

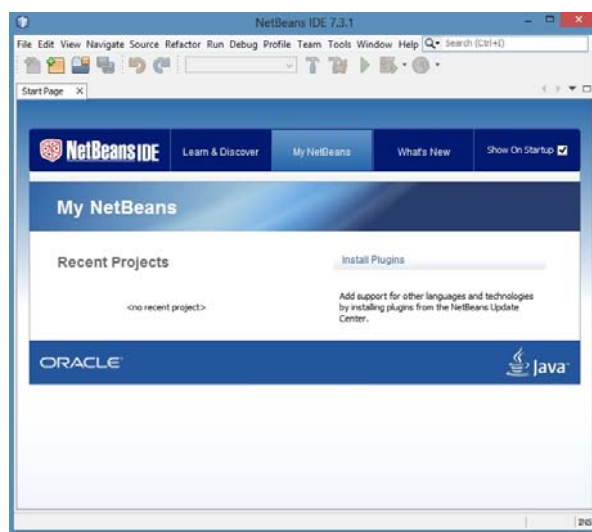
Creation a Java client for consuming SOAP Web Service

After deploying **CalculatorService** Web Service in GlassFish Server, refer post : [How to create your first SOAP based Web Service in Java using JAX-WS ?](#) , the next step is to make a Java client that can consume it. In above article, **CalculatorService** had a method as **sum**. In this post we will create a Java client for consuming sum by passing in two numbers and getting back the sum of it. The steps involved in creating Java client are as follows :

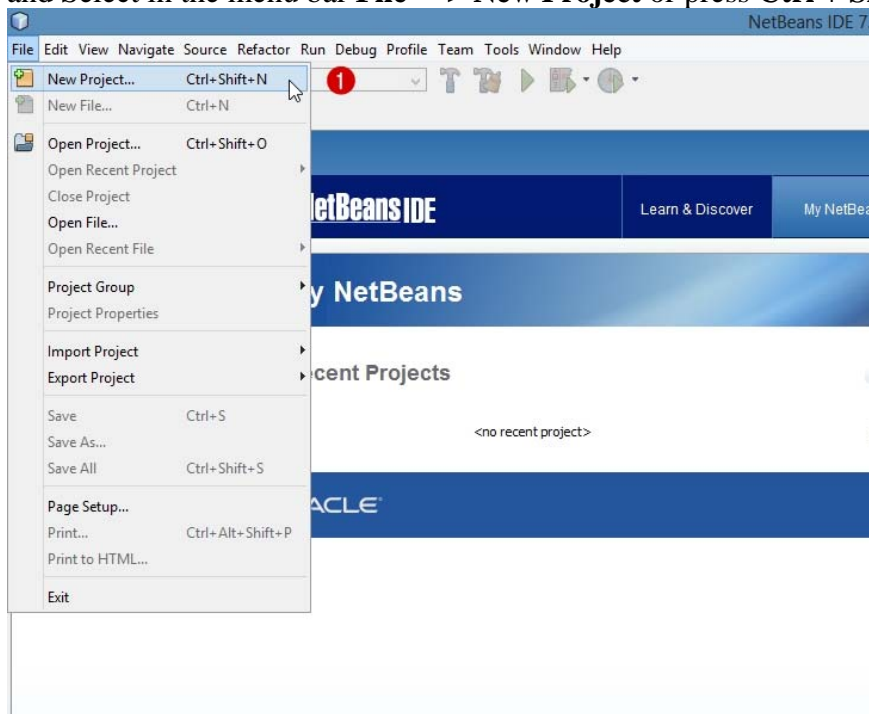
1. **Creating a Java application as CalculatorServiceClient.**
2. **Adding Web Service client to the Java application.**
3. **Creating a Java client class for consuming CalculatorService Web Service.**
4. **Build and Run the Java client.**

1- Creating a Java application as CalculatorServiceClient

Step 1 : Open NetBeans IDE (See fig below)



