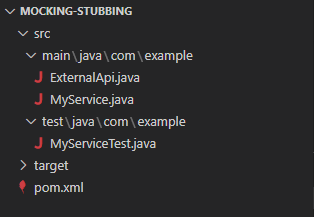
**Week 2**

**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.

**FILE STRUCTURE**

****

**CODE**

*src\main\java\com\example\ExternalApi.java*

package com.example;

public interface ExternalApi {

    String getData();

}

*src\main\java\com\example\MyService.java*

package com.example;

public class MyService {

    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData();

    }

}

*src\test\java\com\example\MyServiceTest.java*

package com.example;

public class MyService {

    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData();

    }

}

*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

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>JUnitMavenManual</artifactId>

    <version>1.0-SNAPSHOT</version>

    <dependencies>

    <dependency>

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

        <artifactId>junit-jupiter</artifactId>

        <version>5.10.0</version>

        <scope>test</scope>

    </dependency>

    <dependency>

        <groupId>org.mockito</groupId>

        <artifactId>mockito-core</artifactId>

        <version>5.12.0</version>

        <scope>test</scope>

    </dependency>

    </dependencies>

    <build>

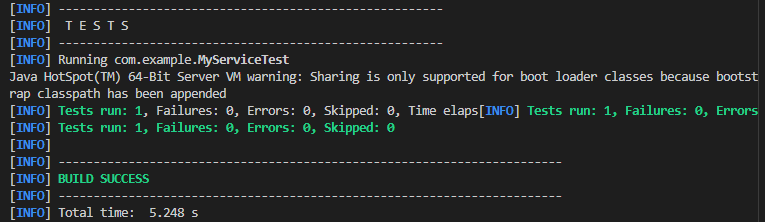
        <sourceDirectory>src/main/java</sourceDirectory>

        <testSourceDirectory>src/test/java</testSourceDirectory>

    </build>

</project>

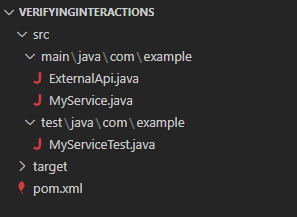
**OUTPUT**



**Exercise 2: Verifying Interactions**

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

**FILE STRUCTURE**

****

**CODE**

*src\main\java\com\example\ExternalApi.java*

package com.example;

public interface ExternalApi {

    String getData(String key);

}

*src\main\java\com\example\MyService.java*

package com.example;

public class MyService {

    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData("My-Service123");

    }

}

*src\test\java\com\example\MyServiceTest.java*

package com.example;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

    @Test

    public void testVerifyInteractionWithSpecificArgument() {

        //Create mock

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

        //Method call

        MyService service = new MyService(mockApi);

        service.fetchData();

        //Verify

        verify(mockApi).getData("My-Service123");

    }

}

*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

         http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>JUnitMavenManual</artifactId>

    <version>1.0-SNAPSHOT</version>

    <dependencies>

    <dependency>

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

        <artifactId>junit-jupiter</artifactId>

        <version>5.10.0</version>

        <scope>test</scope>

    </dependency>

    <dependency>

        <groupId>org.mockito</groupId>

        <artifactId>mockito-core</artifactId>

        <version>5.12.0</version>

        <scope>test</scope>

    </dependency>

    </dependencies>

    <build>

        <sourceDirectory>src/main/java</sourceDirectory>

        <testSourceDirectory>src/test/java</testSourceDirectory>

    </build>

</project>

**OUTPUT**

