**WEEK 2**

**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

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>mockito-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

ExternalApi.java

**package** com.example;

**public** **interface** ExternalApi {

String getData();

}

MyService.java

**package** com.example;

**public** **class** MyService {

**private** **final** ExternalApi externalApi;

**public** MyService(ExternalApi externalApi) {

**this**.externalApi = externalApi;

}

**public** String fetchData() {

**return** externalApi.getData();

}

}

MyServiceTest.java

**package** com.example;

**import** org.junit.jupiter.api.Test;

**import** **static** org.mockito.Mockito.\*;

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

**public** **class** MyServiceTest {

@Test

**public** **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);

// Step 4: Call service method

String result = service.fetchData();

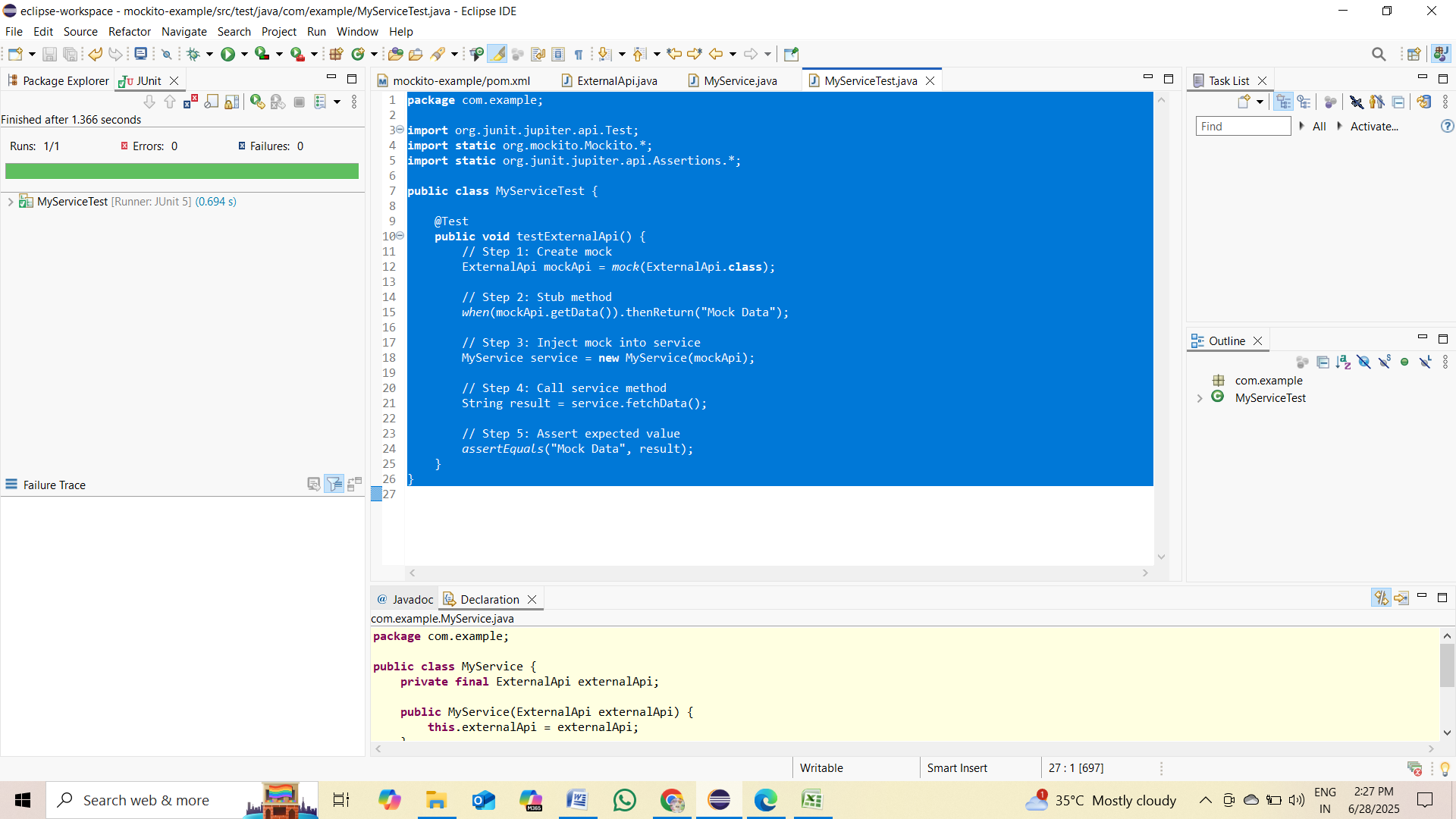
// Step 5: Assert expected value

*assertEquals*("Mock Data", result);

}

}

**OUTPUT**

****

**Exercise 2: Verifying Interactions**

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>mockito-example</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

ExternalApi.java

**package** com.example;

**public** **interface** ExternalApi {

String getData();

}

MyService.java

**package** com.example;

**public** **class** MyService {

**private** **final** ExternalApi externalApi;

**public** MyService(ExternalApi externalApi) {

**this**.externalApi = externalApi;

}

**public** String fetchData() {

**return** externalApi.getData();

}

}

MyServiceTest.java

**package** com.example;

**import** org.junit.jupiter.api.Test;

**import** **static** org.mockito.Mockito.\*;

**public** **class** MyServiceTest {

@Test

**public** **void** testVerifyInteraction() {

// Step 1: Create mock object

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

// Step 2: Use it in service

MyService service = **new** MyService(mockApi);

// Call method that should use the mock

service.fetchData();

// Step 3: Verify interaction

*verify*(mockApi).getData();

}

}

**OUTPUT**

