**Exercise 1: Mocking Databases and Repositories**

**Code:**

package week2.module4\_3\_2;

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

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

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

import org.junit.jupiter.api.Test;

class Repository{

String getData() {

return "Data";

}

}

class RepositoryService{

private Repository repository;

public RepositoryService(Repository repository) {

this.repository=repository;

}

String processData() {

return "Processed "+repository.getData();

}

}

public class Ex1ServiceTest {

*@Test*

public void testProcessData() {

Repository respository=*mock*(Repository.class);

RepositoryService service=new RepositoryService(respository);

*when*(respository.getData()).thenReturn("Mock Data");

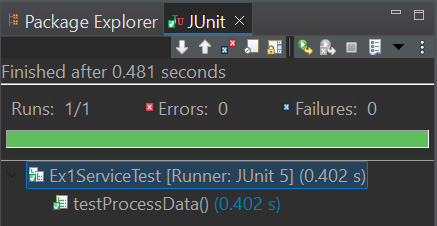
String result=service.processData();

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

}

}

**Output:**

****

**Exercise 2: Mocking External Services (RESTful APIs)**

**Code:**

package week2.module4\_3\_2;

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

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

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

import org.junit.jupiter.api.Test;

class RestApi{

String fetchData() {

return "Data";

}

}

class RestApiService{

private RestApi restApi;

public RestApiService(RestApi restApi) {

this.restApi=restApi;

}

String processData() {

return "Processed "+restApi.fetchData();

}

}

public class Ex2RestApiTest {

*@Test*

public void testProcessData() {

RestApi respository=*mock*(RestApi.class);

RestApiService service=new RestApiService(respository);

*when*(respository.fetchData()).thenReturn("Mock Fetched Data");

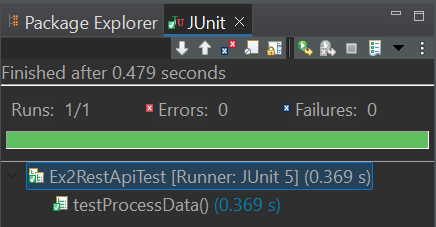
String result=service.processData();

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

}

}

**Output:**



**Exercise 3: Mocking File I/O**

**Code:**

package week2.module4\_3\_2;

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

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

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

import org.junit.jupiter.api.Test;

class FileReader{

String read() {

return "File Content";

}

}

class FileWriter{

void write(String content) {

}

}

class FileService{

private FileReader fileReader;

private FileWriter fileWriter;

public FileService(FileReader fileReader,FileWriter fileWriter) {

this.fileReader=fileReader;

this.fileWriter=fileWriter;

}

void processService() {

String content=fileReader.read();

String processedContent="Processed "+content;

fileWriter.write(processedContent);

}

}

public class Ex3FileServiceTest {

*@Test*

public void testProcessService() {

FileReader fileReader=*mock*(FileReader.class);

FileWriter fileWriter=*mock*(FileWriter.class);

FileService service=new FileService(fileReader,fileWriter);

*when*(fileReader.read()).thenReturn("Mock File Data");

service.processService();

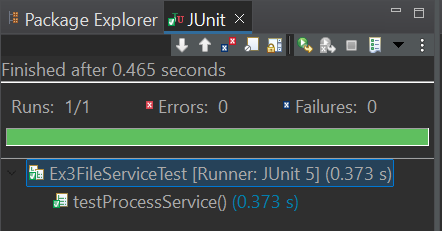
*verify*(fileReader).read();

*verify*(fileWriter).write("Processed Mock File Data");

}

}

**Output:**

****

**Exercise 4: Mocking Network Interactions**

**Code:**

package week2.module4\_3\_2;

import static org.junit.Assert.*assertEquals*;

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

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

import org.junit.jupiter.api.Test;

class NetworkClient{

String connect() {

return "Connection Success";

}

}

class NetworkService{

private NetworkClient networkClient;

public NetworkService(NetworkClient networkClient) {

this.networkClient=networkClient;

}

String processService() {

return networkClient.connect();

}

}

public class Ex4NetworkTest {

*@Test*

public void testProcessService() {

NetworkClient networkClient=*mock*(NetworkClient.class);

*when*(networkClient.connect()).thenReturn("Mock Connection Success");

NetworkService service=new NetworkService(networkClient);

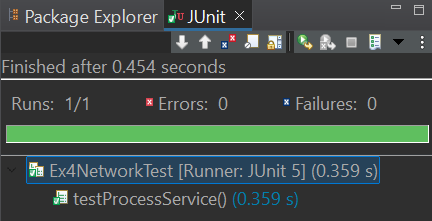
String result=service.processService();

*assertEquals*("Mock Connection Success",result);

}

}

**Output:**

****

**Exercise 5: Mocking Multiple Return Values**

**Code:**

package week2.module4\_3\_2;

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

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

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

import org.junit.jupiter.api.Test;

class Repository1{

String getData() {

return "Data";

}

}

class Service1{

private Repository1 repository;

public Service1(Repository1 repository) {

this.repository=repository;

}

String service() {

return "Processed "+repository.getData();

}

}

public class Ex5MultiReturnTest {

*@Test*

public void testService() {

Repository1 repository=*mock*(Repository1.class);

Service1 service=new Service1(repository);

*when*(repository.getData()).thenReturn("First Mock Data").thenReturn("Second Mock Data");

String result1=service.service();

String result2=service.service();

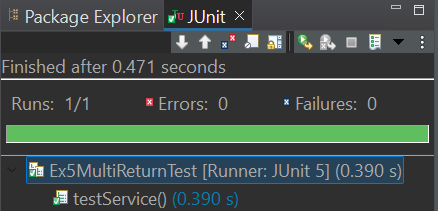
*assertEquals*("Processed First Mock Data",result1);

*assertEquals*("Processed Second Mock Data",result2);

}

}

**Output:**

****