Creating a Spring Web Project Using Maven

Step 1: Generate the Project Structure

Option 1: Using Spring Initializr

Go to https://start.spring.io

Fill in the project details:

Project: Maven

Language: Java

Spring Boot: Latest stable version

Group: com.example (or your domain)

Artifact: demo (or your project name)

Packaging: Jar

Java: 11 or 17 (LTS versions)

Add dependencies:

Spring Web

Click "Generate" to download the project zip file

Unzip and open in your IDE

Option 2: Using Command Line

mvn archetype:generate -DgroupId=com.example -DartifactId=demo -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

Then manually add Spring dependencies to the pom.xml.

Step 2: Project Structure

Your project should have this basic structure:

text

demo/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── example/

│ │ │ └── demo/

│ │ │ └── DemoApplication.java

│ │ └── resources/

│ │ ├── application.properties

│ │ └── static/

│ │ └── templates/

│ └── test/

│ └── java/

│ └── com/

│ └── example/

│ └── demo/

├── pom.xml

Step 3: Complete pom.xml

Here's a complete Maven pom.xml for a Spring Web project:

xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.0</version> <!-- Use latest stable version -->

<relativePath/>

</parent>

<groupId>com.example</groupId>

<artifactId>demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>demo</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

</properties>

<dependencies>

<!-- Spring Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Testing -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Step 4: Create a Simple Controller

Create a new file src/main/java/com/example/demo/HelloController.java:

package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

@GetMapping("/hello")

public String sayHello() {

return "Hello, Spring Boot!";

}

}

Step 5: Main Application Class

The main class should already exist (generated by Spring Initializr):

java

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

Step 6: Run the Application

You can run the application in several ways:

Option 1: Using IDE

Right-click on DemoApplication.java and select "Run"

Option 2: Using Maven

mvn spring-boot:run

Option 3: Build and Run

mvn clean package

java -jar target/demo-0.0.1-SNAPSHOT.jar

Step 7: Test the Application

Open your browser or use curl to test:

curl http://localhost:8080/hello

You should see the response:

text

Hello, Spring Boot!

Additional Configuration

application.properties

Create/edit src/main/resources/application.properties:

properties

# Server port

server.port=8080

# Application name

spring.application.name=DemoApp

# Enable/disable features

spring.main.banner-mode=off