**ExternalApi.java**

public interface ExternalApi {

String getData();

void performAction();

}

**MyService.java**

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

public void doAction() {

api.performAction();

}}

**MyServiceTest.java**

import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

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

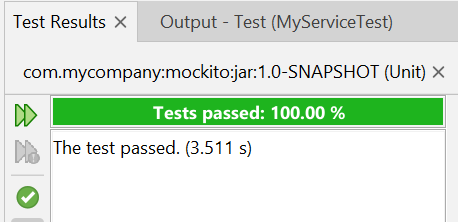
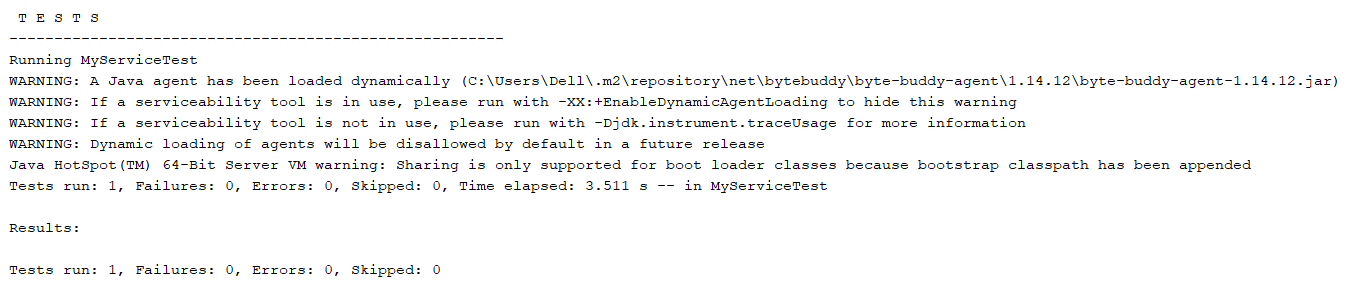
when(mockApi.getData()).thenReturn("Mock Data");

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Data", result);

}}



import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

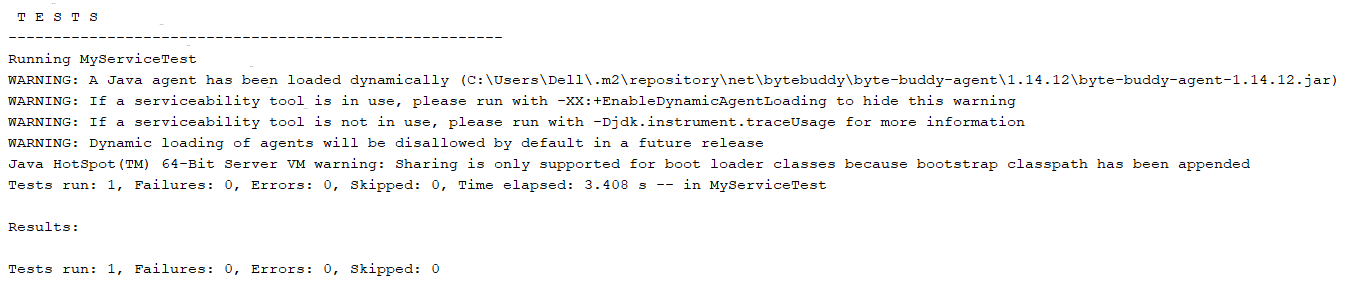
ExternalApi mockApi = Mockito.mock(ExternalApi.class);

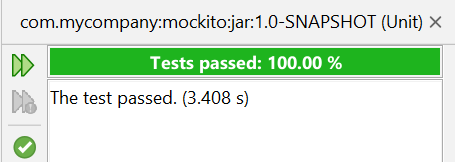
MyService service = new MyService(mockApi);

service.fetchData();

verify(mockApi).getData();

}}





**ExternalApi.java:**

public interface ExternalApi {

String getData();

public void sendMessage(String eq);

}

**MyService.java:**

public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

public void sendWelcomeMessage(String name) {

api.sendMessage(name);

}}

**MyServiceTest.java:**

import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testArgumentMatching() {

