**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.

**Code:**

ExternalAPI.java

package com.cognizant.mockingandstubbing;

interface ExternalApi {

String getData();

}

MyService.java

package com.cognizant.mockingandstubbing;

class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // Calls the API

}

}

MyServiceTest.java

package com.cognizant.mockingandstubbing;

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

import static org.mockito.Mockito.when;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

import com.cognizant.mockingandstubbing.ExternalApi;

import com.cognizant.mockingandstubbing.MyService;

public class MyServiceTest {

@Test

public void testExternalApi() {

// Step 1: Create a mock object for the external API

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

// Step 2: Stub the method to return predefined data

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

// Step 3: Pass the mock to the service

MyService service = new MyService(mockApi);

// Act

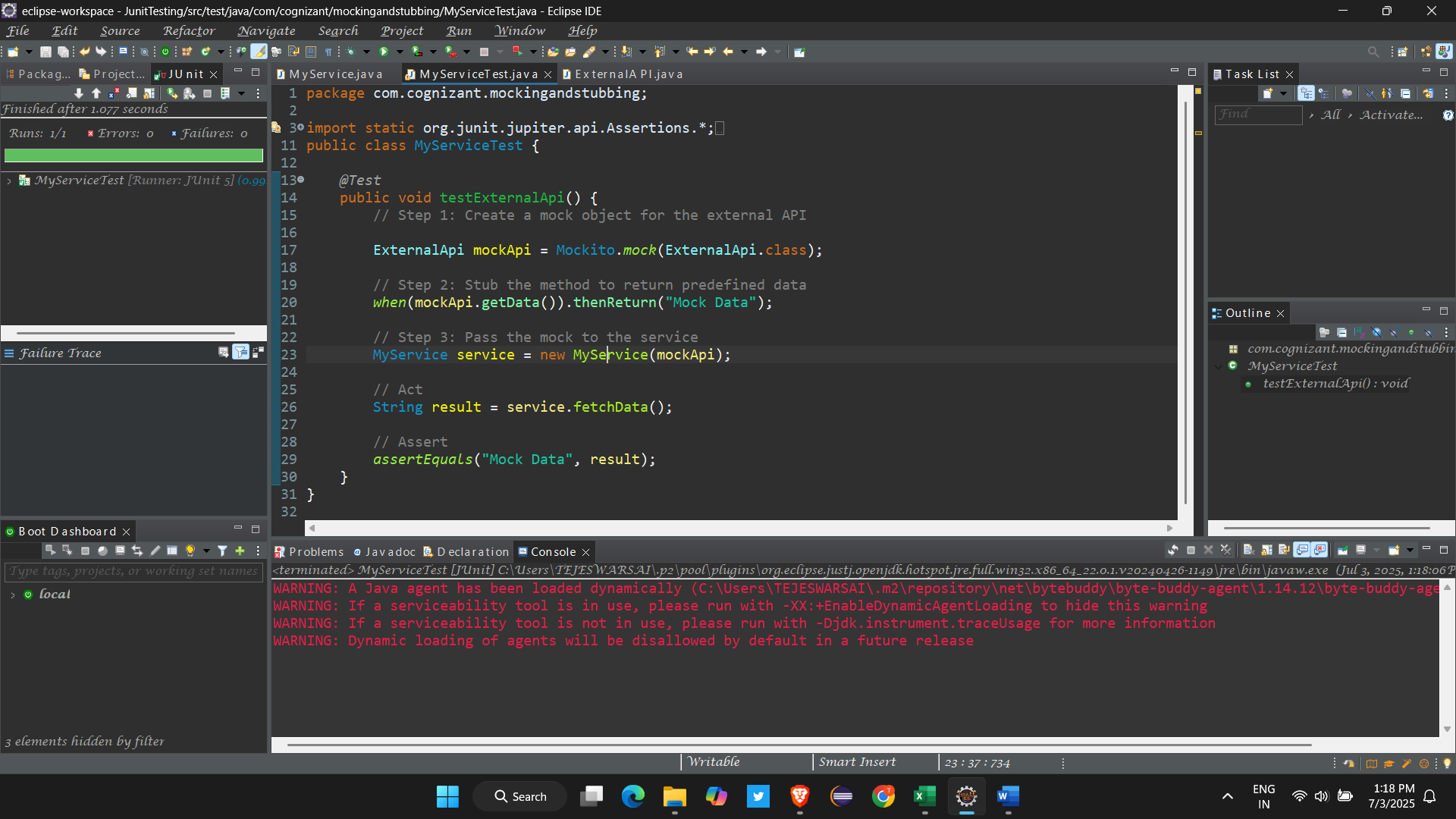
String result = service.fetchData();

// Assert

assertEquals("Mock Data", result);

}

}



**Exercise 2: Verifying Interactions**

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

**Code :**

package com.cognizant.mockingandstubbing;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class ServiceTest {

*@Test*

public void testVerifyInteraction() {

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

MyService service = new MyService(mockApi);

service.fetchData();

*verify*(mockApi).getData();

}

}

A computer screen shot of a computer screen

AI-generated content may be incorrect.