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

**ExternalApi.java**

package mypackage;

public interface ExternalApi {

String getData();

void sendMessage(String message);

void logAction(String action);

void connect

void disconnect();

}

**MyService.java**

package mypackage;

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public void performLoggingSafely() {

try {

api.logAction("SAVE\_EVENT");

} catch (RuntimeException ex) {

System.out.println("Exception caught while logging: " + ex.getMessage());

}

}

}

**MyServiceTest.java**

package mypackage;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

public class MyServiceTest {

@Test

public void testVoidMethodThrowsException() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

doThrow(new RuntimeException("Logging Failed"))

.when(mockApi)

.logAction("SAVE\_EVENT");

service.performLoggingSafely();

System.out.println("Exception was successfully simulated for void method logAction().");

verify(mockApi).logAction("SAVE\_EVENT");

}

}



