**WEEK-2**

**MODULE-4**

**MOCKITO EXERCISES**

**SUPERSET ID:6407550**

**Exercise 7: Handling Void Methods with Exceptions**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- JUnit 4 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>4.11.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**src/main/java**

**StorageHandler.java**

package com.example;

public interface StorageHandler {

void delete(String fileName);

}

**FileService.java**

package com.example;

public class FileService {

private StorageHandler handler;

public FileService(StorageHandler handler) {

this.handler = handler;

} package com.example;

import static org.mockito.Mockito.\*;

import org.junit.Test;

public class FileServiceTest {

@Test(expected = RuntimeException.class)

public void testVoidMethodThrowsException() {

// Step 1: Create mock

StorageHandler mockHandler = mock(StorageHandler.class);

// Step 2: Stub the void method to throw exception

doThrow(new RuntimeException("Delete failed"))

.when(mockHandler).delete("data.txt");

// Step 3: Inject and call method

FileService service = new FileService(mockHandler);

service.deleteFile("data.txt");

// Step 4: Verify the interaction (optional here)

verify(mockHandler).delete("data.txt");

}

}

public void deleteFile(String fileName) {

handler.delete(fileName);

}

}

**src/test/java**

**FileServiceTest.java**

package com.example;

import static org.mockito.Mockito.\*;

import org.junit.Test;

public class FileServiceTest {

@Test(expected = RuntimeException.class)

public void testVoidMethodThrowsException() {

// Step 1: Create mock

StorageHandler mockHandler = mock(StorageHandler.class);

// Step 2: Stub the void method to throw exception

doThrow(new RuntimeException("Delete failed"))

.when(mockHandler).delete("data.txt");

// Step 3: Inject and call method

FileService service = new FileService(mockHandler);

service.deleteFile("data.txt");

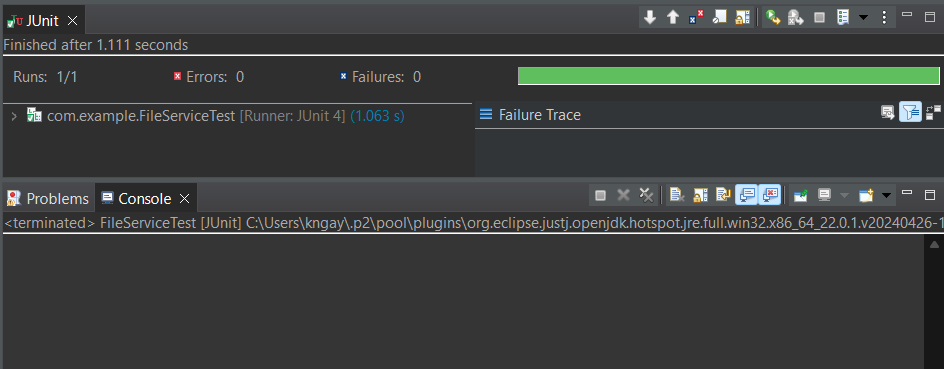
// Step 4: Verify the interaction (optional here)

verify(mockHandler).delete("data.txt");

}

}

**OUTPUT:**

****