1. **Mocking and stubbing**

Dependency adding in pom.xml:

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.11.0</version>

<scope>test</scope>

</dependency>

ExternalApi.java:

package com.example;

public class ExternalApi {

public String getData() {

return "Real Data from API";

}

}

MyService.java:

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // Call the external API

}

}

MyserviceTest.java:

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testExternalApi() {

ExternalApi mockApi = mock(ExternalApi.class);

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

MyService service = new MyService(mockApi);

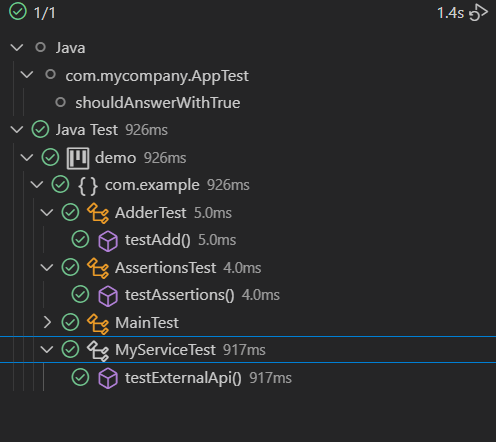
String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

Output:



1. **Verifying Interactions**

Keeping the **ExternalApi.java** and **MyService.java** files same

Adding an extra method in **MyServiceTest.java:**

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData(); // This should call mockApi.getData()

verify(mockApi).getData(); // Passes if getData() was called

}

Output:

