**Mockito Hands-On Exercises**  
  
**Exercise 1:**

**Mocking and Stubbing**pom.xml:

<project> xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.geetha.mock</groupId>

<artifactId>MockitoDemoProject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Mockito Demo Project</name>

<dependencies>

<!-- ✅ JUnit 5 -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.2</version>

<scope>test</scope>

</dependency>

<!-- ✅ Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- ✅ Plugin to run JUnit 5 tests -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.2.5</version>

</plugin>

</plugins>

</build>

</project>

ExternalApi.java:

**package** com.geetha.mock;

**public** **interface** ExternalApi {

String getData();

}

Myservice.java:

**package** com.geetha.mock;

**public** **class** MyService {

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

MyServiceTest.java:

**package** com.geetha.mock;

**import** **static** org.mockito.Mockito.\*;

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

**import** org.junit.jupiter.api.Test;

**public** **class** MyServiceTest {

@Test

**public** **void** testExternalApiMocking() {

ExternalApi mockApi = *mock*(ExternalApi.**class**); // mock object

*when*(mockApi.getData()).thenReturn("Mock Data"); // stub method

MyService service = **new** MyService(mockApi);

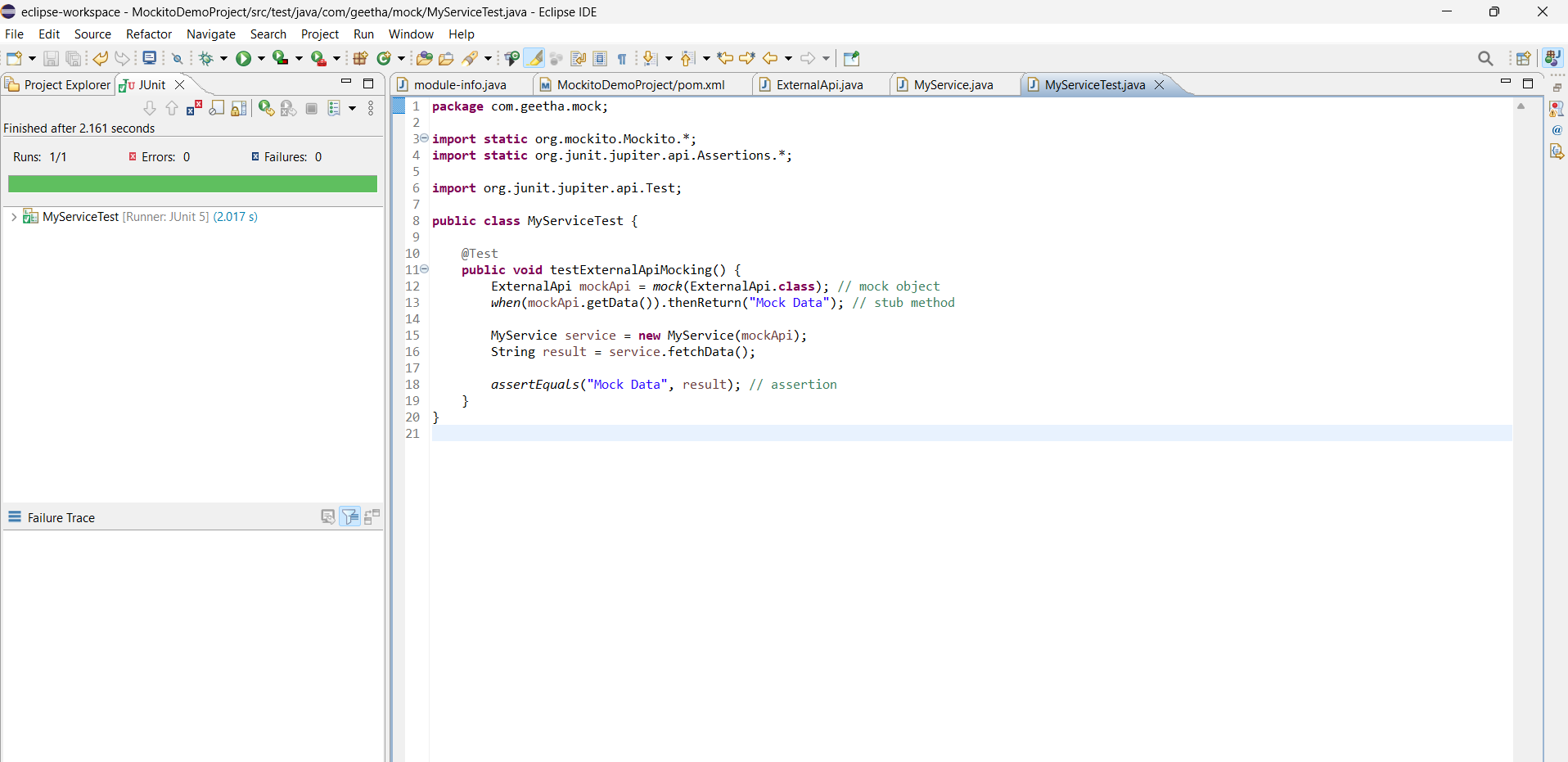
String result = service.fetchData();

