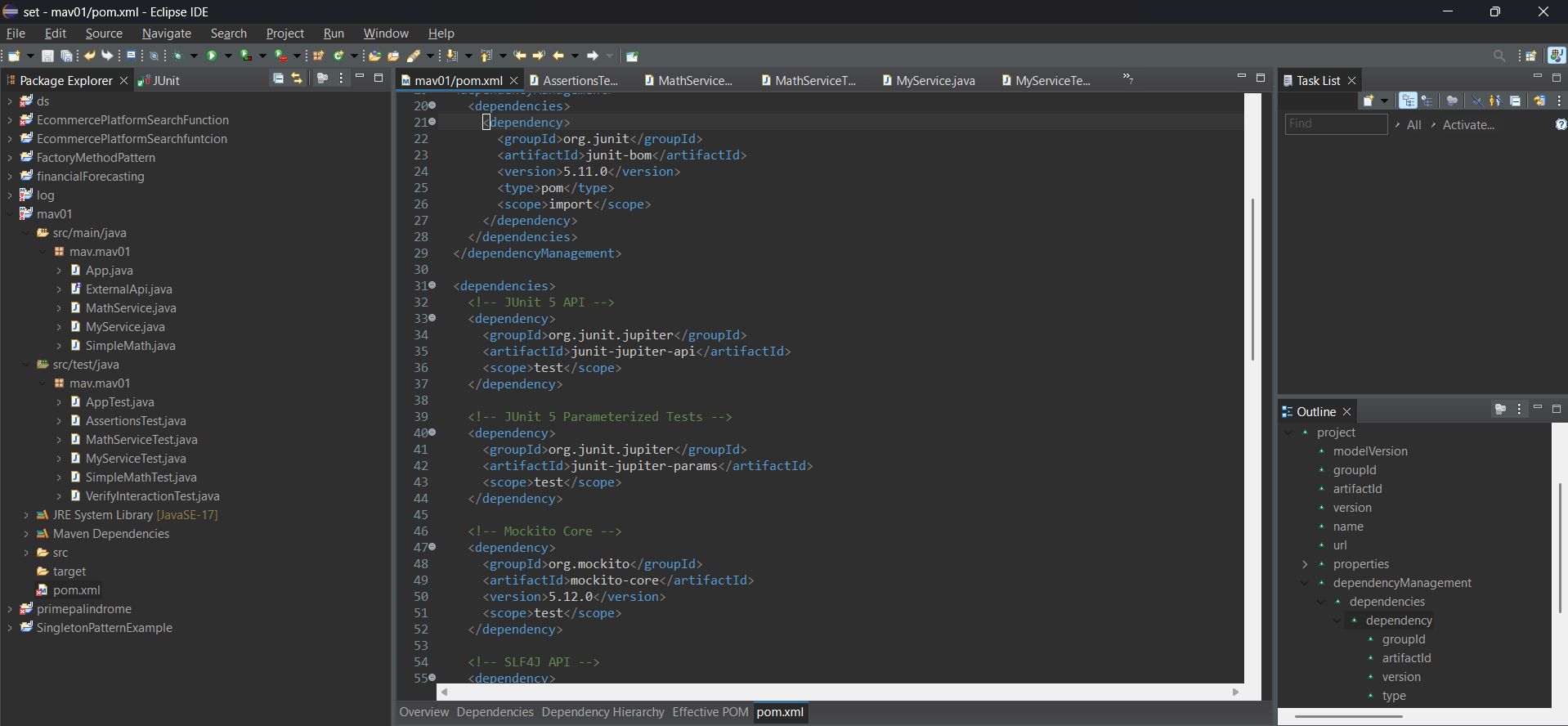
3. Mockito exercises

# 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.

**Adding Dependencies to Pom.xml :**



<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.12.0</version>

<scope>test</scope>

</dependency>

**ExternalApi.java:**

package mav.mav01;

public interface ExternalApi {

String getData();

}

**MyService.java:**

package mav.mav01;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // Real call replaced by mock

}

}

**MyServiceTest.java:**

package mav.mav01;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

@Test

void testExternalApi() {

// Step 1: Create mock

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Stub method

when(mockApi.getData()).thenReturn("Mock Data");

// Step 3: Inject mock into service

MyService service = new MyService(mockApi);

// Act

String result = service.fetchData();

