



# Certified Tech Developer

The Ultimate Degree

Back End I

## ¡Vamos a practicar!

### Objetivo

Luego de ver la gran utilidad que tiene Spring Boot, vamos a crear un nuevo proyecto haciendo uso del mismo. ¡Comencemos!

### Instrucciones

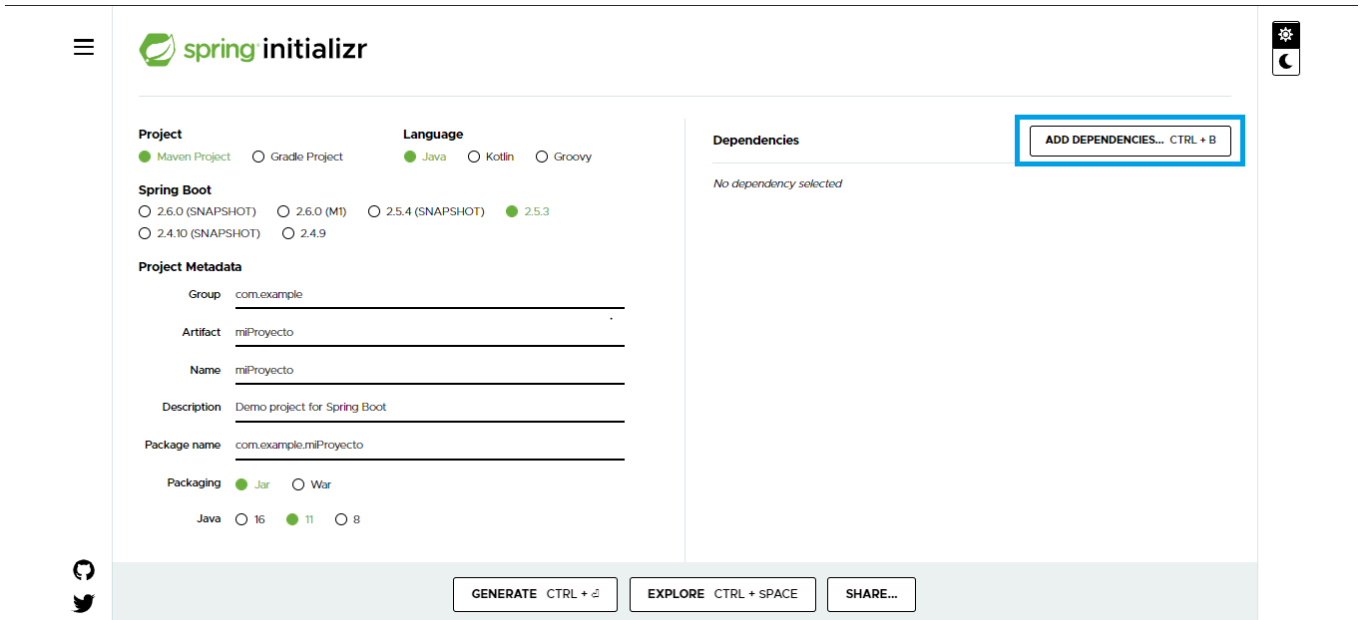
1. Primero, debés ingresar al siguiente link: <https://start.spring.io/>
2. Allí visualizarás la siguiente pantalla. En el apartado "Name" y "Artifact" escribí miProyecto. (Atajo: si lo escribís primero en "Artifact" se completa por defecto el "Name").

Los otros campos debés dejarlos tal cual aparecen.

The screenshot shows the Spring Initializr web application. The interface is divided into several sections:

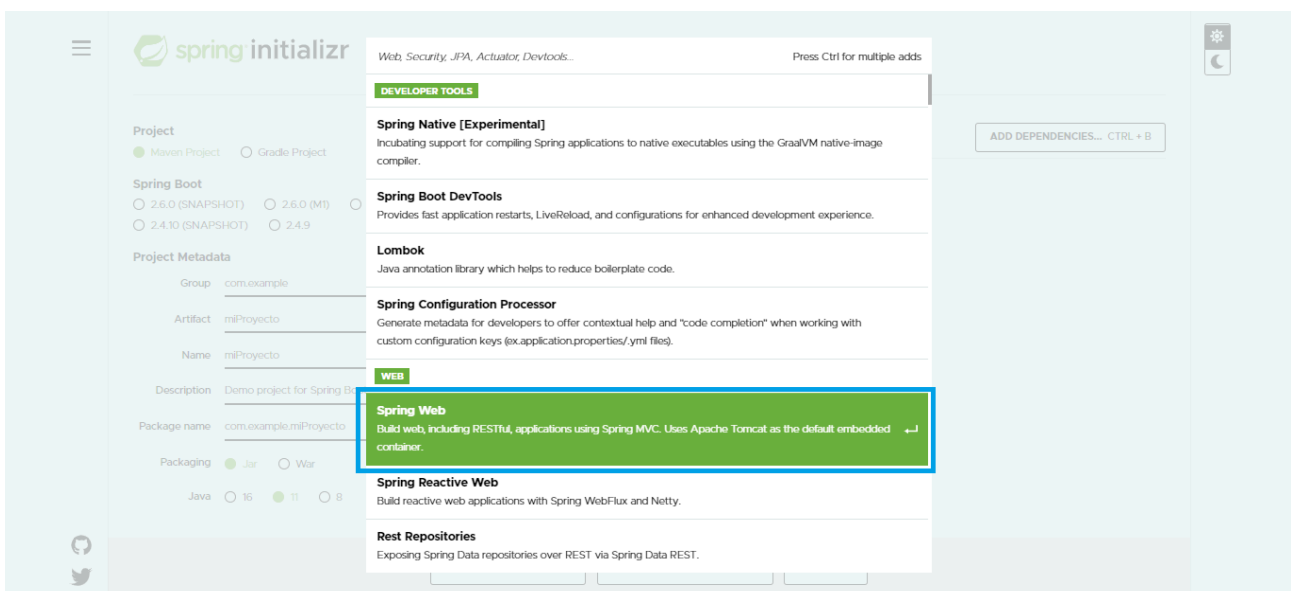
- Project:** Includes radio buttons for "Maven Project" (selected) and "Gradle Project".
- Language:** Includes radio buttons for "Java" (selected), "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for versions: "2.6.0 (SNAPSHOT)", "2.6.0 (M1)", "2.5.4 (SNAPSHOT)", and "2.5.3" (selected). It also includes "2.4.10 (SNAPSHOT)" and "2.4.9".
- Project Metadata:** Includes input fields for "Group" (com.example), "Artifact" (miProyecto), "Name" (miProyecto), "Description" (Demo project for Spring Boot), and "Package name" (com.example.miProyecto).
- Packaging:** Includes radio buttons for "Jar" (selected) and "War".
- Java:** Includes radio buttons for versions: "16", "11" (selected), and "8".
- Dependencies:** Includes a button "ADD DEPENDENCIES... CTRL + B" and the text "No dependency selected".
- Bottom Bar:** Includes buttons "GENERATE CTRL + G", "EXPLORE CTRL + SPACE", and "SHARE...".

- Una vez completo lo mencionado anteriormente, presioná el botón “ADD DEPENDENCIES”.



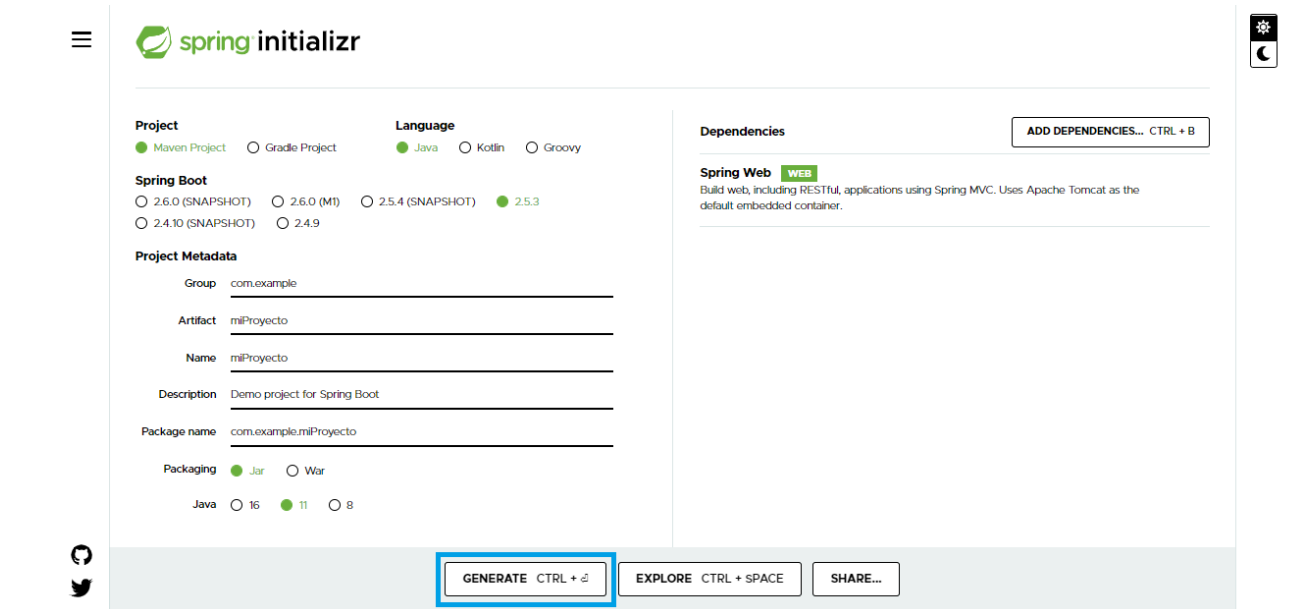
The screenshot shows the Spring Initializr interface. On the left, there are sections for 'Project' (Maven Project selected), 'Language' (Java selected), 'Spring Boot' (2.5.3 selected), and 'Project Metadata' (Group: com.example, Artifact: miProyecto, Name: miProyecto, Description: Demo project for Spring Boot, Package name: com.example.miProyecto). On the right, the 'Dependencies' section is empty with the text 'No dependency selected'. A red box highlights the 'ADD DEPENDENCIES... CTRL + B' button. At the bottom, there are buttons for 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

- Allí se desplegará la siguiente pantalla, donde debés elegir la opción “Spring Web”.



