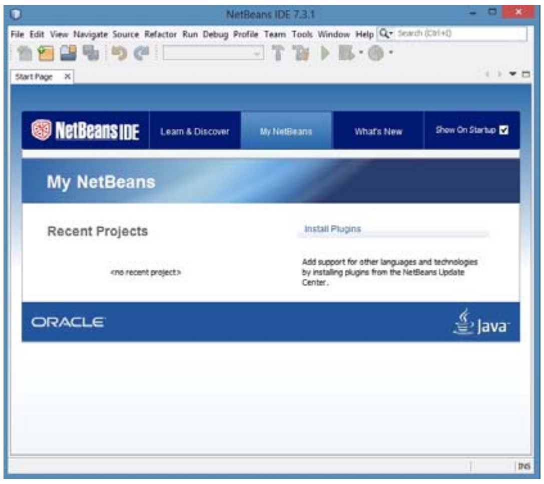
SOAP Web Service in Java (JAX-WS)

In this tutorial, we will show you how to develop a simple SOAP based Web Service in Java using **JAX-WS**, called as "**CalculatorService**" in NetBeans (7.3). In order to demonstrate development of this application we begin with:

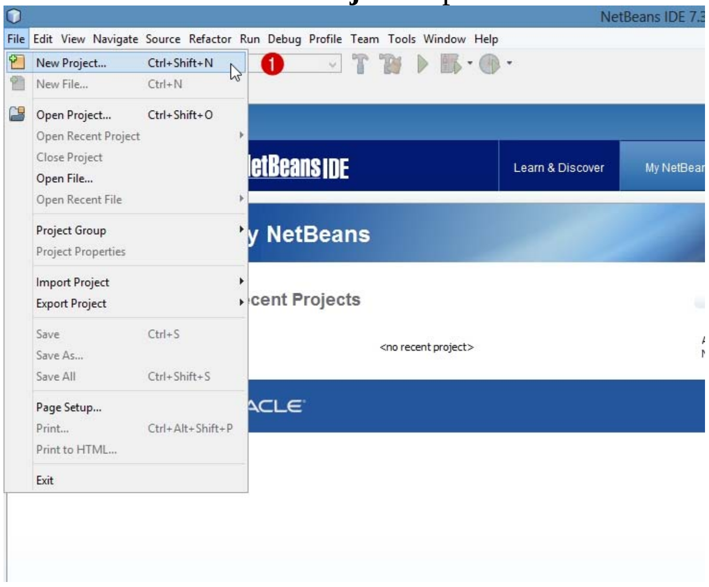
1. Create a Web Application named as "**Calculator**" in NetBeans.
2. Creating a SOAP Web Service called as "**CalculatorService**"
3. Creating a simple operation called as "sum".
4. Deploy and Test the Web Service.

1- Create a Web Application named as "Calculator" in NetBeans.

Step 1: Open **NetBeans IDE** (See fig below)


NetBeans IDE 7.3.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Start Page
NetBeans IDE
Learn & Discover
My NetBeans
What's New
Show On Startup
My NetBeans
Recent Projects
<no recent project>
Install Plugins
Add support for other languages and technologies by installing plugins from the NetBeans Update Center.
ORACLE
Java

and Select in the menu bar **File ---> New Project** or press **Ctrl + Shift + N.** (See fig below)

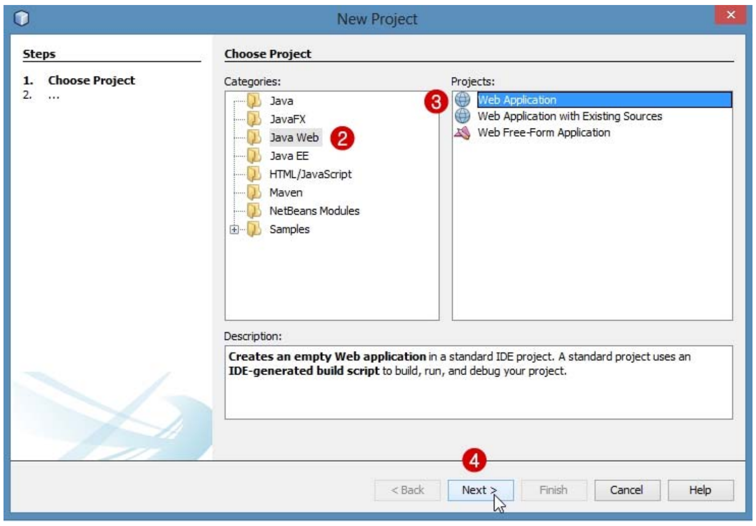


New Project dialog box gets open.

Step 2 : Under **Categories:** select **Java Web**.

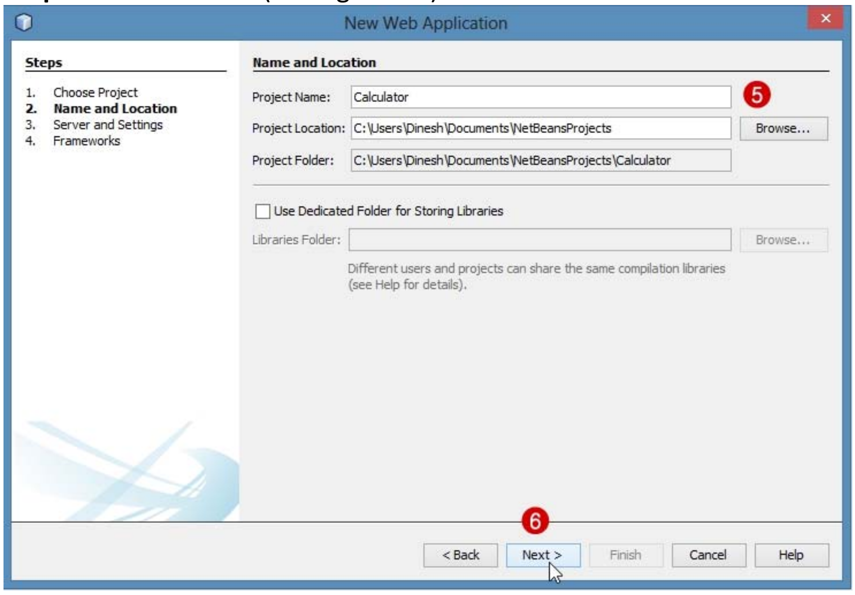
Step 3 : Under **Projects:** select **Web Application.**

Step 4 : Click **Next >** . (See fig below)



Description:
Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

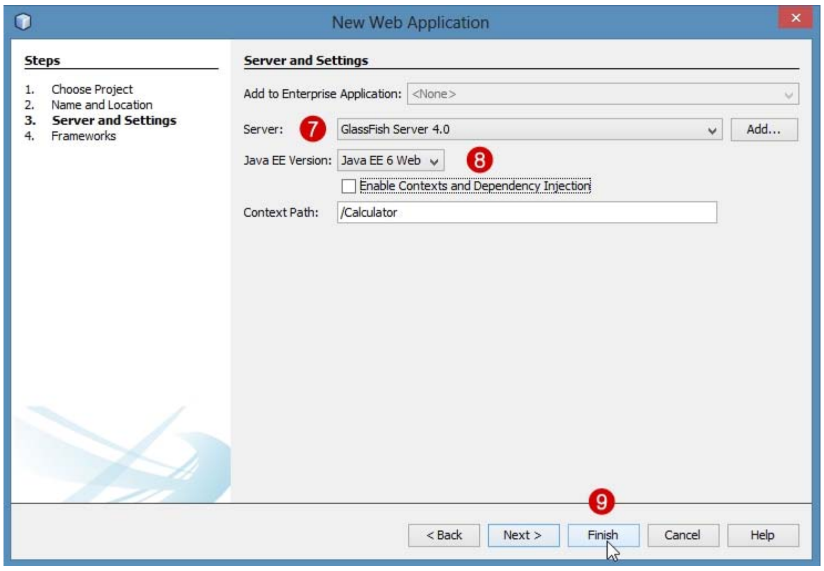
New Web Application dialog box gets open.
Step 5 : Under **Name and Location** tab, enter **Project Name:** as "**Calculator**".
Step 6 : Click **Next >** . (See fig below)



Project Name: Calculator
Project Location: C:\Users\Dinesh\Documents\NetBeansProjects
Project Folder: C:\Users\Dinesh\Documents\NetBeansProjects\Calculator
☐ Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).

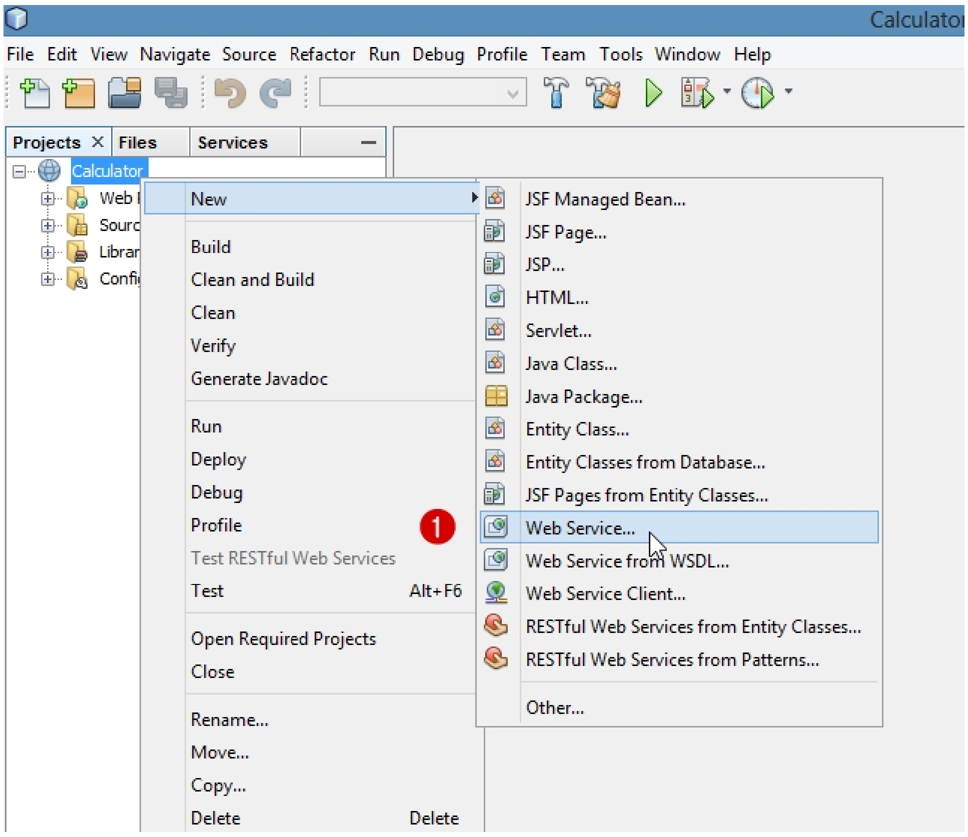
Under the same **New Web Application, Server and Settings** dialog box gets open.

Step 7 : Choose **Server:** as "**GlassFish Server 4.0**". You can also choose "**GlassFish v3 Domain**" as server if you are using old version of NetBeans other than 7.3.1, which comes bundled with "**GlassFish v3 Domain**".
Step 8 : Choose **Java EE 6 Web** as **Java EE Version:**. Keep rest as default.
Step 9 : Click **Finish**. (See fig below)


New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: 7 GlassFish Server 4.0 Add...
Java EE Version: Java EE 6 Web 8
Enable Contexts and Dependency Injection
Context Path: /Calculator
9
< Back
Next >
Finish
Cancel
Help

2- Creating a SOAP Web Service called as "CalculatorService"

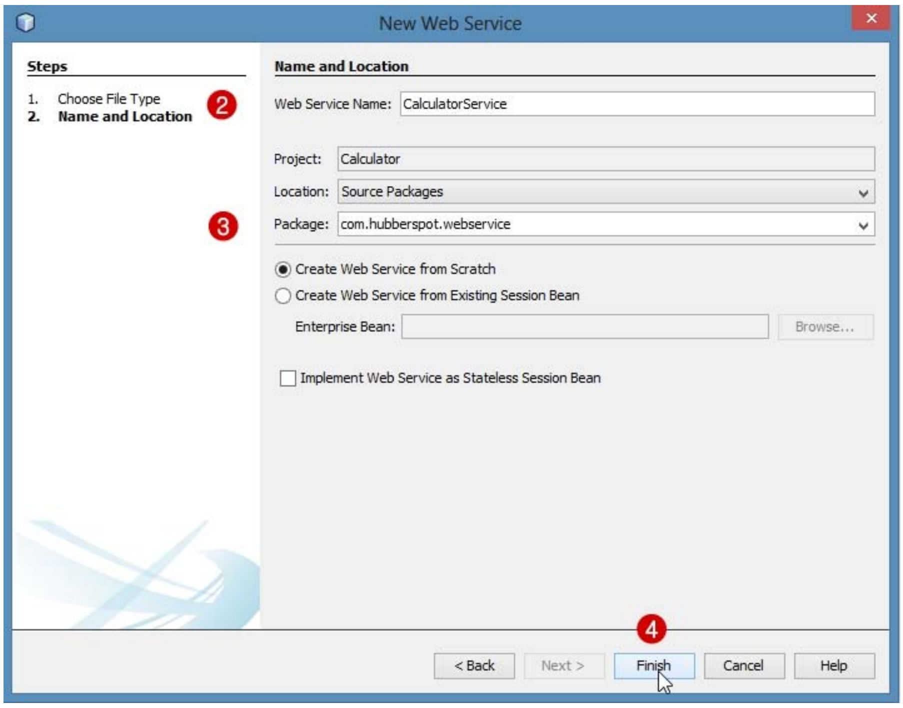
Step 1 : Right click on **Calculator** project and Select **New ---> Web Service...** (see fig below)



Step 2 : New Web Service dialog box gets open. In the **Web Service Name:** textfield enter name as "**CalculatorService**".

Step 3 : Enter the package name for the **CalculatorService** Web Service.

Step 4 : Click **Finish**. (see fig below)



Name and Location

Web Service Name: CalculatorService

Project: Calculator

Location: Source Packages

Package: com.hubberspot.webservice

(●) Create Web Service from Scratch

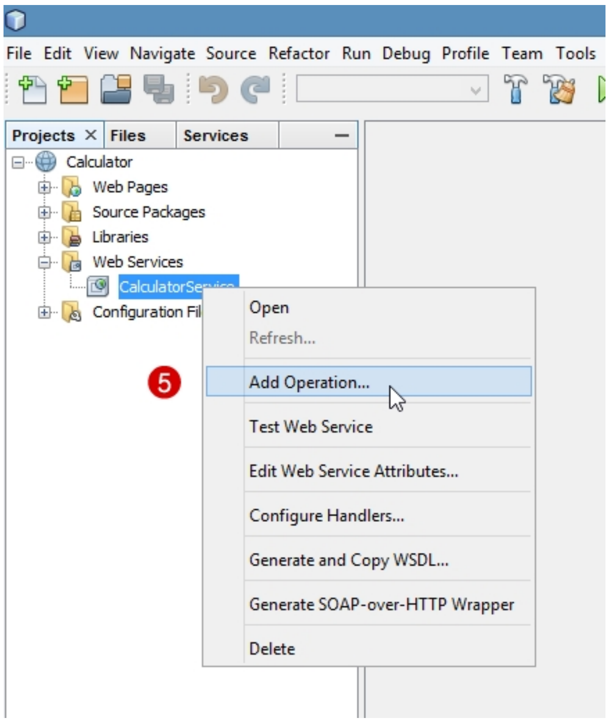
( ) Create Web Service from Existing Session Bean

Enterprise Bean:

[ ] Implement Web Service as Stateless Session Bean

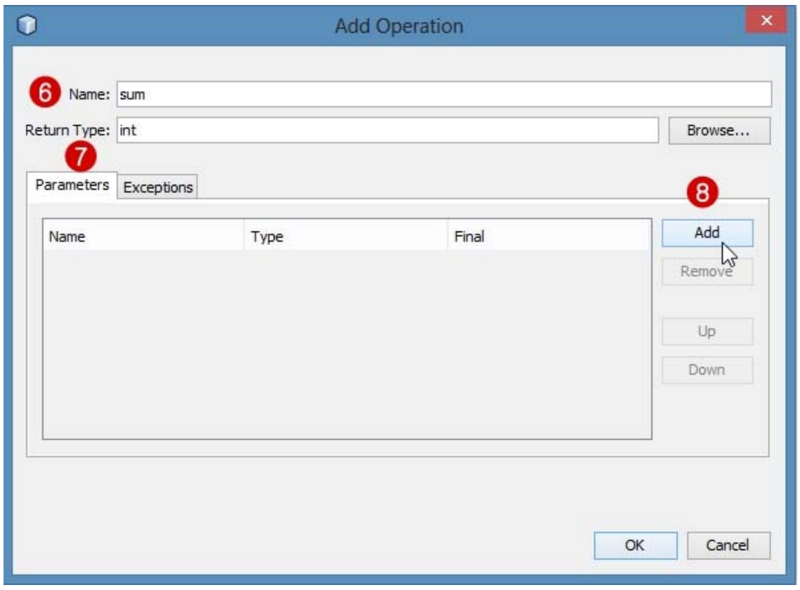
3- Creating a simple operation called as "sum".

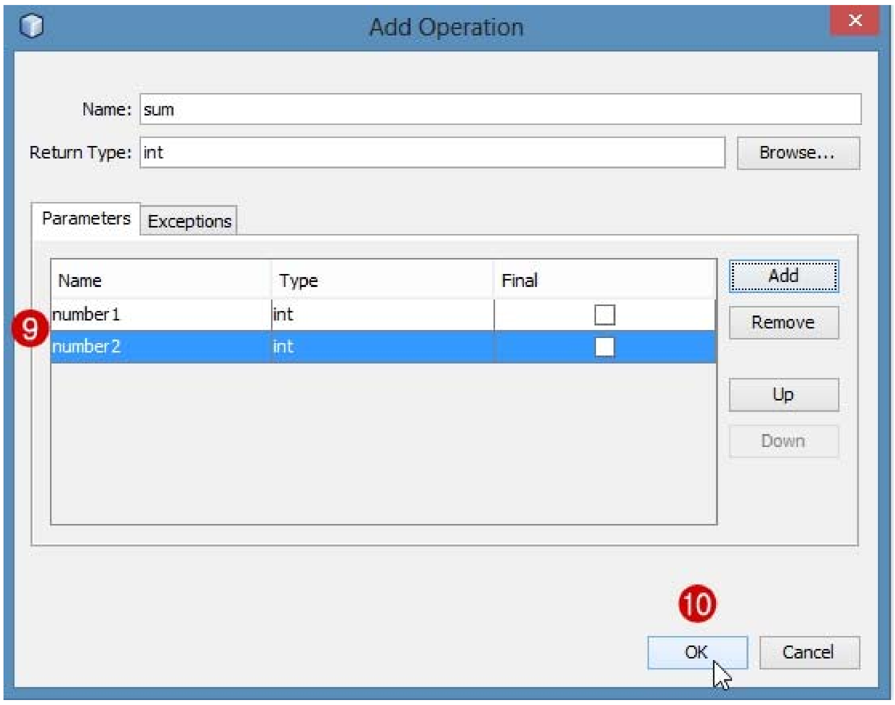
Step 5:- After creating Web Service by name "**CalculatorService**"**.** Under **Web Services** directory of the project, right click on the **CalculatorService** created and click on "**Add Operation**". (see fig below)



Add Operation dialog box gets open.

Step 6 : Enter name of the operation to expose as Web Service method. Here provide as "**sum**".
Step 7 : Enter the return type for the method. Here sum method will calculate sum of two numbers "**number1**" and "**number2**" and return it as a **int** value.
Step 8, 9 and 10 : Enter two **parameters** by clicking **add** button as "**number1**" and "**number2**" of type **int**, whose sum is to be calculated in the method sum. Click **OK**. (see fig below)





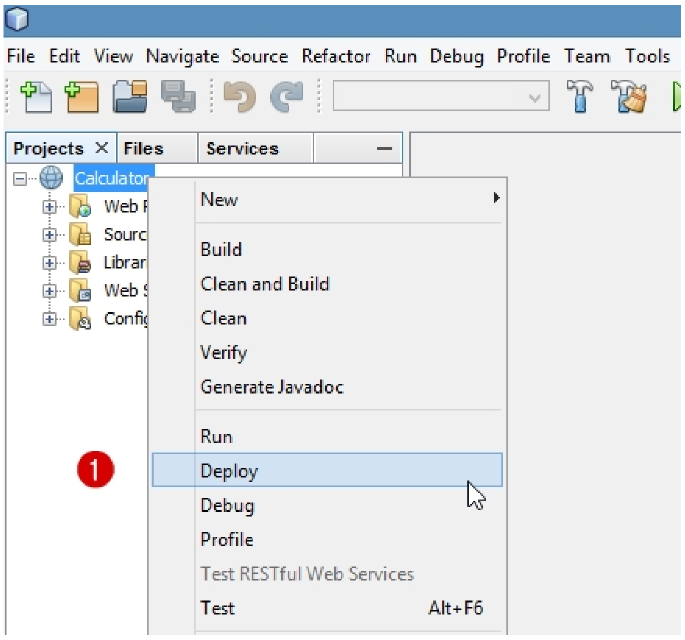
Name	Type	Final
number1	int	☐
number2	int	☐

Open Web Service class by name "**CalculatorService**". Operation by name **sum** gets created having return type as **int**. It gets in two parameters as **number1** and **number2** of the type **int**. The java class is now a Web Service as it is annotated by **@javax.jws.WebService**. The operation sum becomes the exposed method of the Web Service as it is annotated by **@javax.jws.WebMethod**. This method takes in two SOAP request parameters of type int annotated as **@javax.jws.WebParam**. (see java class below).

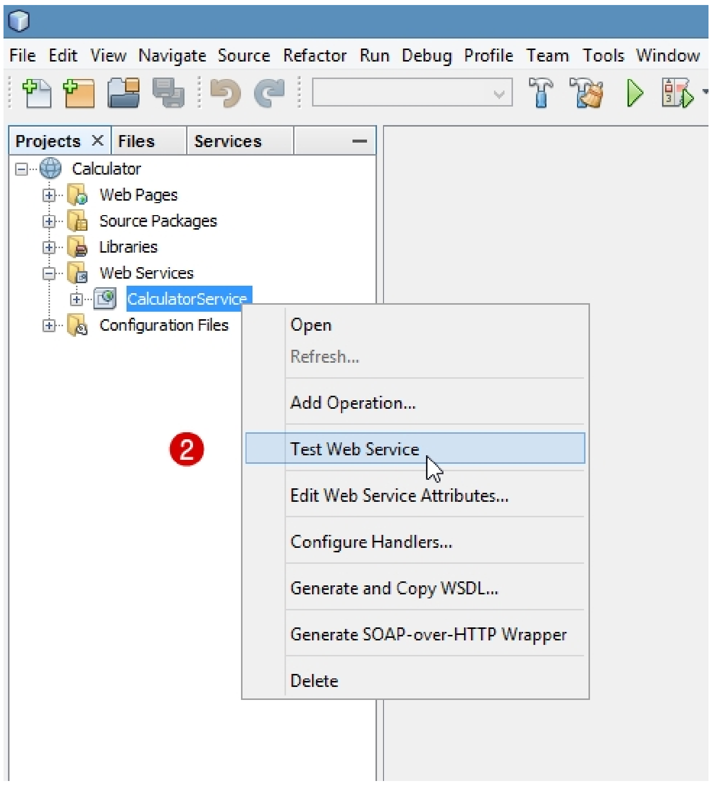
```
package com.hubberspot.webservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

// @WebService annotation makes a class a Web Service
@WebService(serviceName = "CalculatorService")
public class CalculatorService {

    // @WebMethod annotation expose a method as a service.
    @WebMethod(operationName = "sum")
    // @WebParam annotation indicates parameters to method coming from SOAP
request.
    public int sum(@WebParam(name = "number1") int number1,
                   @WebParam(name = "number2") int number2) {

        int sum = 0;
        sum = number1 + number2;
        return sum;
    }
}
```

4 - Deploy and Test the Web Service.

Step 1 : Right click on the "Calculator" project directory and click "Deploy". The Web Service gets deployed on the GlassFish Server. (see fig below)

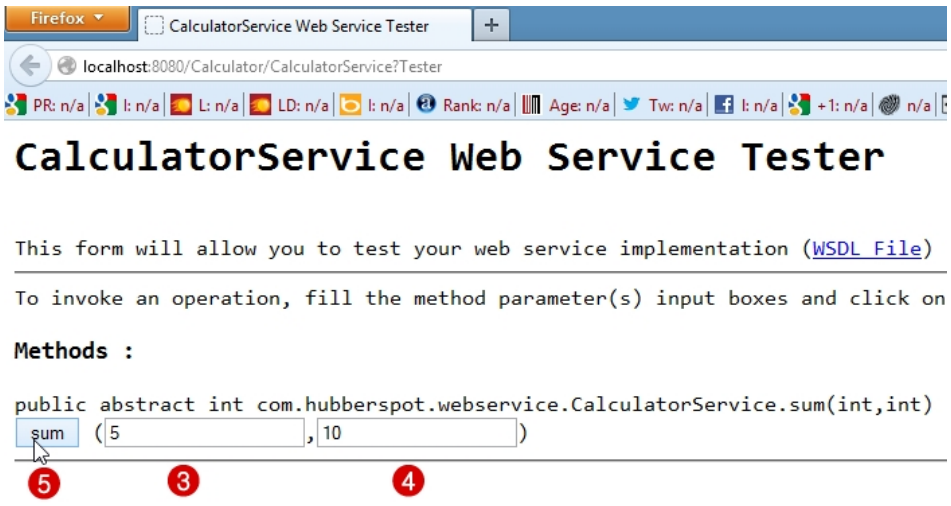


Step 2 : Under Web Services directory of the project, right click on the CalculatorService created and click on "Test Web Service". It opens a browser window **to test the calculatorService Web Service** (see fig below)



A browser window gets open and GlassFish Server creates a Tester client on the URL http://localhost:8080/Calculator/CalculatorService?Tester . The Tester client has a link to the WSDL file created for the Web Service. The link to WSDL file is at : http://localhost:8080/Calculator/CalculatorService?WSDL (see fig and WSDL file below).

Step 3 : In order to test sum method exposed as Web Service. Enter **number1** parameter value on the Tester client say 5.
Step 4 : Enter **number2** parameter value on the Tester client say 10.
Step 5: After the entering the value for **number1** and **number2** variables click **sum** button.


Firefox
CalculatorService Web Service Tester
localhost:8080/Calculator/CalculatorService?Tester
PR: n/a | I: n/a | L: n/a | LD: n/a | I: n/a | Rank: n/a | Age: n/a | Tw: n/a | I: n/a | +1: n/a | n/a

CalculatorService Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on

Methods :

public abstract int com.hubberspot.webservice.CalculatorService.sum(int,int)

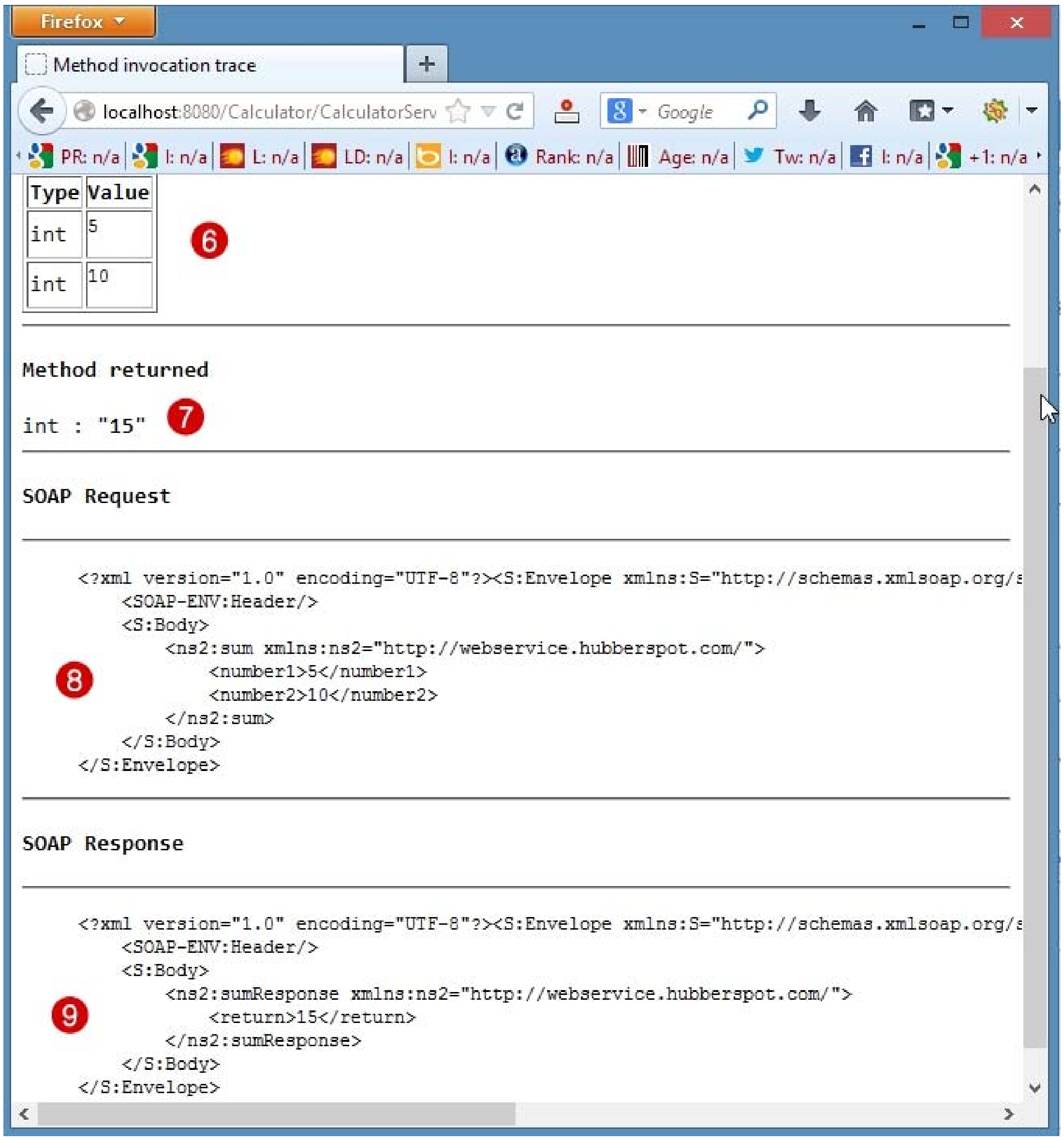
sum (5 , 10)

WSDL file for the CalculatorService Web Service -

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions targetNamespace="http://webservice.hubberspot.com/"
             name="CalculatorService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:tns="http://webservice.hubberspot.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">

  <types>
    <xsd:schema>
      <xsd:import namespace="http://webservice.hubberspot.com/"
                  schemaLocation="CalculatorService_schema1.xsd"/>
    </xsd:schema>
  </types>

  <message name="sum">
    <part name="parameters" element="tns:sum"/>
  </message>
  <message name="sumResponse">
    <part name="parameters" element="tns:sumResponse"/>
  </message>

  <portType name="CalculatorService">
    <operation name="sum">
    <input
wsam:Action="http://webservice.hubberspot.com/CalculatorService/sumRequest"
             message="tns:sum"/>
    <output
wsam:Action="http://webservice.hubberspot.com/CalculatorService/sumResponse
            "message="tns:sumResponse"/>
    </operation>
  </portType>

  <binding name="CalculatorServicePortBinding"
    type="tns:CalculatorService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
     style="document"/>
     <operation name="sum">
       <soap:operation soapAction=""/>
       <input>
         <soap:body use="literal"/>
       </input>
       <output>
         <soap:body use="literal"/>
       </output>
     </operation>
  </binding>

  <service name="CalculatorService">
    <port name="CalculatorServicePort"
     binding="tns:CalculatorServicePortBinding">
     <soap:address
       location="http://localhost:8080/Calculator/CalculatorService"/>
    </port>
  </service>
</definitions>
```

On clicking sum button, "**CalculatorService**" Web Service method called as sum gets executed taking in 5 and 10 as int parameters and returns sum as 15.

Step 6 : It shows method parameters as 5 and 10 of type int.
Step 7 : The return value after execution of sum method.
Step 8 : It shows the **SOAP request** for the "**CalculatorService**" Web Service which is been send in the form of xml. It is sending the values for number1 and number2 for the sum method.
Step 9 : It shows the **SOAP response** coming from the "**CalculatorService**" Web Service, returning the output of sum of number1 and number2.(see fig below)


Firefox
Method invocation trace
localhost:8080/Calculator/CalculatorServ
Google
PR: n/a I: n/a L: n/a LD: n/a I: n/a Rank: n/a Age: n/a Tw: n/a I: n/a +1: n/a
6
7
8
9

Type	Value
int	5
int	10

```
Method returned

int : "15"

SOAP Request

    <?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/s
        <SOAP-ENV:Header/>
        <S:Body>
            <ns2:sum xmlns:ns2="http://webservice.hubberspot.com/">
                <number1>5</number1>
                <number2>10</number2>
            </ns2:sum>
        </S:Body>
    </S:Envelope>

SOAP Response

    <?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/s
        <SOAP-ENV:Header/>
        <S:Body>
            <ns2:sumResponse xmlns:ns2="http://webservice.hubberspot.com/">
                <return>15</return>
            </ns2:sumResponse>
        </S:Body>
    </S:Envelope>
```