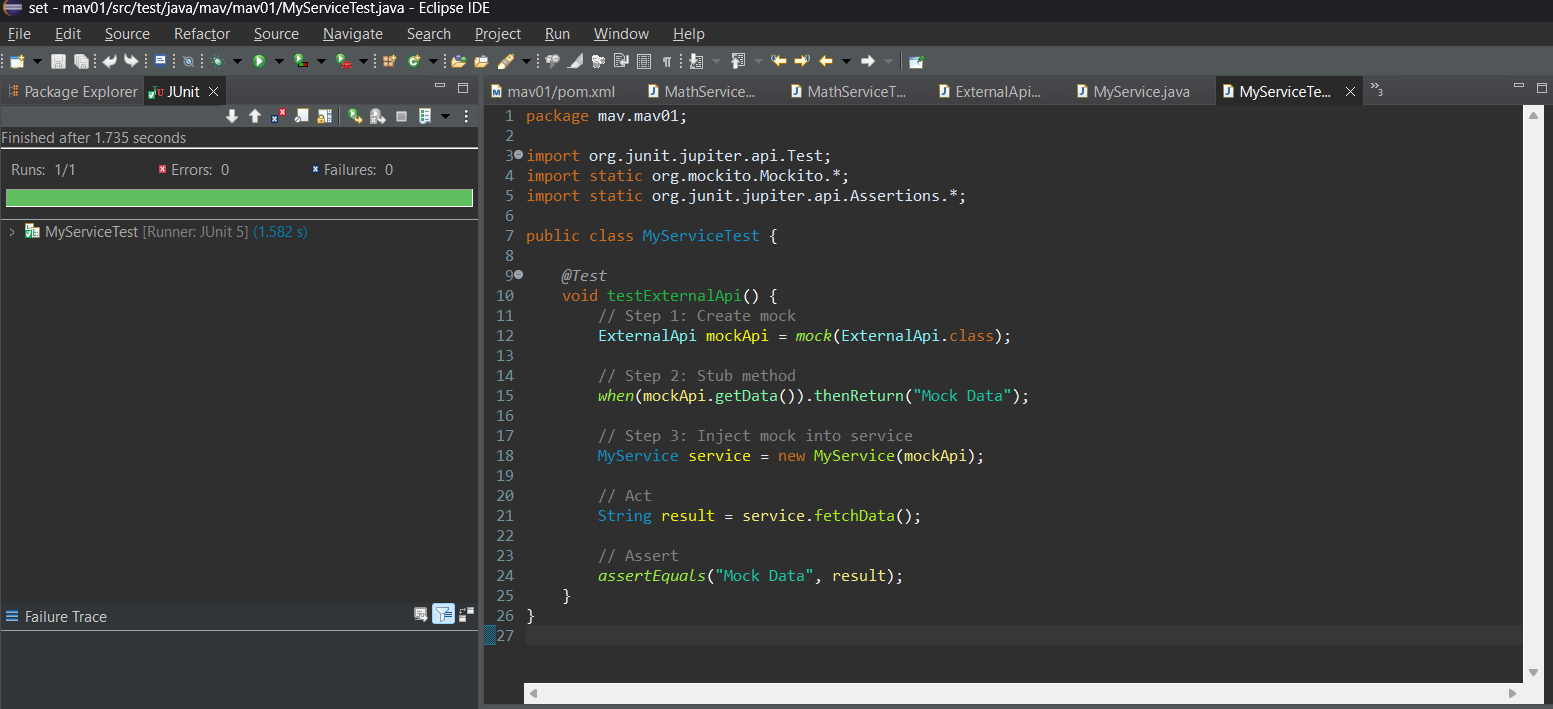
// Assert

assertEquals("Mock Data", result);

}

}

**OUTPUT:**



# Exercise 2: Verifying Interactions

**ExternalApi.java:**

package mav.mav01;

public interface ExternalApi {

String getData();

}

**MyService.java:**

package mav.mav01;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // Real call replaced by mock

}

}

**VerifiyInteractionTest.java:**

package mav.mav01;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class VerifyInteractionTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create a mock

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Inject into service

MyService service = new MyService(mockApi);

// Step 3: Call method that should invoke getData()

service.fetchData();

// Step 4: Verify that mockApi.getData() was called

verify(mockApi).getData();

}

}

**OUTPUT:**