*assertEquals*("Mock Data", result); // assertion

}

}

**OUTPUT:**



**Exercise 2:**

**Verifying Interactions**

pom.xml:

<project> xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.geetha.mock</groupId>

<artifactId>MockitoDemoProject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Mockito Demo Project</name>

<dependencies>

<!-- ✅ JUnit 5 -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.2</version>

<scope>test</scope>

</dependency>

<!-- ✅ Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- ✅ Plugin to run JUnit 5 tests -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.2.5</version>

</plugin>

</plugins>

</build>

</project>

ExternalApi.java:

**package** com.geetha.mock;

**public** **interface** ExternalApi {

String getData();

}

Myservice.java:

**package** com.geetha.mock;

**public** **class** MyService {

**private** ExternalApi api;

**public** MyService(ExternalApi api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

MyServiceTest.java:

**package** com.geetha.mock;

**import** **static** org.mockito.Mockito.\*;

**import** org.junit.jupiter.api.Test;

**public** **class** MyServiceTest {

@Test

**public** **void** testVerifyMethodCalled() {

ExternalApi mockApi = *mock*(ExternalApi.**class**);

MyService service = **new** MyService(mockApi);

System.***out***.println("Calling service.fetchData()...");

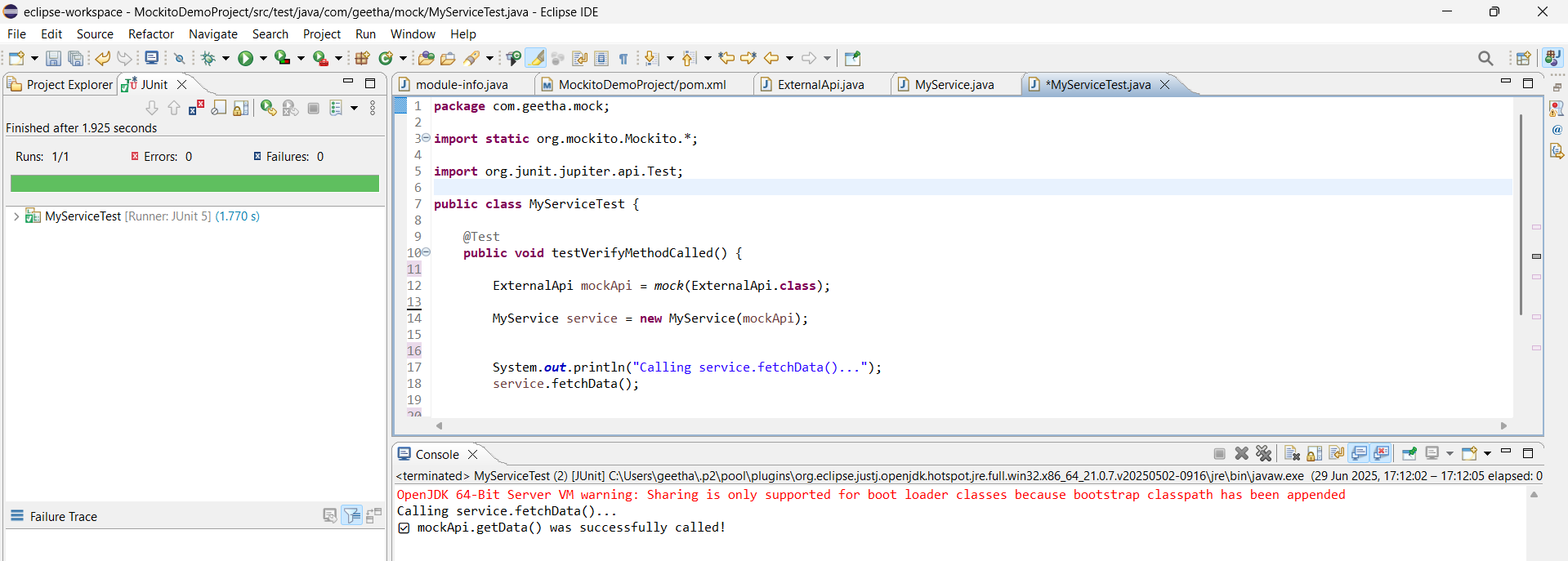
service.fetchData();

*verify*(mockApi).Data();

System.***out***.println("✅ mockApi.getData() was successfully called!");

}

}

**OUTPUT:**  


**Logging using SLJ4 framework**

**Exercise 1:**

**Logging Error Messages and Warning Levels**

Pom.xml:

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.geetha.logging</groupId>

<artifactId>LoggingApp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>LoggingApp</name>

<dependencies>

<!-- SLF4J API -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<!-- Logback (backend for SLF4J) -->

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

</dependencies>

</project>

LoggingExample.java:

**package** com.geetha.logging;

**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:**  