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

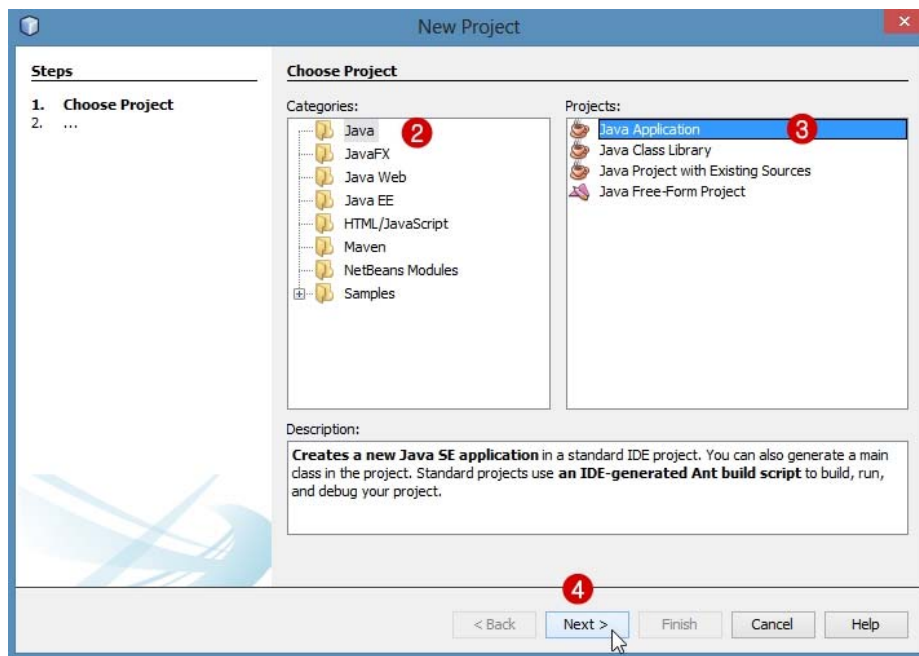


New Project dialog box gets open.

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

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

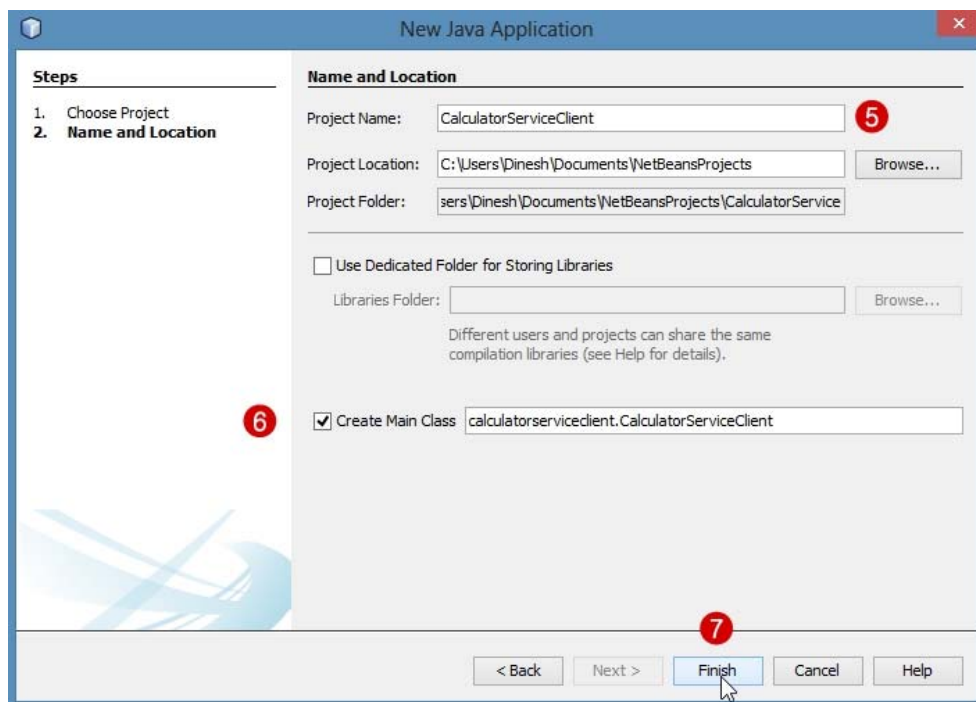
Step 4 : Click **Next >** . (See fig below)



Step 5 : Under **Name and Location**, provide **Project Name:** as "**CalculatorServiceClient**".

