

**Solution**

**Mockito/pml.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>Mockito</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- JUnit 5 -->

<dependency>

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

<artifactId>junit-jupiter</artifactId>

<version>5.9.2</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.2.0</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M5</version>

</plugin>

</plugins>

</build>

</project>

**ExternalApi.java**

package com.example;

public interface ExternalApi {

String getData();

void save(String data);

String getStatus();

String fetchById(int id);

}

**MyService.java**

package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

public void saveData(String data) {

api.save(data);

}

public String getStatus() {

return api.getStatus();

}

public String fetchById(int id) {

return api.fetchById(id);

}

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testMockingAndStubbing() {

ExternalApi mockApi = mock(ExternalApi.class);

when(mockApi.getData()).thenReturn("Mock Data");

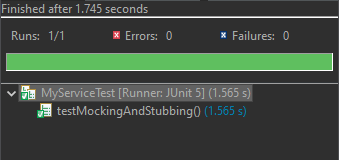
MyService service = new MyService(mockApi);

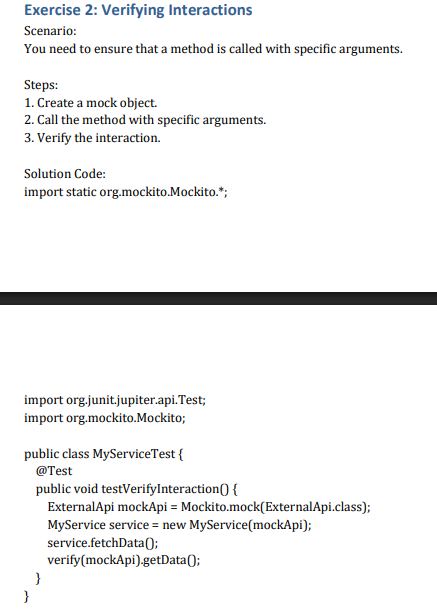
assertEquals("Mock Data", service.fetchData());

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

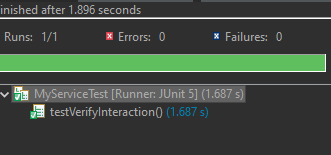
service.fetchData();

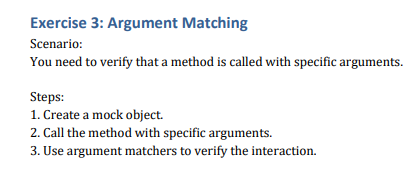
verify(mockApi).getData();

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testArgumentMatching() {

ExternalApi mockApi = mock(ExternalApi.class);

when(mockApi.fetchById(anyInt())).thenReturn("Result");

MyService service = new MyService(mockApi);

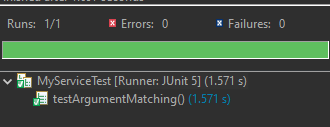
assertEquals("Result", service.fetchById(42));

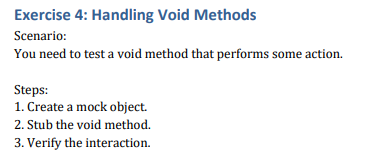
verify(mockApi).fetchById(eq(42));

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testVoidMethod() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

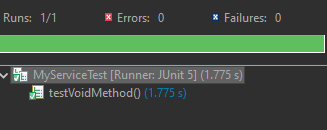
service.saveData("Data");

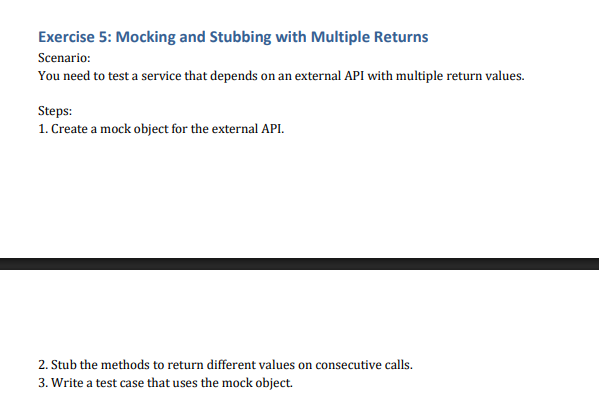
verify(mockApi).save("Data");

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testMultipleReturns() {

ExternalApi mockApi = mock(ExternalApi.class);

when(mockApi.getStatus()).thenReturn("Loading", "Ready");

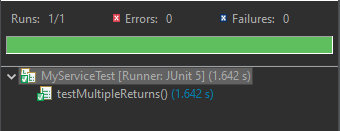
assertEquals("Loading", mockApi.getStatus());

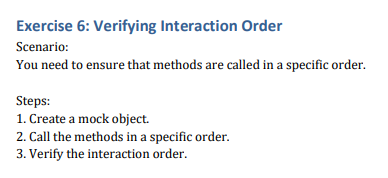
assertEquals("Ready", mockApi.getStatus());

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testInteractionOrder() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

service.fetchData();

service.saveData("Done");

InOrder order = inOrder(mockApi);

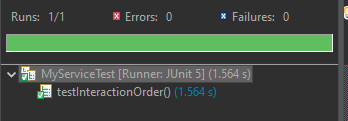
order.verify(mockApi).getData();

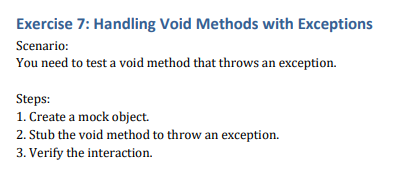
order.verify(mockApi).save("Done");

}

}

**Output:**





**Solution:**

package com.example;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

import org.mockito.InOrder;

public class MyServiceTest {

@Test

public void testVoidMethodWithException() {

ExternalApi mockApi = mock(ExternalApi.class);

doThrow(new RuntimeException("Error")).when(mockApi).save(anyString());

final MyService service = new MyService(mockApi);

assertThrows(RuntimeException.class, () -> service.saveData("Test"));

}

}

**Output:**

