Click "Add Dependencies" or CTRL +B.
 - b. In the popup window, scroll through your options. You will see a list of dependencies and the ones you hover over are highlighted.
 - c. Lombok is an useful library that shows up near the top of the list; add it by clicking on that line.
 - d. To add more dependencies, repeat the process.

How to generate the project and download the resultant project

You can download the file by clicking "Generate" or CTRL+ENTER. Doing so generates a zip file and saves the file to your downloads folder.

How to get started using your newly generated boilerplate

1. Once you have downloaded the zip file, move it from the downloads folder to another folder on your pc or to the desktop so that you can find it in the future.
2. Unzip the file by right clicking and choosing "Extract All."
3. Once you have extracted the files, you can right click on the top-level folder and choose "Edit with IntelliJ IDEA Community Edition." In IntelliJ the project may be slow to open, just be patient.
4. Once it opens, try building the project by selecting "Build Project" from the "Build" menu, or typing CTRL+F9. It should build without issue. If it does not, please review the FAQ for troubleshooting.

How to add custom code to your new boilerplate

1. Insert a simple "Hello World" program to your main method. The main method has been generated automatically; it can be found in your Application entry point file (which has also been generated automatically).
2. The name of the file will be your project name and it will be located in the path src/main/java/package. Here is some code you could use:

```
Scanner in = new Scanner(System.in);  
System.out.println("Enter your name: ");  
String myname = in.nextLine();  
System.out.println("Hello " + myname + "!");
```

3. Please note that this code requires an import statement as well, above the class definition:

```
import java.util.Scanner;
```

4. Run the program.
 - a. Once you have entered your custom code, you can try running the program.
 - b. This particular program will run in the terminal at the bottom of IntelliJ and takes some input. Enter your name at the prompt and see it echoed back to you.

FAQ: Questions and Answers

This FAQ answers 6 questions regarding setup and troubleshooting for the above tutorial.

1. Question: What is boilerplate and why do I need it?
Answer: In this context, boilerplate code refers to the basic setup of a Java software repository which a developer would need to reproduce in the same manner with the same code for multiple such repositories. In other words, boilerplate is the scaffolding required by your project, which will be much the same from one repository to another. The advantage of a tool that generates boilerplate, such as Spring Initializr, is that you can generate the basic setup with custom configurations without having to edit the basic required code manually.
2. Question: What versions of dependencies are included?
Answer: Choose your dependencies without worrying about the versions. All dependencies that are included are the latest version compatible with the Spring Boot build you chose. This is one of the advantages of using the Initializr; you do not need to choose specific versions and need not be concerned that they will not be compatible with one another. Spring Initializr will solely include compatible versions.
3. Question: When I downloaded the package, and tried to run it, I got an error related to the Java version. Why did I get this message?
Answer: Choose the version of Java that you have installed locally in order for the repository to run. Alternatively, you can install a newer version of Java on your machine.
4. Question: Nothing happens when I try to run the program after downloading; why is that?
Answer: You will need to add some functionality! Spring Initializr provided you the foundation of a Java program; to get action out of it you will need to add some code of your own. See the end of the above tutorial for more information on how to add custom code to test the project.
5. Question: Do I need IntelliJ or can I use a different IDE?
Answer: IntelliJ is a great IDE when it comes to Java, and there is a community version that is free to download and use. There are, however, other IDEs suitable for coding work. Look into other IDEs if IntelliJ is not sufficient for you. Examples include Eclipse and Visual Studio Code.
6. Question: I am not on Windows. Can I still use Spring Initializr?
Answer: Yes, Spring Initializr is a website and can be used from any internet friendly Operating System. On Mac, you can access the internet from your favorite browser, and then download and unzip the boilerplate in much the same way as on Windows.