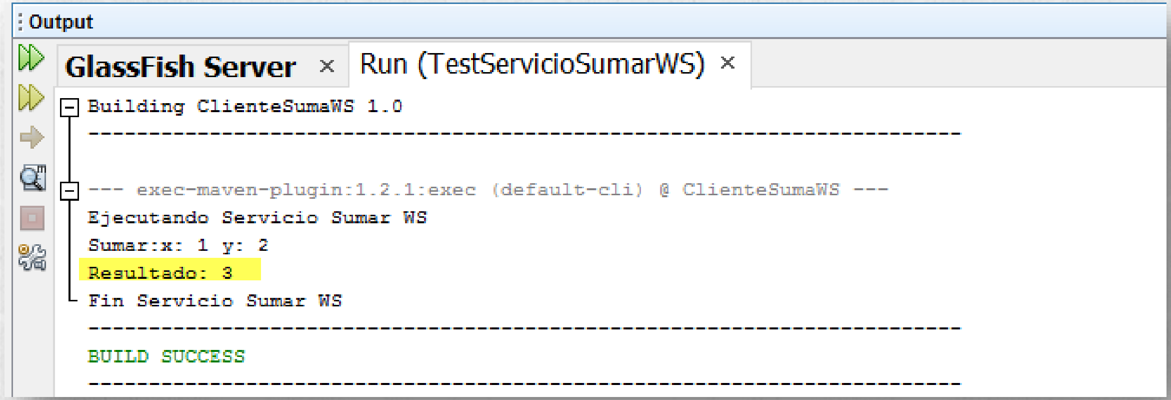
JAVA EE COURSE

HELLO WORLD WEBSERVICE JAX-WS CLIENT

By the expert: Eng. Ubaldo Acosta

EXERCISE OBJECTIVE

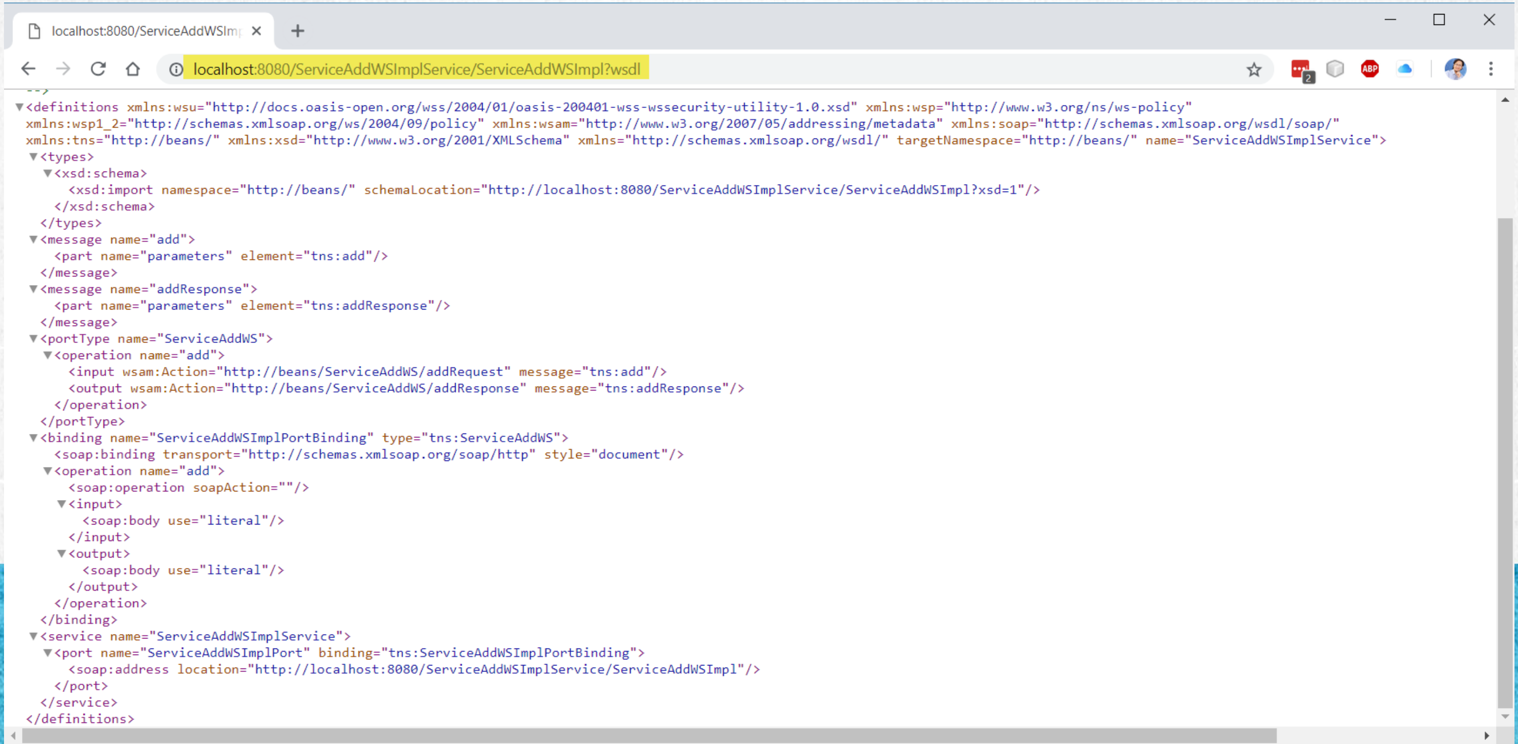
The objective of the exercise is to create a HelloWorld Web Service Client created in the previous exercise. The result is shown below:



```
Building ClienteSumaWS 1.0
------------------------------------------------------------------------

--- exec-maven-plugin:1.2.1:exec (default-cli) @ ClienteSumaWS ---