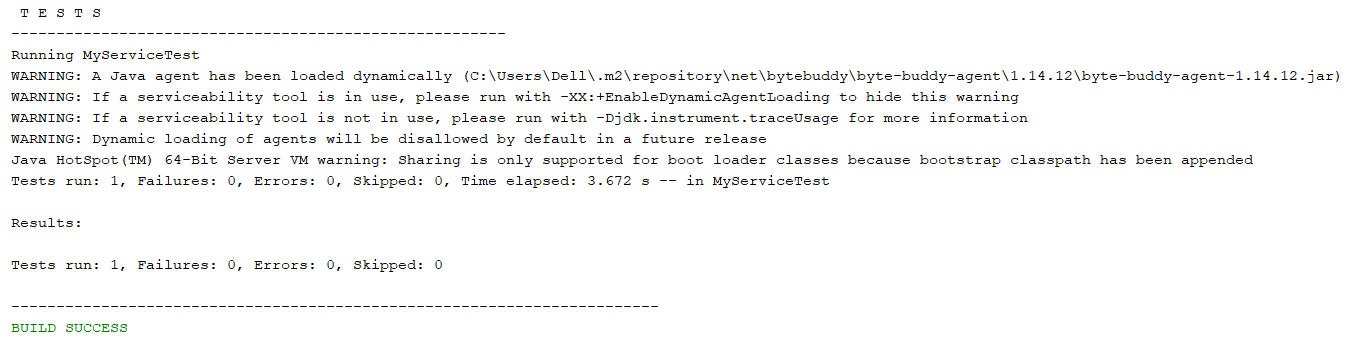
ExternalApi mockApi = Mockito.mock(ExternalApi.class);

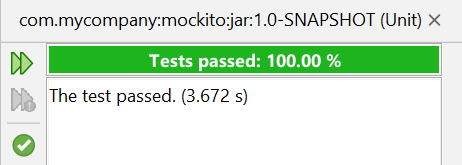
MyService service = new MyService(mockApi);

service.sendWelcomeMessage("Swarna");

verify(mockApi).sendMessage(eq("Swarna"));

}}





**MyServiceTest.java:**

import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testPerformActionCalled() {

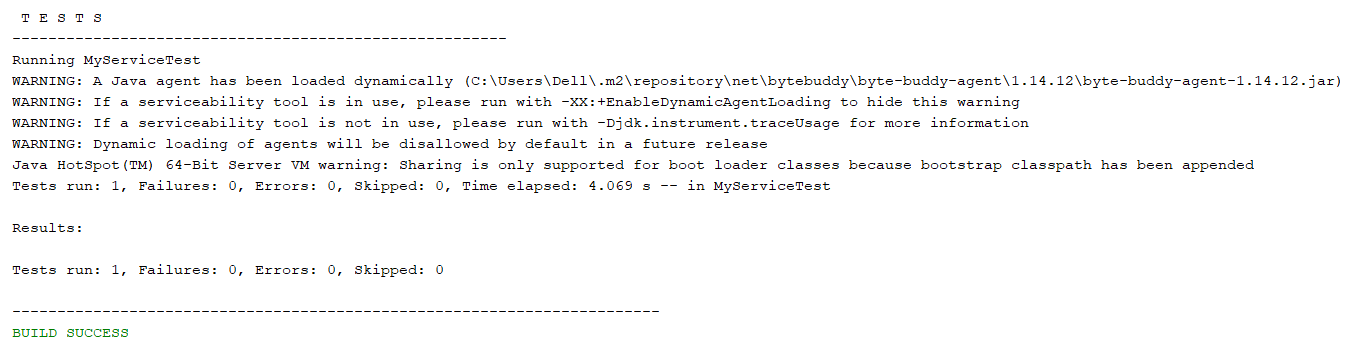
ExternalApi mockApi = Mockito.mock(ExternalApi.class);

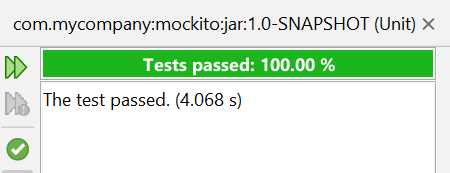
MyService service = new MyService(mockApi);

service.doAction();

verify(mockApi).performAction();

}}





import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testMultipleReturns() {

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

when(mockApi.getData())

.thenReturn("First", "Second", "Third");