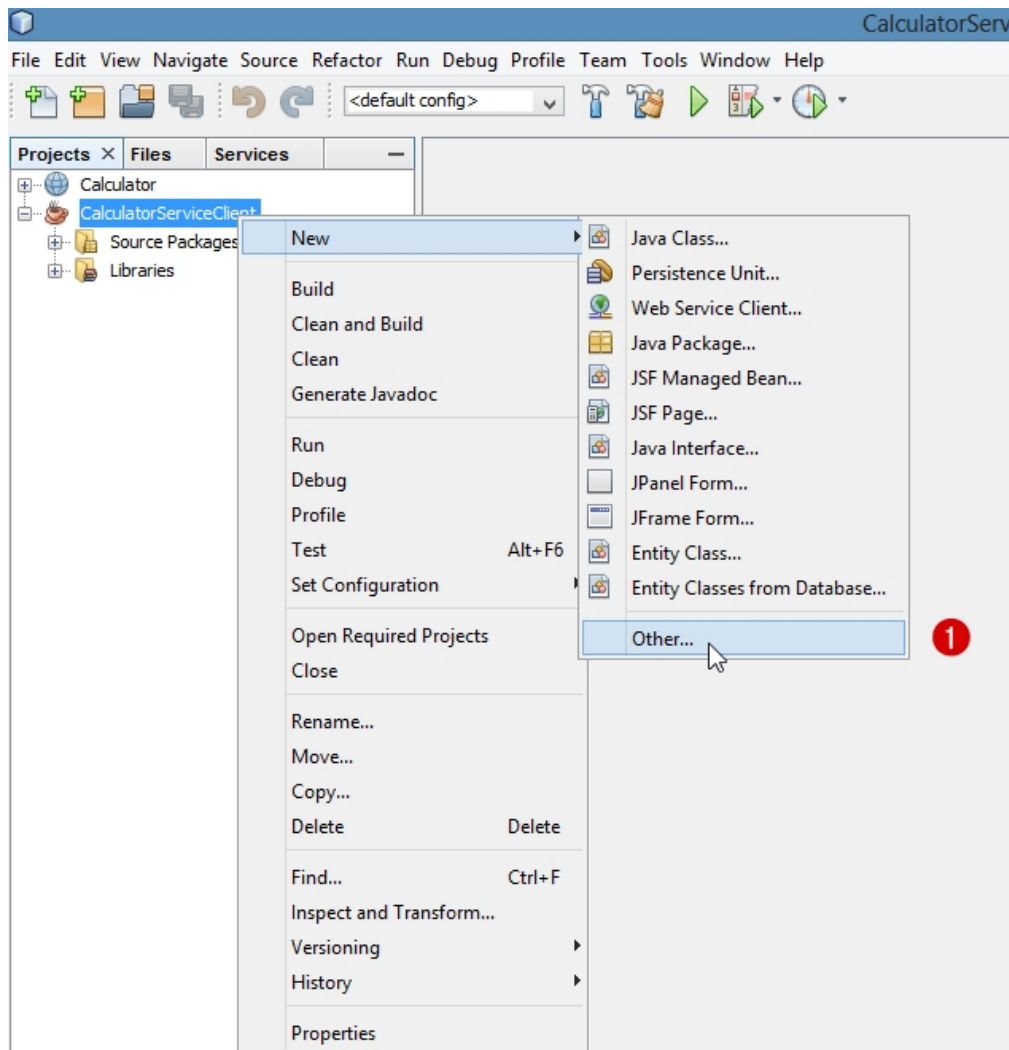
Step 6 : Select check box for **(Create Main Class)** as it creates a main class which will be our Web Service Java Client.

Step 7 : Click **Finish**. (see fig below)



2- Adding Web Service client to the Java application.

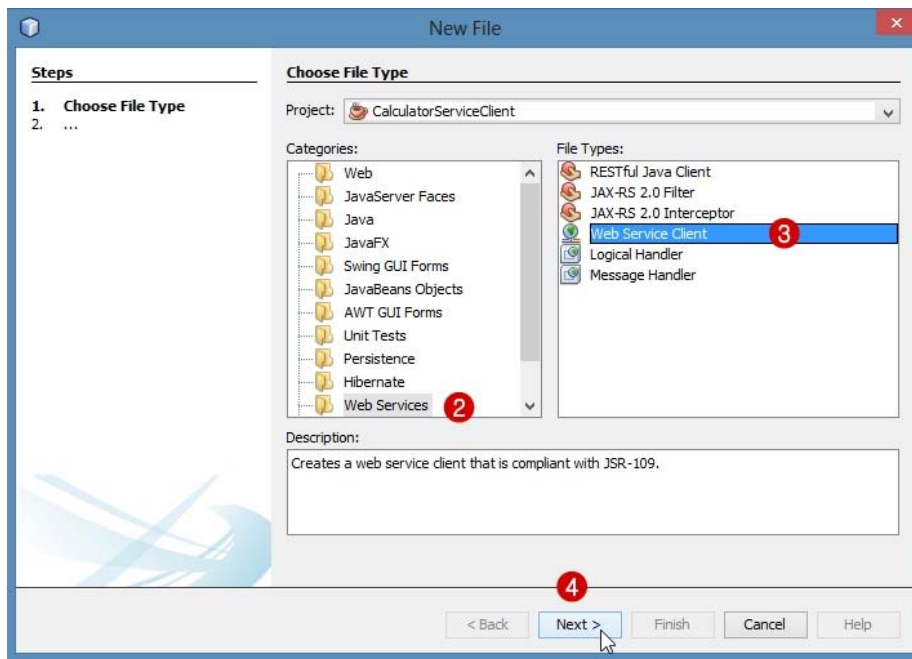
Step 1: Right click **CalculatorServiceClient** project directory and browse to **New --- > Other.** (see fig below)



