Exercise 1: Create a Spring Web Project using Maven

Use Web Spring to create a project with the required dependencies.

Generate the zip file and extract it.

**Solution:**

**SpringLearnApplication.java:**

package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 System.*out*.println("Starting Spring Boot Application...");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 System.*out*.println("Spring Boot Application Started Successfully!");  
 }  
}

**pom.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>4.0.0-SNAPSHOT</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Spring Boot Learning project</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <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>  
 <repositories>  
 <repository>  
 <id>spring-milestones</id>  
 <name>Spring Milestones</name>  
 <url>https://repo.spring.io/milestone</url>  
 <snapshots>  
 <enabled>false</enabled>  
 </snapshots>  
 </repository>  
 <repository>  
 <id>spring-snapshots</id>  
 <name>Spring Snapshots</name>  
 <url>https://repo.spring.io/snapshot</url>  
 <releases>  
 <enabled>false</enabled>  
 </releases>  
 </repository>  
 </repositories>  
 <pluginRepositories>  
 <pluginRepository>  
 <id>spring-milestones</id>  
 <name>Spring Milestones</name>  
 <url>https://repo.spring.io/milestone</url>  
 <snapshots>  
 <enabled>false</enabled>  
 </snapshots>  
 </pluginRepository>  
 <pluginRepository>  
 <id>spring-snapshots</id>  
 <name>Spring Snapshots</name>  
 <url>https://repo.spring.io/snapshot</url>  
 <releases>  
 <enabled>false</enabled>  
 </releases>  
 </pluginRepository>  
 </pluginRepositories>  
  
</project>

Application.properties:

# Server Configuration  
server.port=8080  
server.servlet.context-path=/api  
  
# Application Configuration  
spring.application.name=spring-learn  
spring.profiles.active=dev  
  
# DevTools Configuration  
spring.devtools.restart.enabled=true  
spring.devtools.livereload.enabled=true  
  
# Logging Configuration  
logging.level.com.cognizant.springlearn=DEBUG  
logging.level.org.springframework.web=DEBUG  
logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n

**HelloController.java:**

package com.cognizant.springlearn.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
  
 @GetMapping("/")  
 public String home() {  
 return "Welcome to Spring Boot Learning Project!";  
 }  
  
 @GetMapping("/hello")  
 public String hello() {  
 return "Hello from Spring Boot!";  
 }  
  
 @GetMapping("/hello/{name}")  
 public String helloName(@PathVariable String name) {  
 return "Hello, " + name + "!";  
 }  
  
 @GetMapping("/greet")  
 public String greet(@RequestParam(defaultValue = "World") String name) {  
 return "Greetings, " + name + "!";  
 }  
}

**SpringLearnApplicationTests.java:**

package com.cognizant.springlearn;  
  
import org.junit.jupiter.api.Test;  
import org.springframework.boot.test.context.SpringBootTest;  
  
@SpringBootTest  
class SpringLearnApplicationTests {  
  
 @Test  
 void contextLoads() {  
 System.*out*.println("Context loads successfully!");  
 }  
}

**HelloControllerTests.java:**

package com.cognizant.springlearn.controller;  
  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest;  
import org.springframework.test.web.servlet.MockMvc;  
  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*content*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
@WebMvcTest(HelloController.class)  
public class HelloControllerTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @Test  
 public void testHome() throws Exception {  
 mockMvc.perform(*get*("/"))  
 .andExpect(*status*().isOk())  
 .andExpect(*content*().string("Welcome to Spring Boot Learning Project!"));  
 }  
  
 @Test  
 public void testHello() throws Exception {  
 mockMvc.perform(*get*("/hello"))  
 .andExpect(*status*().isOk())  
 .andExpected(*content*().string("Hello from Spring Boot!"));  
 }  
}

**Output:**

