# Spring Boot – Hello World RESTful Web Service

Objective:

To create a simple Spring Boot RESTful web service that returns a 'Hello World' message when accessed through a web browser or REST client.

## 1. Prerequisites

- JDK 8 or above  
- Maven or Gradle  
- Spring Boot (web dependency)  
- IDE (e.g., IntelliJ IDEA, Eclipse, or VS Code)

## 2. Folder Structure

HelloWorldRest/  
├── src/  
│ ├── main/  
│ │ ├── java/  
│ │ │ └── com/example/helloworld/  
│ │ │ ├── HelloWorldController.java  
│ │ │ └── HelloWorldApplication.java  
│ │ └── resources/  
│ │ └── application.properties  
├── pom.xml (for Maven) or build.gradle (for Gradle)

## 3. Step-by-Step Implementation

### Step 1: Create Spring Boot Application Class

File: HelloWorldApplication.java

package com.example.helloworld;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class HelloWorldApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(HelloWorldApplication.class, args);  
 }  
}

### Step 2: Create REST Controller

File: HelloWorldController.java

package com.example.helloworld;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloWorldController {  
  
 @GetMapping("/hello")  
 public String helloWorld() {  
 return "Hello, World!";  
 }  
}

### Step 3: Add Application Configuration

File: application.properties

# Optional: change default port  
# server.port=8081

### Step 4: Build and Run the Application

For Maven:

mvn spring-boot:run

For Gradle:

./gradlew bootRun

## 4. Testing the Endpoint

Open a browser or REST client and access:

http://localhost:8080/hello

Output:

Hello, World!