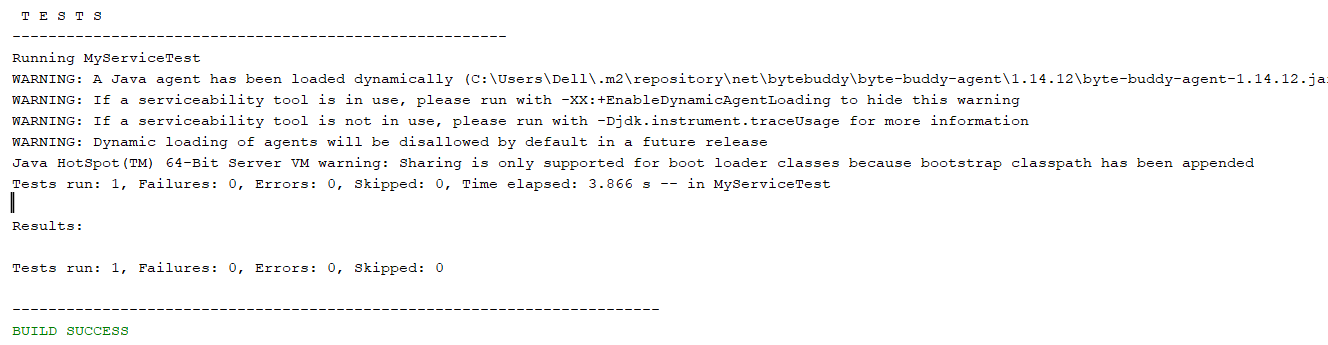
assertEquals("First", mockApi.getData());

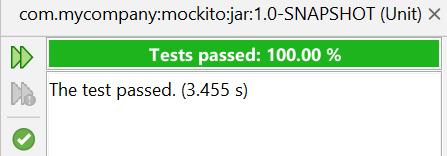
assertEquals("Second", mockApi.getData());

assertEquals("Third", mockApi.getData());

}

}





public class MyService {

private final ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

public void doAction() {

api.performAction();

}

public void sendWelcomeMessage(String name) {

System.out.println("Hi, "+name);

}

public void process() {

api.getData();

api.performAction();

} }

import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.InOrder;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testMethodCallOrder() {

ExternalApi mockApi = mock(ExternalApi.class);

MyService service = new MyService(mockApi);

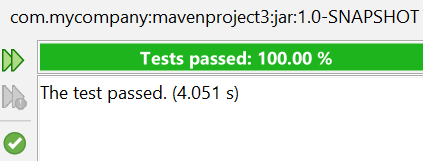
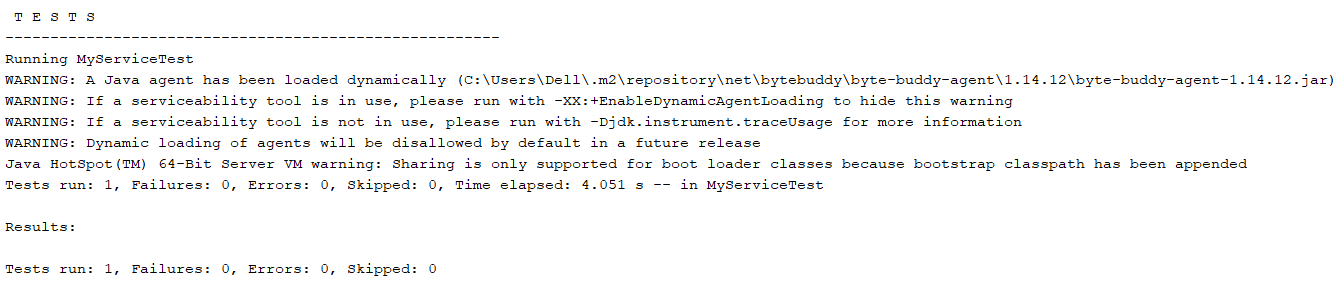
service.process();

InOrder inOrder = inOrder(mockApi);

inOrder.verify(mockApi).getData();

inOrder.verify(mockApi).performAction();

}}



import com.mycompany.mavenproject3.ExternalApi;

import com.mycompany.mavenproject3.MyService;

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

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.InOrder;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVoidMethodThrowsException() {

ExternalApi mockApi = mock(ExternalApi.class);

doThrow(new RuntimeException("API failed"))

.when(mockApi).performAction();

MyService service = new MyService(mockApi);

try {

service.process();

} catch (RuntimeException e) {

assert(e.getMessage().equals("API failed"));

}

verify(mockApi).performAction();

}}