Step 2:- New File dialog box gets open. Under **Categories:** select **Web Services.**

Step 3:- Under **File Types:** select **Web Service Client**

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

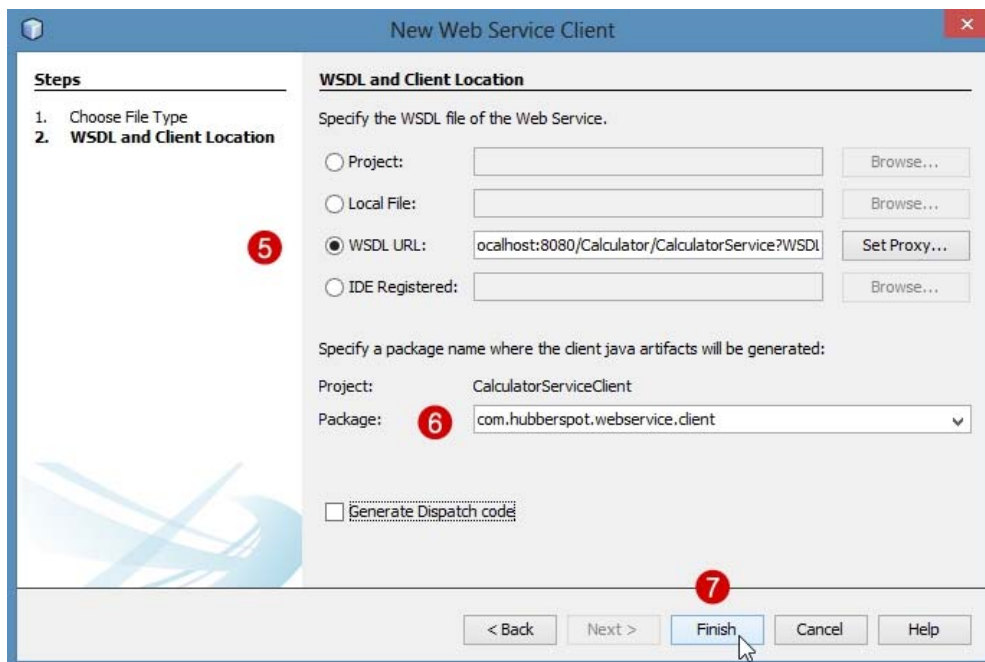


Step 5 : **New Web Service Client** dialog box gets open. Under **WSDL and Client Location** , specify the WSDL file location for the **CalculatorService** Web Service by following options as **Project :** , **Local File :** , **WSDL URL :** , **IDE Registered:**.

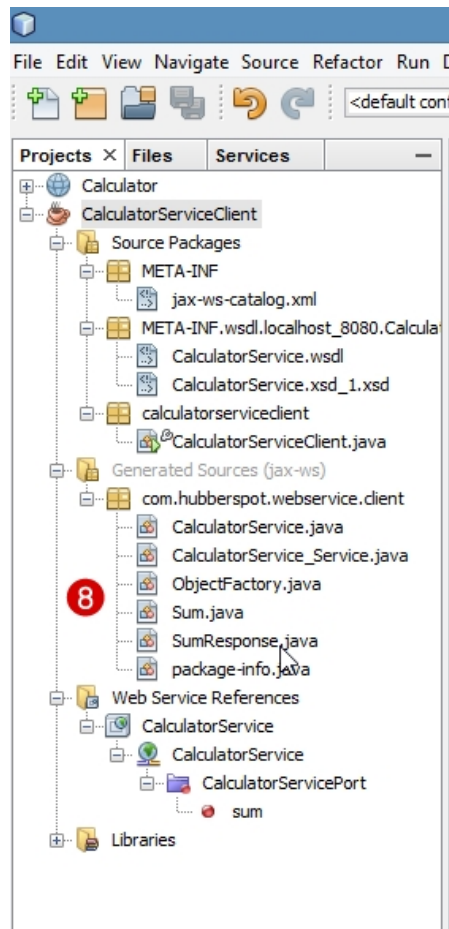
Select **WSDL URL** and provide the url for the deployed **CalculatorService** WSDL on the localhost. As we have deployed CalculatorService on the GlassFish Server in the above mentioned post, here it is pointing at "**http://localhost:8080/Calculator/CalculatorService?WSDL**", localhost is the ip address for it. If Web Service is deployed on another server provide the ip address for it on the WSDL url.