Ejecutando Servicio Sumar WS
Sumar:x: 1 y: 2
Resultado: 3
Fin Servicio Sumar WS
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

START FROM THE PREVIOUS EXERCISE

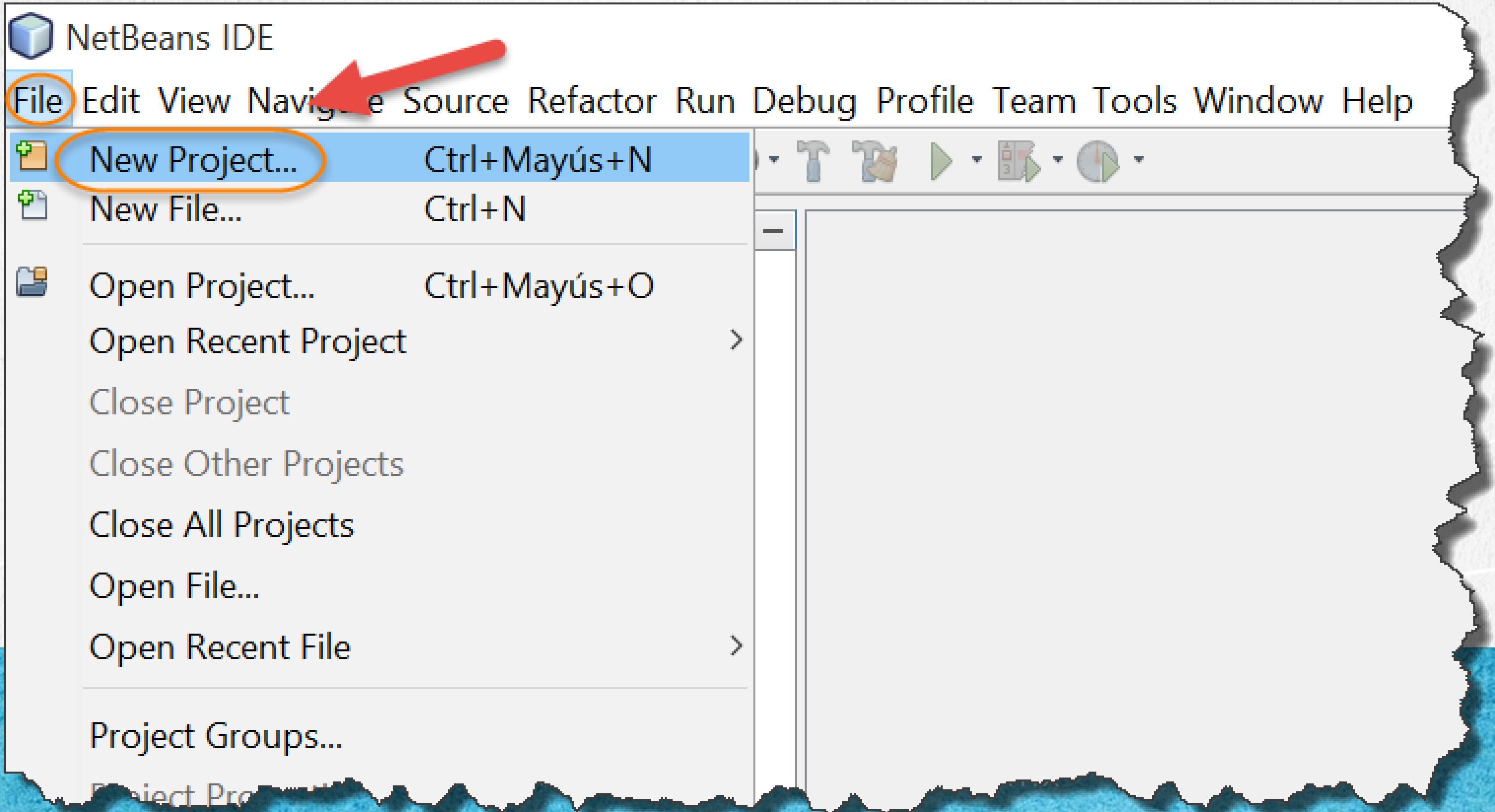
We will start from the previous year. It should already be deployed in Glassfish and working. We must be able to visualize the following url: http://localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl?wsdl


localhost:8080/ServiceAddWSIm
localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl?wsdl

