**Exercise 3**

**Spring Boot REST API Testing with JUnit and MockMvc**

**Overview**

This document demonstrates how to write a basic unit test using Spring Boot's testing framework and MockMvc to test a simple REST API endpoint.

**Maven Dependencies**

Add the following dependency to your pom.xml:

***<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
</dependency>***

**Sample Controller (CountryController.java)**

***@RestController  
public class CountryController {  
  
 @GetMapping("/country")  
 public ResponseEntity<Country> getCountry() {  
 Country country = new Country("India", "IN");  
 return ResponseEntity.ok(country);  
 }  
}***

**Test Class (SpringLearnApplicationTests.java)**

This class contains unit tests for the REST endpoint:

***@SpringBootTest  
@AutoConfigureMockMvc  
public class SpringLearnApplicationTests {  
  
 @Autowired  
 private CountryController countryController;  
  
 @Autowired  
 private MockMvc mvc;  
  
 @Test  
 public void contextLoads() {  
 assertNotNull(countryController);  
 }  
  
 @Test  
 public void testGetCountry() throws Exception {  
 mvc.perform(get("/country"))  
 .andExpect(status().isOk());  
 }  
}***

**Notes**

- Ensure the test class is in the correct package structure.

- Use @SpringBootTest and @AutoConfigureMockMvc annotations for integration testing.

- Import all necessary classes such as MockMvc, ResultActions, etc.