

# **Lab Exercise 8- API Testing Using JUnit for Spring Boot**

---

## **1. Objective**

In this lab, you will:

1. Write JUnit tests for your CRUD API
  2. Test your service layer (UserService)
  3. Test your controller layer (UserController) by calling methods directly
  4. Validate returned objects and ResponseEntity
- 

## **2. Prerequisites**

Your project must already contain:

User.java

UserService.java

UserController.java

From Lab Exercise 4.

JUnit dependencies are included by:

spring-boot-starter-test

---

### 3. Project Structure With Tests

```
simpleapi/
├── src/
│   ├── main/java/com/example/simpleapi/
│   │   ├── controller/UserController.java
│   │   ├── service/UserService.java
│   │   └── model/User.java
│   └── test/java/com/example/simpleapi/
│       ├── UserServiceTest.java
│       └── UserControllerTest.java
└── pom.xml
```

### 4. Test 1: Service Layer Test (UserServiceTest)

Create file:

```
src/test/java/com/example/simpleapi/UserServiceTest.java
```

Code:

```
package com.example.simpleapi;

import com.example.simpleapi.model.User;
import com.example.simpleapi.service.UserService;
import org.junit.jupiter.api.Test;

import java.util.List;

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

```
public class UserServiceTest {  
  
    @Test  
  
    public void testfindAll() {  
  
        UserService svc = new UserService();  
  
        List<User> users = svc.findAll();  
  
  
        assertTrue(users.size() >= 2); // Alice and Bob seeded  
    }  
  
  
    @Test  
  
    public void testSaveUser() {  
  
        UserService svc = new UserService();  
  
        User user = new User(null, "Hitesh", "hitesh@test.com");  
  
        User saved = svc.save(user);  
  
  
        assertNotNull(saved.getId());  
        assertEquals("Hitesh", saved.getName());  
    }  
  
  
    @Test  
  
    public void testfindById() {  
  
        UserService svc = new UserService();  
  
        User u = svc.findById(1L);
```

```
assertNotNull(u);

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

}

@Test

public void testDeleteUser() {

    UserService svc = new UserService();

    boolean deleted = svc.delete(1L);

    assertTrue(deleted);

    assertNull(svc.findById(1L));

}

}
```

---

## 5. Test 2: Controller Layer Test (UserControllerTest)

Create file:

```
src/test/java/com/example/simpleapi/UserControllerTest.java
```

Code:

```
package com.example.simpleapi;

import com.example.simpleapi.controller.UserController;
import com.example.simpleapi.model.User;
```

```
import com.example.simpleapi.service.UserService;  
import org.junit.jupiter.api.Test;  
import org.springframework.http.ResponseEntity;  
  
import java.util.List;  
  
import static org.junit.jupiter.api.Assertions.*;  
  
public class UserControllerTest {  
  
    @Test  
    public void testListUsers() {  
        UserService svc = new UserService();  
        UserController controller = new UserController(svc);  
  
        List<User> list = controller.list();  
  
        assertTrue(list.size() >= 2);  
    }  
  
    @Test  
    public void test GetUser() {  
        UserService svc = new UserService();  
        UserController controller = new UserController(svc);
```

```
    ResponseEntity<User> response = controller.get(1L);

    assertEquals(200, response.getStatusCode().value());
    assertEquals("Alice", response.getBody().getName());
}

@Test
public void testCreateUser() {

    UserService svc = new UserService();
    UserController controller = new UserController(svc);

    User user = new User(null, "Hitesh", "hitesh@example.com");

    ResponseEntity<User> response = controller.create(user);

    assertEquals(201, response.getStatusCode().value());
    assertEquals("Hitesh", response.getBody().getName());
    assertNotNull(response.getHeaders().getLocation());
}

@Test
public void testUpdateUser() {
    UserService svc = new UserService();
```

```
UserController controller = new UserController(svc);

User updated = new User(null, "Updated Name", "updated@example.com");

ResponseEntity<User> response = controller.update(1L, updated);

assertEquals(200, response.getStatusCode().value());
assertEquals("Updated Name", response.getBody().getName());

}

@Test
public void testDeleteUser() {

    UserService svc = new UserService();
    UserController controller = new UserController(svc);

    ResponseEntity<Void> response = controller.delete(1L);

    assertEquals(204, response.getStatusCode().value());
    assertNull(svc.findById(1L));
}

=====
```

## 5. Running the Tests

### Using Maven:

```
mvn test
```

### Using IDE:

Right-click → Run As → JUnit Test

Expected output:

```
Tests run: 10, Failures: 0, Errors: 0
```

```
BUILD SUCCESS
```

```
=====
```