Verifying Interactions in JUnit with Mockito

**Project Structure**

MyMockTestProject/

├── pom.xml

└── src

├── main

│ └── java

│ └── org

│ └── example

│ ├── ExternalApi.java

│ └── MyService.java

└── test

└── java

└── org

└── example

└── MyServiceTest.java

**Dependencies (pom.xml)**

<dependencies>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.9.3</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>

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

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

<version>2.22.2</version>

<configuration>

<includes>

<include>\*\*/\*Test.java</include>

</includes>

</configuration>

</plugin>

</plugins>

</build>

**Java Classes and Test Code**

**🔸 ExternalApi.java**

package org.example;

public interface ExternalApi {

String getData();

}

**🔸 MyService.java**

*(Located in src/main/java/org/example/)*

package org.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData(); // interaction we're verifying

}

}

**🔹 MyServiceTest.java**

*(Located in src/test/java/org/example/)*

package org.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create a mock object

ExternalApi mockApi = mock(ExternalApi.class);

// Step 2: Inject mock into service

MyService service = new MyService(mockApi);

// Step 3: Call method that should trigger the interaction

service.fetchData();

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

verify(mockApi).getData();

}

}

Run the Test

Right-click MyServiceTest.java → **Run 'MyServiceTest'**

**OUTPUT**