```
-->
▼<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://beans/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://beans/" name="ServiceAddWSImplService">
 ▼<types>
   ▼<xsd:schema>
      <xsd:import namespace="http://beans/" schemaLocation="http://localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl?xsd=1"/>
    </xsd:schema>
  </types>
 ▼<message name="add">
    <part name="parameters" element="tns:add"/>
  </message>
 ▼<message name="addResponse">
    <part name="parameters" element="tns:addResponse"/>
  </message>
 ▼<portType name="ServiceAddWS">
   ▼<operation name="add">
      <input wsam:Action="http://beans/ServiceAddWS/addRequest" message="tns:add"/>
      <output wsam:Action="http://beans/ServiceAddWS/addResponse" message="tns:addResponse"/>
    </operation>
  </portType>
 ▼<binding name="ServiceAddWSImplPortBinding" type="tns:ServiceAddWS">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
   ▼<operation name="add">
      <soap:operation soapAction=""/>
     ▼<input>
        <soap:body use="literal"/>
      </input>
     ▼<output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
 ▼<service name="ServiceAddWSImplService">
   ▼<port name="ServiceAddWSImplPort" binding="tns:ServiceAddWSImplPortBinding">
      <soap:address location="http://localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl"/>
    </port>
  </service>
</definitions>
```

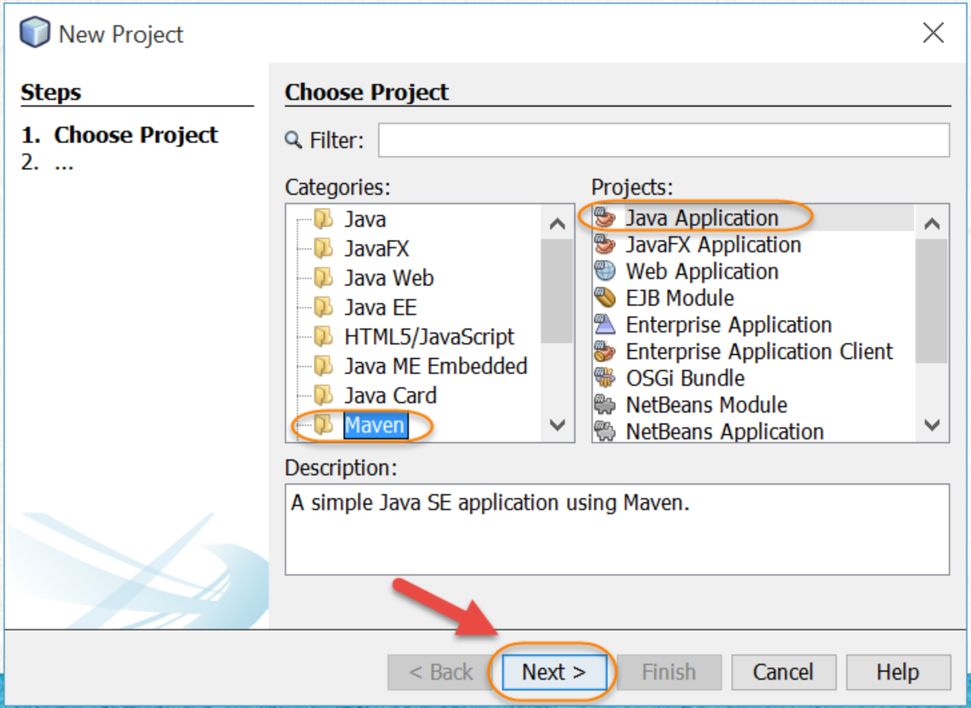
1. CREATE A NEW PROJECT

We create the ClientAddWS project:



1. CREATE A NEW PROJECT

We create the ClientAddWS project as a maven project:



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Filter:

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- HTML5/JavaScript
- Java ME Embedded
- Java Card
- Maven

Projects:

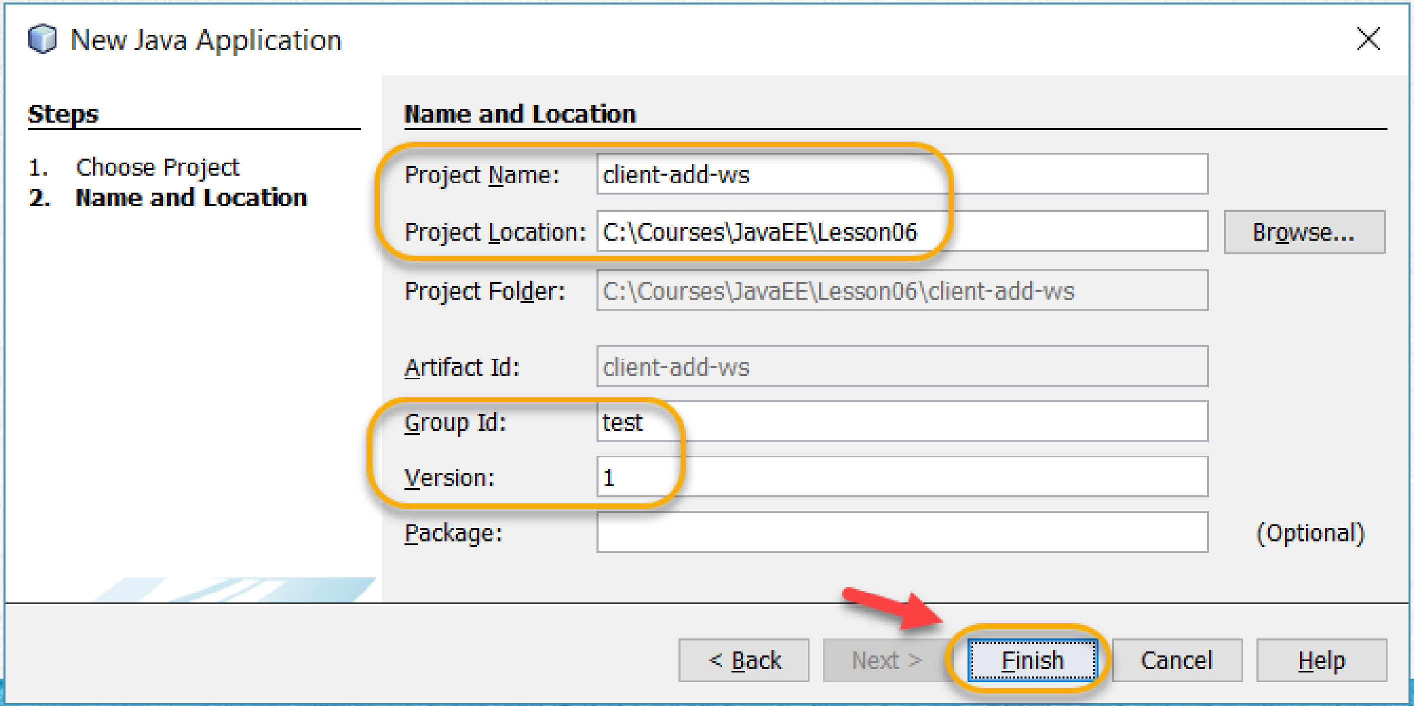
- Java Application
- JavaFX Application
- Web Application
- EJB Module
- Enterprise Application
- Enterprise Application Client
- OSGi Bundle
- NetBeans Module
- NetBeans Application

Description:

A simple Java SE application using Maven.

1. CREATE A NEW PROJECT

We create the ClientAddWS project as a maven project:


New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:	client-add-ws	
Project Location:	C:\Courses\JavaEE\Lesson06	Browse...
Project Folder:	C:\Courses\JavaEE\Lesson06\client-add-ws	
Artifact Id:	client-add-ws	
Group Id:	test	
Version:	1	
Package:		(Optional)

2. MODIFY THE CODE

pom.xml:

Click to download

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>test</groupId>
    <artifactId>client-add-ws</artifactId>
    <version>1</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>8.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
```

2. MODIFY THE CODE

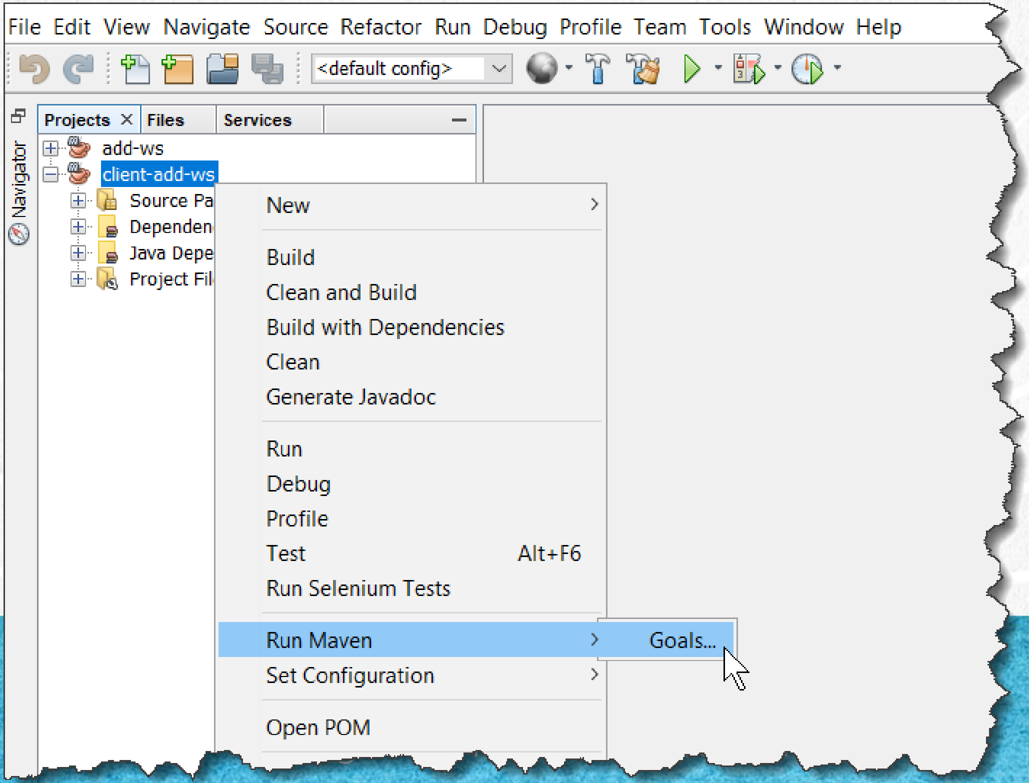
pom.xml:

Click to download

```
    <build>
        <plugins>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>jaxws-maven-plugin</artifactId>
                <version>2.5</version>
                <configuration>
                    <vmArgs>
                        <vmArg>-Djavax.xml.accessExternalSchema=all</vmArg>
                    </vmArgs>
                </configuration>
                <executions>
                    <execution>
                        <goals>
                            <goal>wsimport</goal>
                        </goals>
                        <configuration>
                            <wsdlUrls>
                                <wsdlUrl>http://localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl?wsdl</wsdlUrl>
                            </wsdlUrls>
                            <packageName>wsclient</packageName>
                            <sourceDestDir>${basedir}/src/main/java</sourceDestDir>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</project>
```

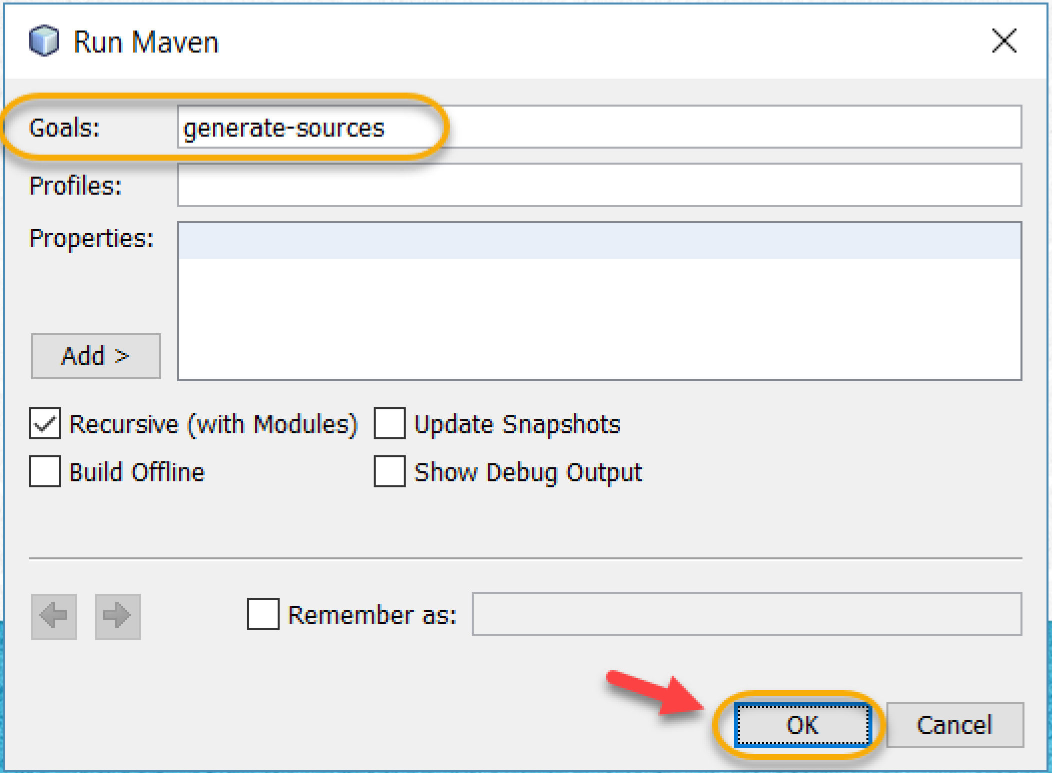
3. Generate the WS Client Code

We execute the client to generate the Web Service code by means of the wsimport command added to the pom.xml file:



3. Generate the WS Client Code

We execute the client to generate the Web Service code. We write **generate-sources** in the field of Goals to generate the client code:


Run Maven

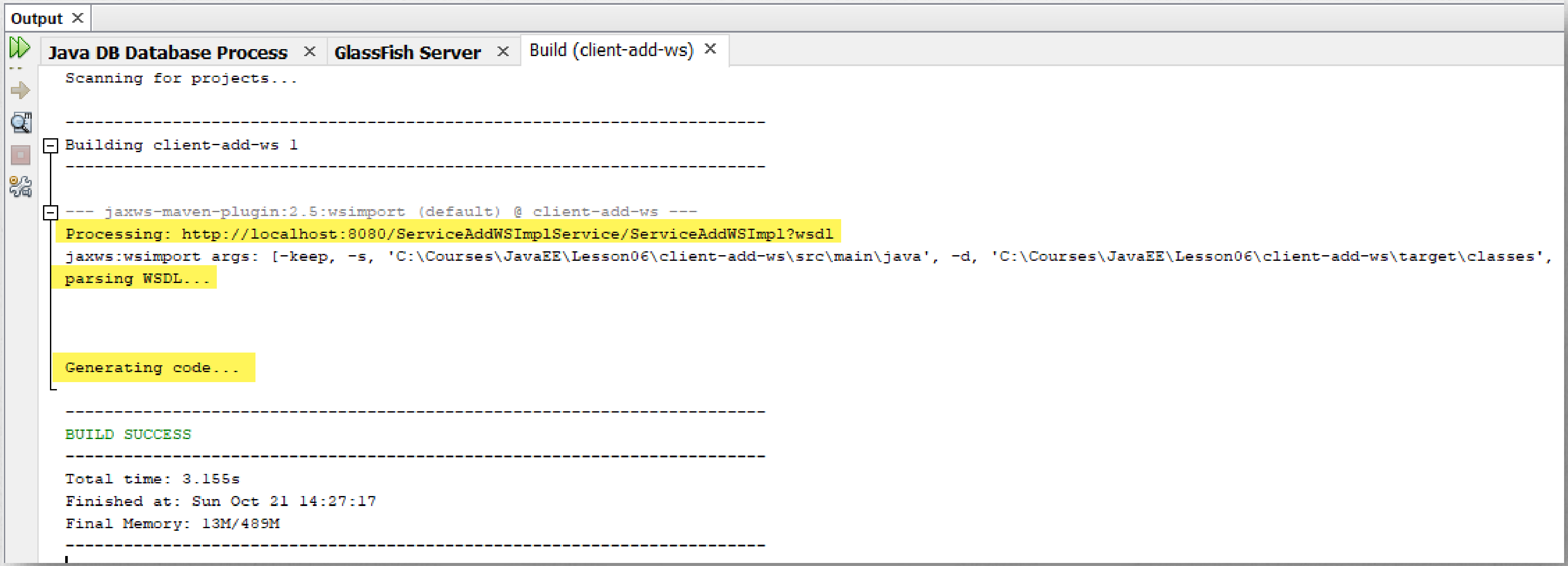
Goals: generate-sources

Profiles:

Properties:

3. GENERATE THE WS CLIENT CODE

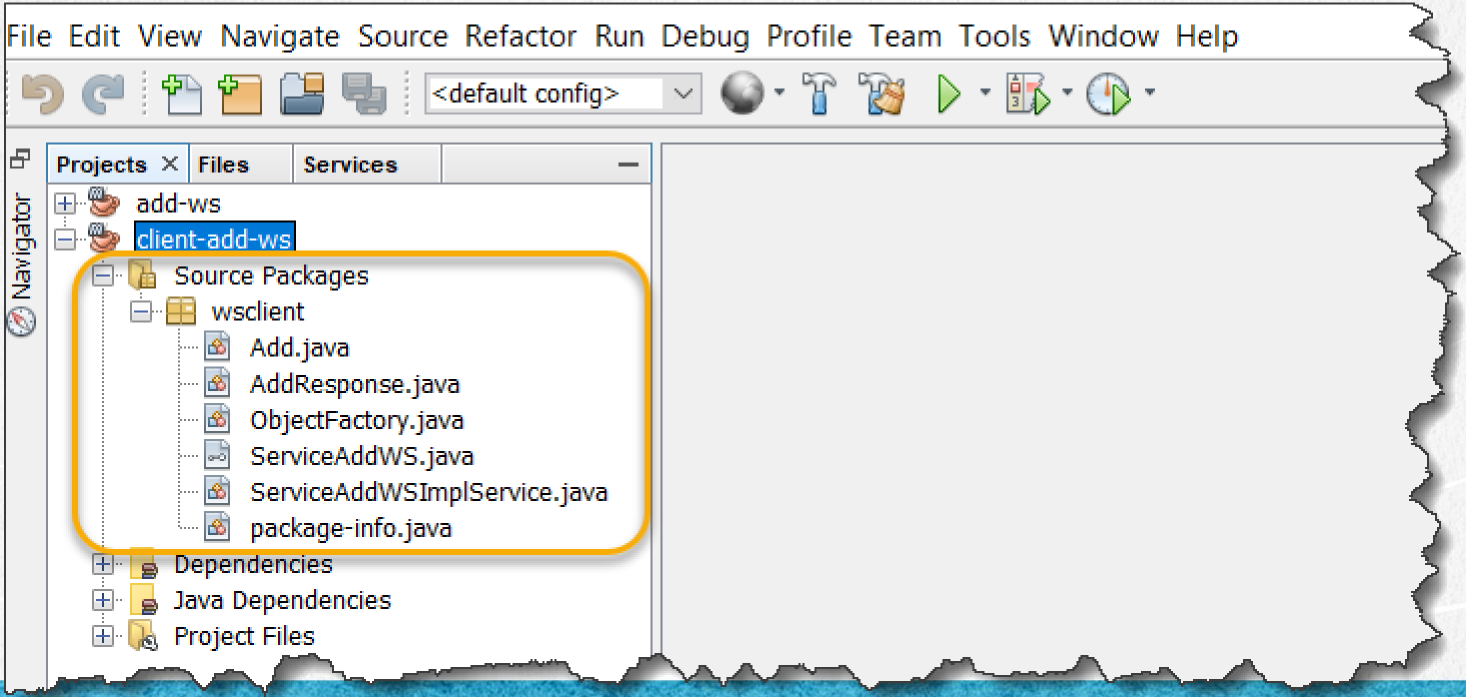
We should see an output similar to this one:



```
Scanning for projects...

------------------------------------------------------------------------
Building client-add-ws 1
------------------------------------------------------------------------

--- jaxws-maven-plugin:2.5:wsimport (default) @ client-add-ws ---
Processing: http://localhost:8080/ServiceAddWSImplService/ServiceAddWSImpl?wsdl
jaxws:wsimport args: [-keep, -s, 'C:\Courses\JavaEE\Lesson06\client-add-ws\src\main\java', -d, 'C:\Courses\JavaEE\Lesson06\client-add-ws\target\classes',
parsing WSDL...



Generating code...

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 3.155s
Finished at: Sun Oct 21 14:27:17
Final Memory: 13M/489M
------------------------------------------------------------------------
```

3. GENERATE THE WS CLIENT CODE

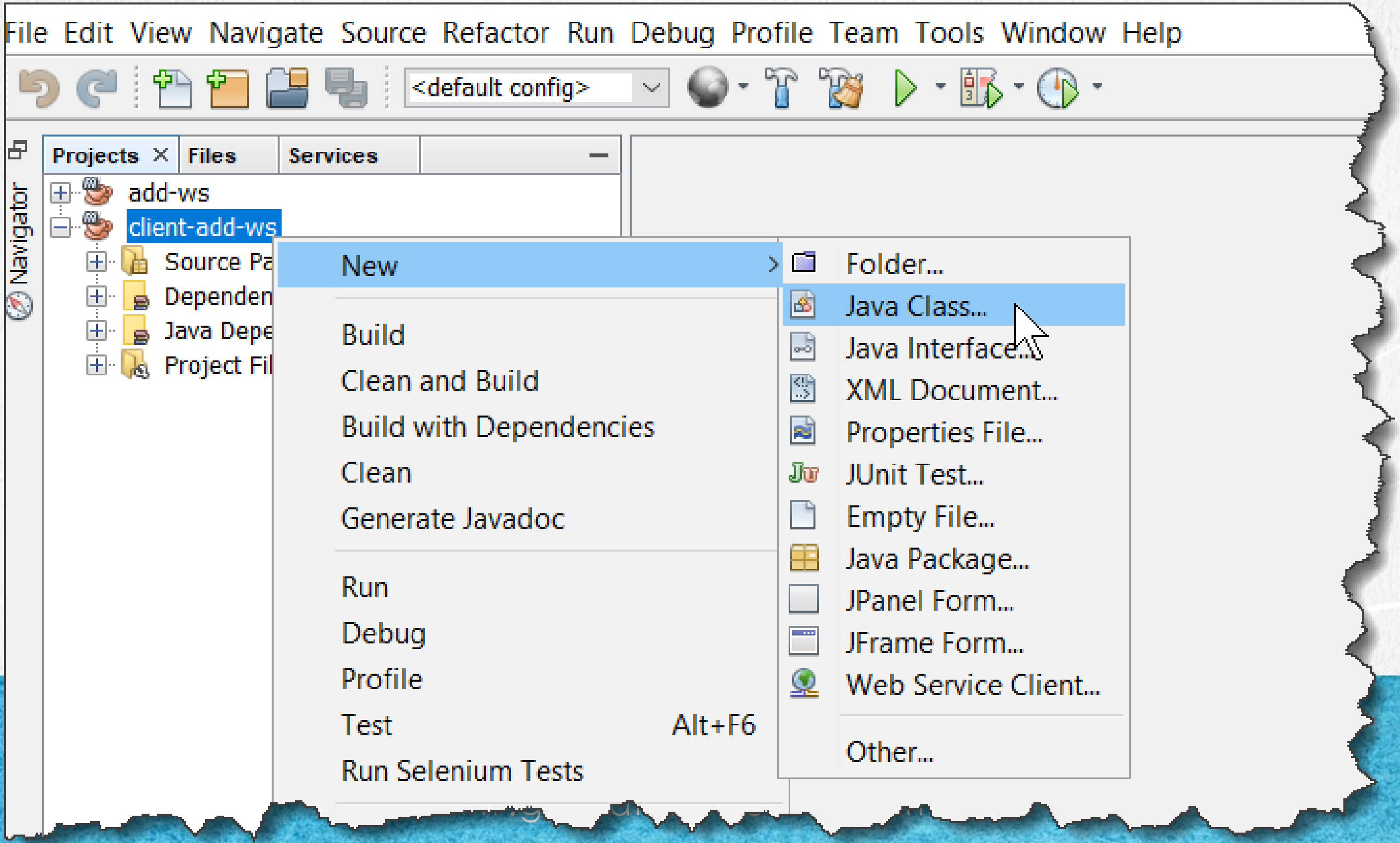
We must see the code generated in the project:


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects
Files
Services
Navigator

```
add-ws
client-add-ws
  Source Packages
    wsclient
      Add.java
      AddResponse.java
      ObjectFactory.java
      ServiceAddWS.java
      ServiceAddWSImplService.java
      package-info.java
  Dependencies
  Java Dependencies
  Project Files
```

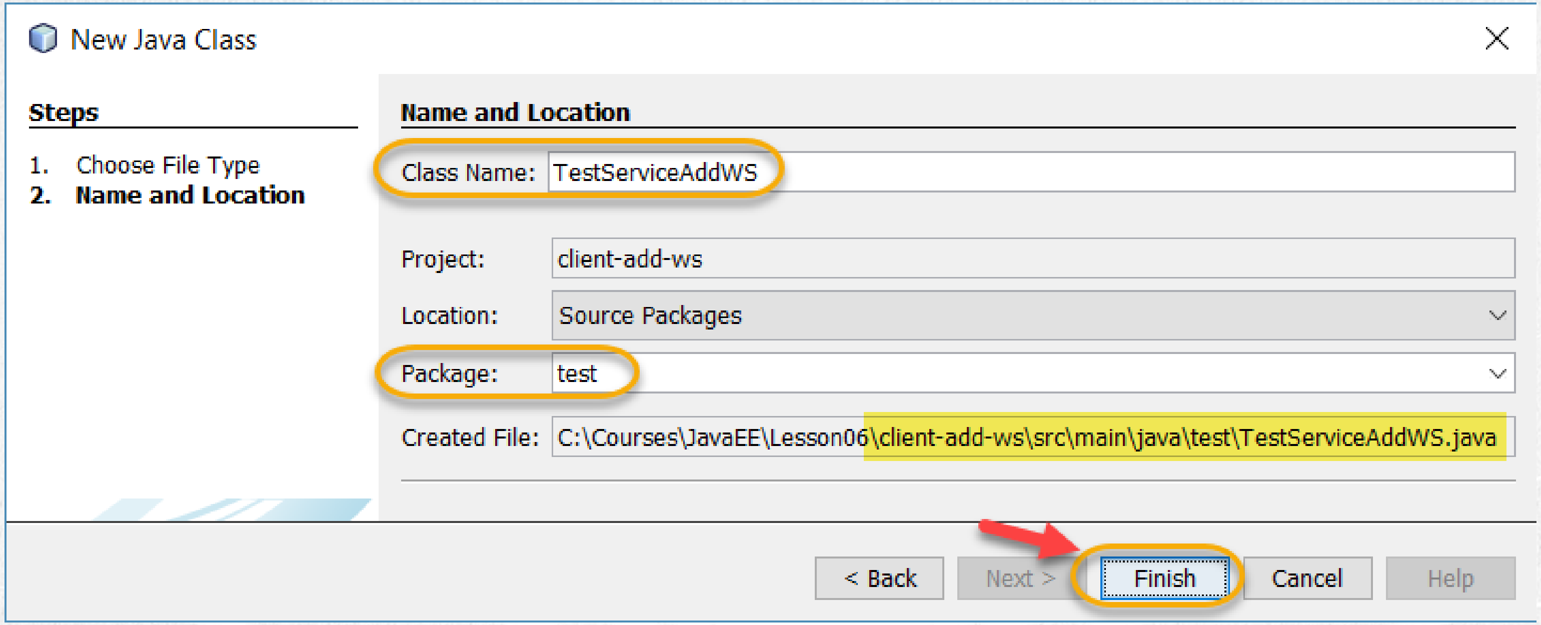
4. CREATE A JAVA CLASS

We create the class TestServiceAddWS.java:



4. CREATE A JAVA CLASS

We create the class TestServiceAddWS.java:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: TestServiceAddWS

Project: client-add-ws

Location: Source Packages

Package: test

Created File: C:\Courses\JavaEE\Lesson06\client-add-ws\src\main\java\test\TestServiceAddWS.java

5. MODIFY THE FILE

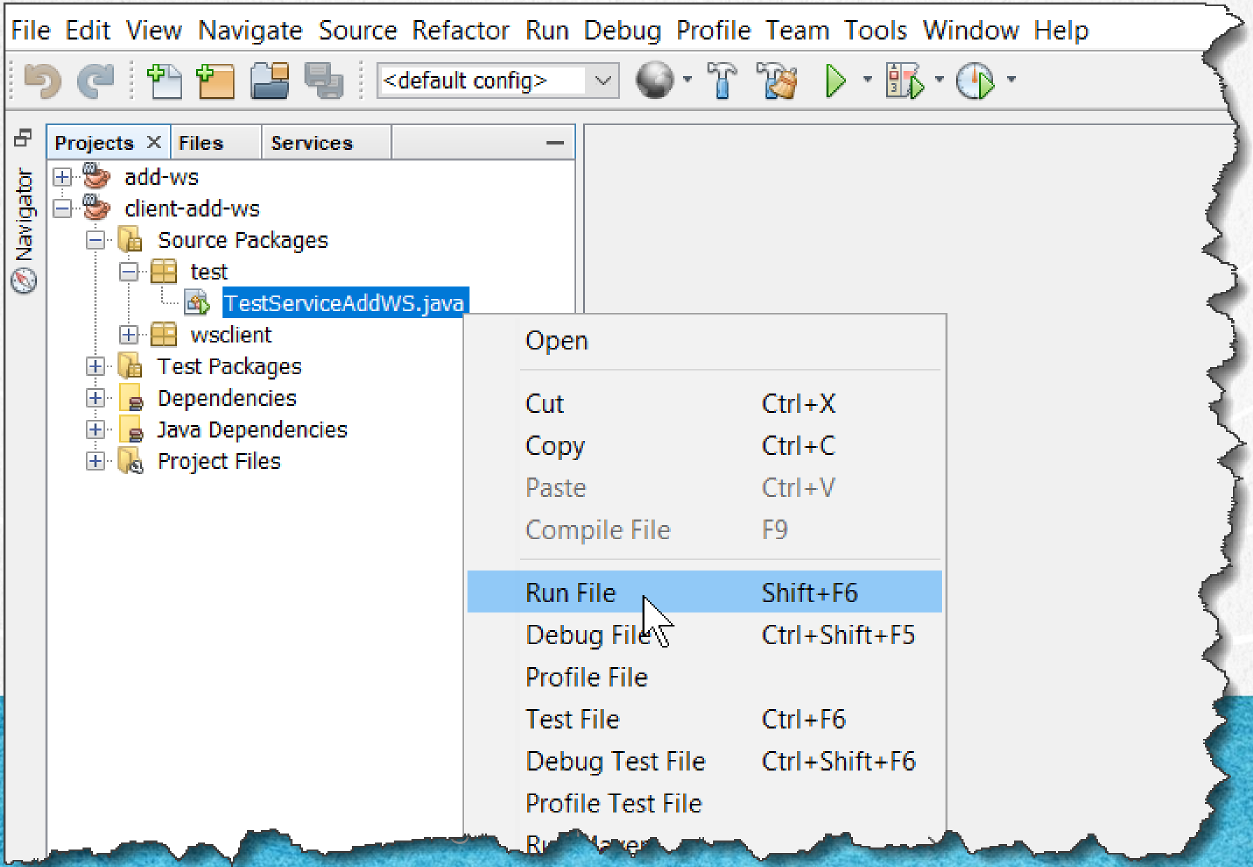
TestServiceAddWS.java:

Click to download

```java
package test;

import wsclient.ServiceAddWS;
import wsclient.ServiceAddWSImplService;

public class TestServiceAddWS {

    public static void main(String[] args) {
        ServiceAddWS addWSService = new ServiceAddWSImplService().getServiceAddWSImplPort();
        System.out.println("Running WS Add Service");
        int x = 1;
        int y = 2;
        System.out.println("Add:" + "x: " + x + " y: " + y);
        System.out.println("Result: " + addWSService.add(x, y));
        System.out.println("End of Add Service WS");
    }
}
```

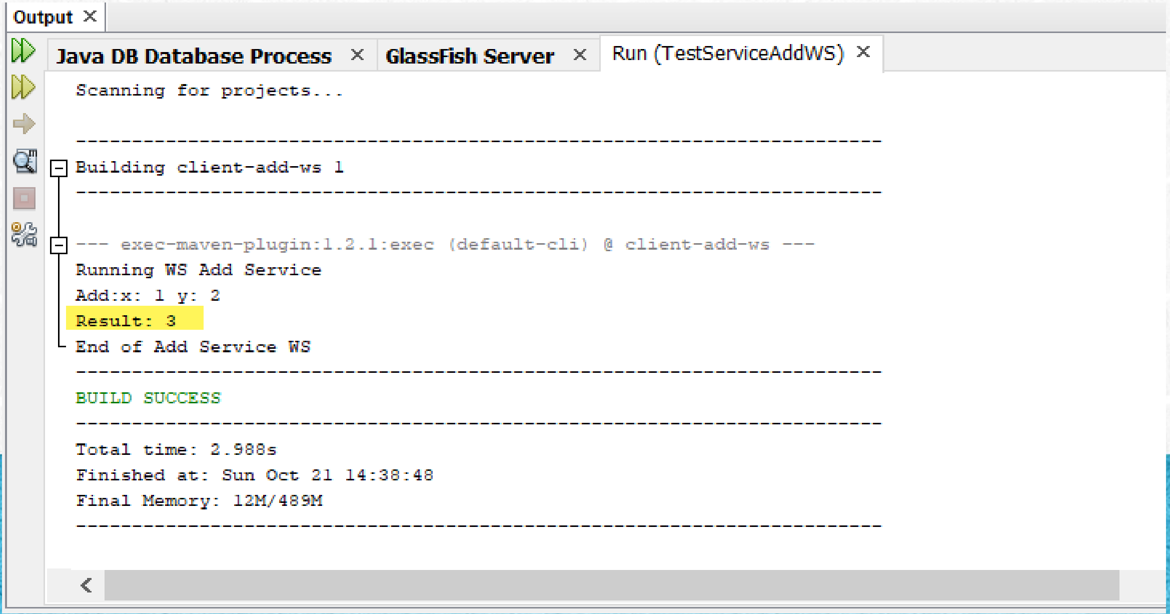
6. Execute the application

Execute the class:



6. EXECUTE THE APPLICATION

We see the result of consulting the Web service. As we have already mentioned, the Web Service of the previous exercise must be deployed, since this client consumes the add-on web service, and the response of the call to the requested Web Service is received.



```
Scanning for projects...

------------------------------------------------------------------------
Building client-add-ws 1
------------------------------------------------------------------------

--- exec-maven-plugin:1.2.1:exec (default-cli) @ client-add-ws ---
Running WS Add Service
Add:x: 1 y: 2
Result: 3
End of Add Service WS
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 2.988s
Finished at: Sun Oct 21 14:38:48
Final Memory: 12M/489M
------------------------------------------------------------------------
```

EXERCISE CONCLUSION

With this exercise we created a client to be able to consume the Add Web service created and published in the previous exercise.

This is the same procedure that can be followed for web services published when we have available the url of the web service wsdl.

ONLINE COURSE

JAVA EE JAKARTA EE

By: Eng. Ubaldo Acosta

JAVA EE COURSE

www.globalmentoring.com.mx