**Cognizant Java FSE – (Deep Skilling)**

**(WEEK-2)**

**MODULE 2:** TDD using Mockito

**MODULE 3:** SLF4J logging framework

**Submitted by**

Name: S Subhashri

Roll No: 111522102152

Email: 22102152@rmd.ac.in

College: RMD ENGINEERING COLLEGE

Batch: Java FSE – 2026

Superset ID: 6397078

**Mockito Hands-On Exercises:**

**Exercise 1: Mocking and Stubbing**

**Scenario:**

You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

**Steps:**

1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

3. Write a test case that uses the mock object

**Solution:**

**ExternalApi.java**

package com.example;  
  
public interface ExternalApi {  
 String getData();  
}

**MyService.java**

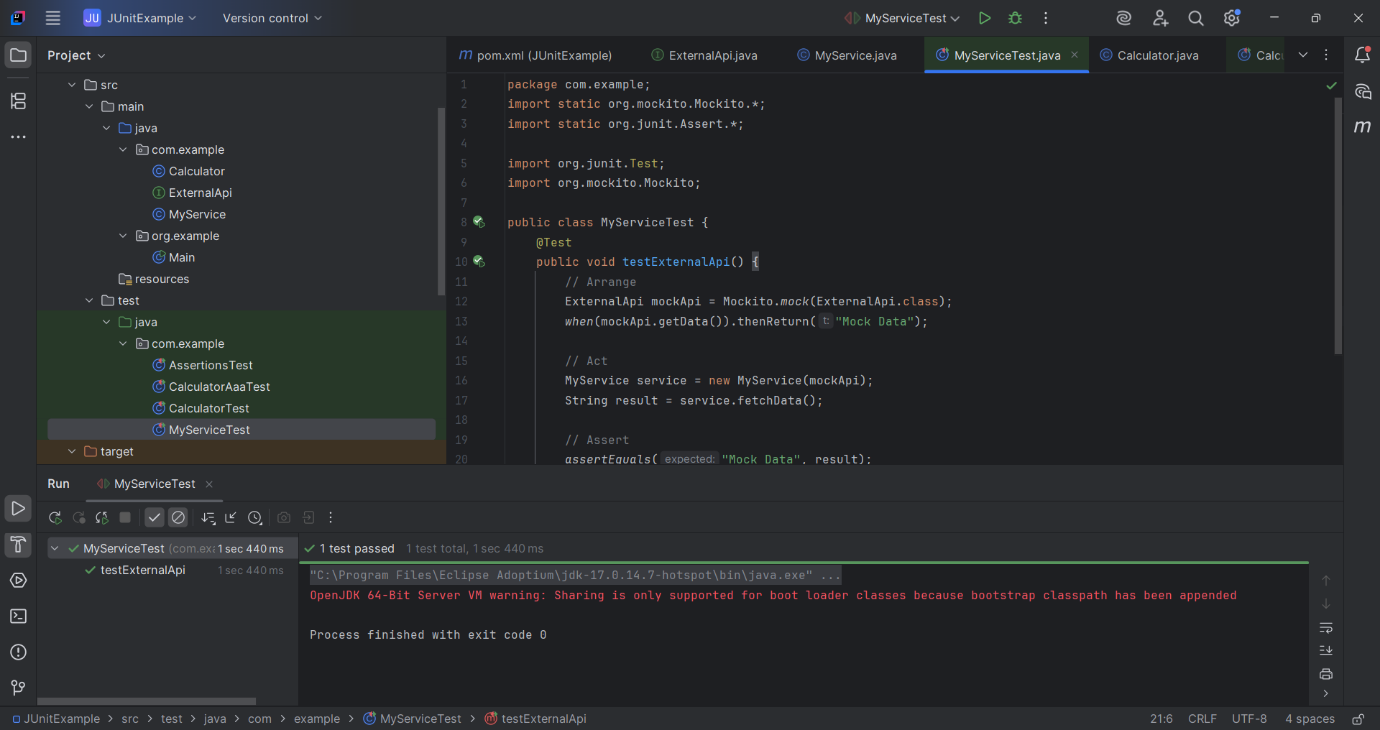
package com.example;  
  
public class MyService {  
 private ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}

**MyServiceTest.java**

package com.example;  
import static org.mockito.Mockito.\*;  
import static org.junit.Assert.\*;  
  
import org.junit.Test;  
import org.mockito.Mockito;  
  
public class MyServiceTest {

@Test  
 public void testExternalApi() {  
 // Arrange  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 // Act  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 // Assert  
 *assertEquals*("Mock Data", result);  
 }  
}

**Output:**

****

**Exercise 2: Verifying Interactions**

**Scenario:**

You need to ensure that a method is called with specific arguments.

**Steps:**

1. Create a mock object.

2. Call the method with specific arguments.

3. Verify the interaction.

**Solution:**

**MyServiceVerifyTest.java**

package com.example;  
import static org.mockito.Mockito.\*;  
import org.junit.Test;  
import org.mockito.Mockito;  
public class MyServiceVerifyTest {  
 @Test  
 public void testVerifyInteraction() {  
 // Create a mock of the ExternalApi  
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);  
  
 // Inject the mock into the service  
 MyService service = new MyService(mockApi);  
  
 // Call the method that internally calls mockApi.getData()  
 service.fetchData();  
  
 // Verify that getData() was actually called  
 *verify*(mockApi).getData();  
 }  
}

**Output:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Logging using SLF4J:**

**Exercise 1: Logging Error Messages and Warning Levels**

**Task:**

Write a Java application that demonstrates logging error messages and warning levels using SLF4J.

**Solution:**

**LoggingExample.java**

package com.example;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
public class LoggingExample {  
 private static final Logger *logger* = LoggerFactory.*getLogger*(LoggingExample.class);  
  
 public static void main(String[] args) {  
 *logger*.error("This is an error message");

*logger*.warn("This is a warning message");  
 }  
}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**