***WEEK 2***

***Mocking Dependencies in Spring Tests using Mockito***

**Exercise 1: Mocking a Service Dependency in a Controller Test**

**UserController.java**

package com.example.demo.controller;

import com.example.demo.model.User;

import com.example.demo.service.UserService;

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

import org.springframework.http.ResponseEntity;

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

*@RestController*

*@RequestMapping*("/users")

public class UserController {

*@Autowired*

private UserService userService;

*@GetMapping*("/{id}")

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

return ResponseEntity.*ok*(userService.getUserById(id));

}

}

**User.java**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

*@Entity*

public class User {

*@Id*

private Long id;

private String name;

// Constructors

public User() {}

public User(Long id, String name) {

this.id = id;

this.name = name;

}

// Getters & Setters

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

**UserRepository.java**

package com.example.demo.repository;

import com.example.demo.model.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {}

**UserService.java**

package com.example.demo.service;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

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

import org.springframework.stereotype.Service;

*@Service*

public class UserService {

*@Autowired*

private UserRepository userRepository;

public User getUserById(Long id) {

return userRepository.findById(id).orElse(null);

}

}

**UserControllerTest.java**

package com.example.demo;

import com.example.demo.controller.UserController;

import com.example.demo.model.User;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

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.\*;

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;

*@WebMvcTest*(UserController.class)

public class UserControllerTest {

*@Autowired*

private MockMvc mockMvc;

*@*~~MockBean~~

private UserService userService;

*@Test*

public void testGetUser() throws Exception {

User mockUser = new User(1L, "Indhumathi");

*when*(userService.getUserById(1L)).thenReturn(mockUser);

mockMvc.perform(*get*("/users/1"))

.andExpect(*status*().isOk())

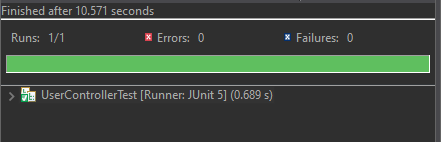
.andExpect(*jsonPath*("$.id").value(1L))

.andExpect(*jsonPath*("$.name").value("Indhumathi"));

}

}

**OUTPUT**

****

**Exercise 2: Mocking a Repository in a Service Test**

**User.java**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

*@Entity*

public class User {

*@Id*

private Long id;

private String name;

public User() {}

public User(Long id, String name) {

this.id = id;

this.name = name;

}

// Getters and setters

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

**UserRepository.java**

package com.example.demo.repository;

import com.example.demo.model.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {}

**UserService.java**

package com.example.demo.service;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

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

import org.springframework.stereotype.Service;

*@Service*

public class UserService {

*@Autowired*

private UserRepository userRepository;

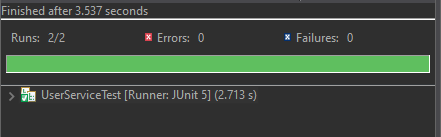
public User getUserById(Long id) {

return userRepository.findById(id).orElse(null);

}

}

**OUTPUT**

****

**Exercise 3: Mocking a Service Dependency in an Integration Test**

**DemoApplication.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);

}

}

**UserController.java**

package com.example.demo.controller;

import com.example.demo.model.User;

import com.example.demo.service.UserService;

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

import org.springframework.http.ResponseEntity;

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

*@RestController*

*@RequestMapping*("/users")

public class UserController {

*@Autowired*

private UserService userService;

*@GetMapping*("/{id}")

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

return ResponseEntity.*ok*(userService.getUserById(id));

}

}

**User.java**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

*@Entity*

public class User {

*@Id*

private Long id;

private String name;

public User() {}

public User(Long id, String name) {

this.id = id;

this.name = name;

}

// Getters and setters

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

**UserRepository.java**

package com.example.demo.repository;

import com.example.demo.model.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {}

**UserService.java**

package com.example.demo.service;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

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

import org.springframework.stereotype.Service;

*@Service*

public class UserService {

*@Autowired*

private UserRepository userRepository;

public User getUserById(Long id) {

return userRepository.findById(id).orElse(null);

}

}

**UserIntegrationTest.java**

package com.example.demo;

import com.example.demo.model.User;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

import static org.mockito.Mockito.*when*;

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

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.mock.mockito.~~MockBean~~;

import org.springframework.http.MediaType;

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.\*;

*@SpringBootTest*

*@AutoConfigureMockMvc*

public class UserIntegrationTest {

*@Autowired*

private MockMvc mockMvc;

*@*~~MockBean~~

private UserService userService;

*@Test*

public void testGetUser\_IntegrationWithMockedService() throws Exception {

User mockUser = new User(1L, "Indhumathi");

*when*(userService.getUserById(1L)).thenReturn(mockUser);

mockMvc.perform(*get*("/users/1")

.contentType(MediaType.***APPLICATION\_JSON***))

.andExpect(*status*().isOk())

.andExpect(*jsonPath*("$.id").value(1L))

.andExpect(*jsonPath*("$.name").value("Indhumathi"));

}

}

**OUTPUT**

