**Mockito - Mocking Dependencies in Spring Tests**

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

User.java

@Entity

public class User {

@Id

private Long id;

private String name;

}

UserService.java

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public User getUserById(Long id) {

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

}

}

UserController.java

@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));

}

}

UserControllerTest.java

@ExtendWith(MockitoExtension.class)

public class UserControllerTest {

@InjectMocks

private UserController userController;

@Mock

private UserService userService;

@Test

public void testGetUser() {

User user = new User();

user.setId(1L);

user.setName("John");

Mockito.when(userService.getUserById(1L)).thenReturn(user);

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

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

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

UserRepository.java

public interface UserRepository extends JpaRepository<User, Long> {

}

UserServiceTest.java

@ExtendWith(MockitoExtension.class)

public class UserServiceTest {

@InjectMocks

private UserService userService;

@Mock

private UserRepository userRepository;

@Test

public void testGetUserById() {

User user = new User();

user.setId(1L);

user.setName("Alice");

Mockito.when(userRepository.findById(1L)).thenReturn(Optional.of(user));

User result = userService.getUserById(1L);

assertEquals("Alice", result.getName());

}

}

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

UserIntegrationTest.java

@SpringBootTest

@AutoConfigureMockMvc

public class UserIntegrationTest {

@Autowired

private MockMvc mockMvc;

@MockBean

private UserService userService;

@Test

public void testGetUser() throws Exception {

User user = new User();

user.setId(1L);

user.setName("Bob");

Mockito.when(userService.getUserById(1L)).thenReturn(user);

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

.andExpect(status().isOk())

.andExpect(jsonPath("$.name").value("Bob"));

}

}