**File: Mocking\_Dependencies\_Spring\_Tests**

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

**Steps:**

1. Create the User entity.

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class User {

@Id

private Long id;

private String name;

// Getters and Setters

}

1. Create the UserService.

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);

}

}

1. Create the UserController.

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));

}

}

1. Write UserControllerTest mocking the service dependency.

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

import org.junit.jupiter.api.extension.ExtendWith;

import org.springframework.http.ResponseEntity;

import static org.junit.jupiter.api.Assertions.\*;

@ExtendWith(MockitoExtension.class)

public class UserControllerTest {

@Mock

private UserService userService;

@InjectMocks

private UserController userController;

@Test

public void testGetUser() {

User mockUser = new User();

mockUser.setId(1L);

mockUser.setName("John Doe");

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

ResponseEntity<User> response = userController.getUser(1L);

assertEquals(1L, response.getBody().getId());

assertEquals("John Doe", response.getBody().getName());

}

}

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

**Steps:**

1. Create the User entity (as shown in Exercise 1).
2. Create the UserRepository.

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

public interface UserRepository extends JpaRepository<User, Long> {

}

1. Create the UserService (as shown in Exercise 1).
2. Write UserServiceTest mocking the repository dependency.

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.jupiter.MockitoExtension;

import org.junit.jupiter.api.extension.ExtendWith;

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Optional;

@ExtendWith(MockitoExtension.class)

public class UserServiceTest {

@Mock

private UserRepository userRepository;

@InjectMocks

private UserService userService;

@Test

public void testGetUserById() {

User mockUser = new User();

mockUser.setId(2L);

mockUser.setName("Jane Smith");

when(userRepository.findById(2L)).thenReturn(Optional.of(mockUser));

User result = userService.getUserById(2L);

assertNotNull(result);

assertEquals(2L, result.getId());

assertEquals("Jane Smith", result.getName());

}

}

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

**Steps:**

1. Create the User entity (as shown earlier).
2. Create the UserService and UserController (as shown earlier).
3. Write the UserIntegrationTest using @SpringBootTest and @AutoConfigureMockMvc.

Example:

import org.junit.jupiter.api.Test;

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.test.web.servlet.MockMvc;

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

import static org.mockito.Mockito.\*;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

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

import org.springframework.http.MediaType;

@SpringBootTest

@AutoConfigureMockMvc

public class UserIntegrationTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserService userService;

@Test

public void testGetUserIntegration() throws Exception {

User mockUser = new User();

mockUser.setId(3L);

mockUser.setName("Test User");

when(userService.getUserById(3L)).thenReturn(mockUser);

mockMvc.perform(get("/users/3")

.accept(MediaType.APPLICATION\_JSON))

.andExpect(status().isOk())

.andExpect(jsonPath("$.id").value(3L))

.andExpect(jsonPath("$.name").value("Test User"));

}

}