Mocking and Stubbig:  
  
**external api:**  
package com.example;

public interface ExternalApi {

String getData();

}  
  
**service:**  
package com.example;

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}  
  
**Service test:**  
package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class MyServiceTest {

@Test

public void testExternalApi() {

ExternalApi mockApi = mock(ExternalApi.class);

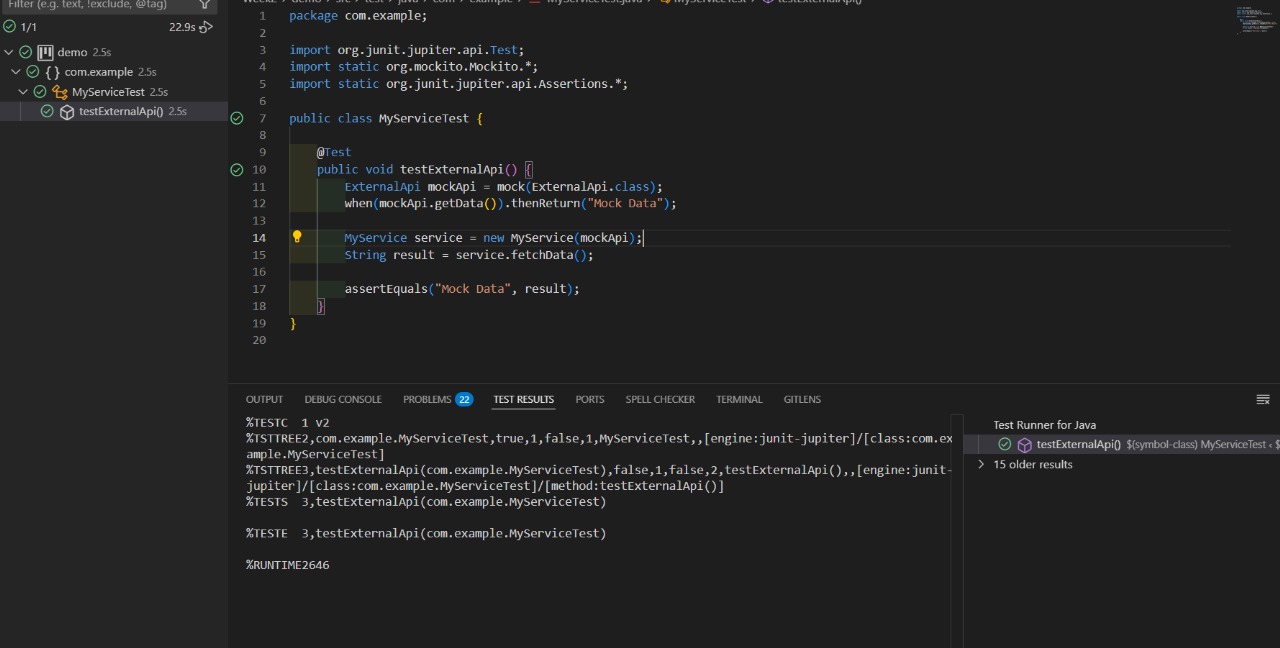
when(mockApi.getData()).thenReturn("Mock Data");

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}  
  
output:  


Verifying Interaction:

**Externalapi:**

package com.example;  
public interface ExternalApi {

String getData();

}  
  
**service:**

package com.example;  
public class MyService {

private final ExternalApi externalApi;

public MyService(ExternalApi externalApi) {

this.externalApi = externalApi;

}

public void fetchData() {

String data = externalApi.getData(); // Interaction to be verified

System.out.println("Fetched: " + data);

}

}  
  
**test code:**  
  
import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create mock object

ExternalApi mockApi = mock(ExternalApi.class);

// Optional: stub behavior (not required for verification)

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

// Step 2: Inject mock into service and call method

MyService service = new MyService(mockApi);

service.fetchData();

// Step 3: Verify interaction

verify(mockApi).getData();

}

}  
 **output:**

