**Mocking Dependencies in Spring Tests using Mockito**

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

**Write a unit test for a Spring controller that uses a service to fetch data.**

**UserControllerTest.java**

import com.example.demo.controller.UserController;

import com.example.demo.model.User;

import com.example.demo.service.UserService;

@WebMvcTest(UserController.class)

public class UserControllerTest {

    @Autowired

    private MockMvc mockMvc;

    @MockBean

    private UserService userService;

    @Test

    public void testGetUserById() throws Exception {

        User user = new User();

        user.setId(1L);

        user.setName("Yashaswini");

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

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

                .accept(MediaType.APPLICATION\_JSON))

                .andExpect(status().isOk())

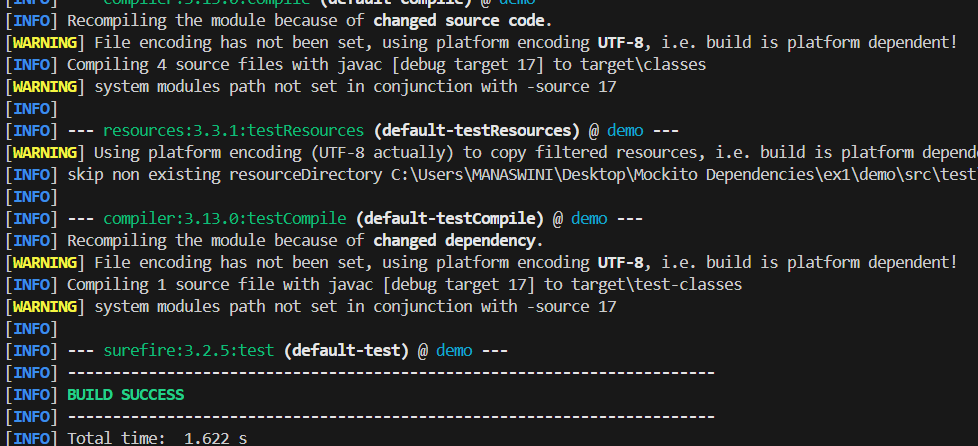
                .andExpect(jsonPath("$.id").value(1L))

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

    }

}

**Output:**

****

**Exercise 2: Mocking a Repository in a Service Test \*\*Task:\*\* Write a unit test for a Spring service that uses a repository to fetch data. Mock the repository dependency using Mockito.**

**UserServiceTest.java**

import com.example.demo.model.User;

import com.example.demo.repository.UserRepository;

import com.example.demo.service.UserService;

@ExtendWith(MockitoExtension.class)

public class UserServiceTest {

    @Mock

    private UserRepository userRepository;

    @InjectMocks

    private UserService userService;

    @Test

    public void testGetUserById() {

        User mockUser = new User();

        mockUser.setId(1L);

        mockUser.setName("Yash");

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

        User result = userService.getUserById(1L);

        assertNotNull(result);

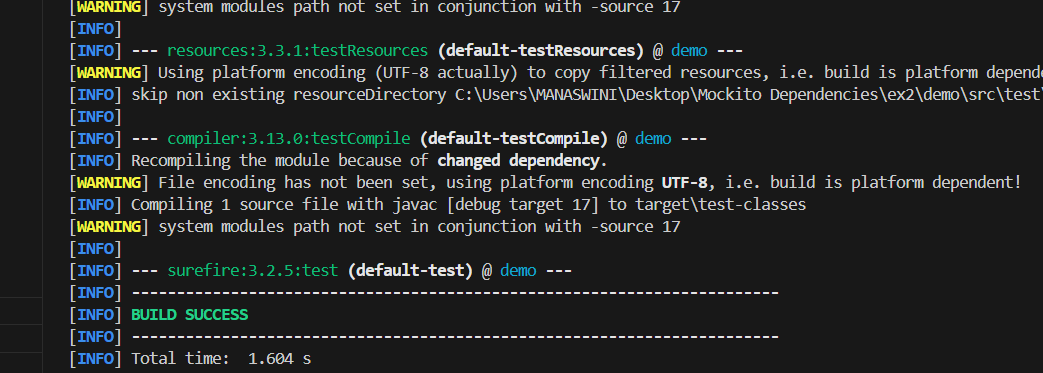
        assertEquals("Yash", result.getName());

        assertEquals(1L, result.getId());

    }

}

**Output:**



**Exercise 3: Mocking a Service Dependency in an Integration Test \*\*Task:\*\* Write an integration test for a Spring Boot application that mocks a service dependency using Mockito.**

**UserServiceTest.java**

import com.example.demo.model.User;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

@SpringBootTest

@AutoConfigureMockMvc

public class UserIntegrationTest {

    @Autowired

    private MockMvc mockMvc;

    @MockBean

    private UserService userService;

    @Test

    public void testGetUserWithMockedService() throws Exception {

        User user = new User();

        user.setId(1L);

        user.setName("Bob");

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

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

                .accept(MediaType.APPLICATION\_JSON))

                .andExpect(status().isOk())

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

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

    }

}

**Output:**