The screenshot shows the Spring Initializr interface with the dependency selection modal open. The modal lists various dependency categories: 'DEVELOPER TOOLS' (Spring Native [Experimental], Spring Boot DevTools, Lombok, Spring Configuration Processor) and 'WEB' (Spring Web, Spring Reactive Web). The 'Spring Web' option is highlighted with a red box. The background shows the same project configuration as the previous screenshot.

5. Luego, presioná "Generate". Ahora se descargará un archivo miProyecto.zip, el cual debés descomprimir y abrir el IntelliJ.



**Project**

☒ Maven Project ☐ Gradle Project

**Language**

☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**

☐ 2.6.0 (SNAPSHOT) ☐ 2.6.0 (M1) ☐ 2.5.4 (SNAPSHOT) ☒ 2.5.3

☐ 2.4.10 (SNAPSHOT) ☐ 2.4.9

**Project Metadata**

Group:

Artifact:

Name:

Description:

Package name:

**Packaging**

☒ Jar ☐ War

Java: ☐ 16 ☒ 11 ☐ 8

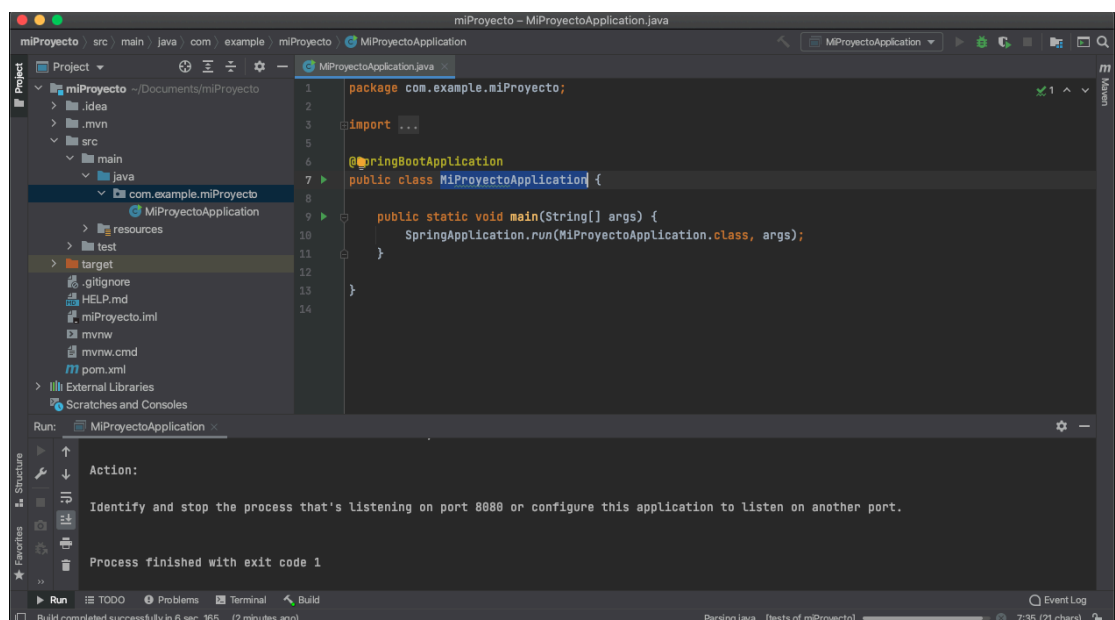
**Dependencies**

**Spring Web** ☒ WEB

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

**Buttons:** GENERATE CTRL + G, EXPLORE CTRL + SPACE, SHARE...

6. Desde el IntelliJ Archivo -> nuevo -> project from existing source. Debés buscar la carpeta miProyecto que descomprimiste y abrirlo desde el archivo pom.xml.
7. Buscá la clase MiProyectoApplication y ejecutá el método main.



```

1 package com.example.miProyecto;
2
3 import ...
4
5
6 @SpringBootApplication
7 public class MiProyectoApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(MiProyectoApplication.class, args);
11     }
12
13 }
14

```

**Run:** MiProyectoApplication

Action:

Identify and stop the process that's listening on port 8080 or configure this application to listen on another port.

Process finished with exit code 1

Build completed successfully in 6 sec, 165 ... (2 minutes ago)

8. Ahora copia y pegá este código para poder crear el Controller. Es a modo de prueba, más adelante veremos qué es un Controller:

```
package com.example.miProyecto;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class MiProyectoApplication {

    public static void main(String[] args) {
        SpringApplication.run(MiProyectoApplication.class, args);
    }

    @GetMapping
    public String holaMundo(){
        return "holaMundo";
    }
}
```

9. Luego, debés ir al browser y correr el localhost:8080/  
10. ¿Obtuviste el "Hola Mundo" en pantalla? 👍👎

**Ahora ya tenés un proyecto Spring boot creado  
y listo para usarse. 🙌🔥💪**