**Exercise 8: Test Controller Exception Handling**

**Task:** Add and test a @ControllerAdvice for handling exceptions.

**GlobalExceptionHandler.java :-**

package com.example.demo.exception;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.NoSuchElementException;

@ControllerAdvice

public class GlobalExceptionHandler {

@ExceptionHandler(NoSuchElementException.class)

public ResponseEntity<String> handleNotFound(NoSuchElementException ex) {

return ResponseEntity.status(HttpStatus.NOT\_FOUND).body("User not found");

}

**}**

**UserController.java :-**

package com.example.demo.controller;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.NoSuchElementException;

@RestController

@RequestMapping("/users")

public class UserController {

@Autowired

private UserRepository userRepository;

@GetMapping("/{id}")

public ResponseEntity<User> getUser(@PathVariable Long id) {

User user = userRepository.findById(id)

.orElseThrow(() -> new NoSuchElementException("User not found"));

return ResponseEntity.ok(user);

}

}

**UserControllerExceptionTest.java :-**

package com.example.demo;

import com.example.demo.controller.UserController;

import com.example.demo.exception.GlobalExceptionHandler;

import com.example.demo.repository.UserRepository;

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.boot.test.mock.mockito.MockBean;

import org.springframework.test.web.servlet.MockMvc;

import java.util.Optional;

import static org.mockito.Mockito.when;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

@WebMvcTest(controllers = UserController.class)

public class UserControllerExceptionTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserRepository userRepository;

@Test

public void testUserNotFoundException() throws Exception {

Long userId = 99L;

// Simulate empty DB result

when(userRepository.findById(userId)).thenReturn(Optional.empty());

mockMvc.perform(get("/users/" + userId))

.andExpect(status().isNotFound())

.andExpect(content().string("User not found"));

}

}

Pom.xml :-

<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

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>exception-demo</artifactId>

<version>1.0.0</version>

<name>Exception Handling Demo</name>

<properties>

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

<spring.boot.version>3.2.0</spring.boot.version>

</properties>

<dependencies>

<!-- Spring Boot Web -->

<dependency>

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

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

</dependency>

<!-- Spring Data JPA -->

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 in-memory DB -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Starter Test (JUnit, Mockito, MockMvc, AssertJ) -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

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

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

<version>${spring.boot.version}</version>

</plugin>

<!-- Java Compiler Plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

<compilerArgs>

<arg>-parameters</arg> <!-- Helps with argument reflection -->

</compilerArgs>

</configuration>

</plugin>

</plugins>

</build>

</project>

Output :-