Step 6:- Enter the package name.

Step 7:- Click **Finish**. (see fig below)



Step 8 : It creates client side artifacts for the **Service Endpoint Interface (SEI)**, which acts as a Proxy for the SOAP Web Service. (see fig below)



3- Creating a Java client class for consuming CalculatorService Web Service.

Open the **CalculatorServiceClient.java** and add the following code to it.

```
package calculatorserviceclient;

import com.hubberspot.webservice.client.CalculatorService;
import com.hubberspot.webservice.client.CalculatorService_Service;

public class CalculatorServiceClient {
    public static void main(String[] args) {
        int number1 = 6;
        int number2 = 12;
        // Create the proxy Web Service and call the sum method
        // using the port of the proxy service.
        CalculatorService_Service service = new
        CalculatorService_Service();
        CalculatorService port = service.getCalculatorServicePort();

        // retrieve the sum of number1 and number2
        int sum = port.sum(number1, number2);

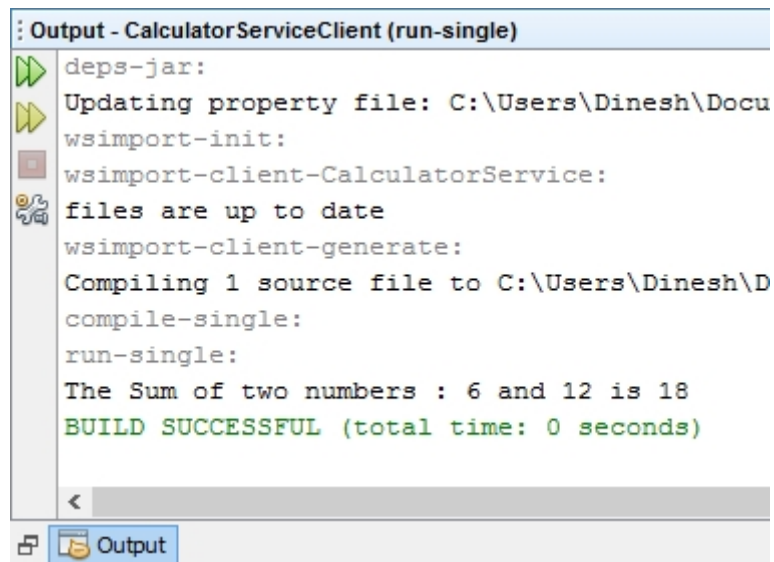
        System.out.println("The Sum of two numbers : " + number1
            + " and " + number2 + " is " + sum);
    }
}
```

3- Build and Run the Java client

Step 1 : Right click **CalculatorServiceClient** project directory and select **Build**.

Step 2 : Run the **CalculatorServiceClient.java** class

Output of the program :



```
: Output - CalculatorServiceClient (run-single)
>> deps-jar:
>> Updating property file: C:\Users\Dinesh\Docu
>> wsimport-init:
>> wsimport-client-CalculatorService:
>> files are up to date
>> wsimport-client-generate:
>> Compiling 1 source file to C:\Users\Dinesh\D
>> compile-single:
>> run-single:
>> The Sum of two numbers : 6 and 12 is 18
>> BUILD SUCCESSFUL (total time: 0 seconds)
```

The screenshot shows the Eclipse IDE's Output window for the project 'CalculatorServiceClient'. The window title is ': Output - CalculatorServiceClient (run-single)'. The output text shows the following steps: 'deps-jar:', 'Updating property file: C:\Users\Dinesh\Docu', 'wsimport-init:', 'wsimport-client-CalculatorService:', 'files are up to date', 'wsimport-client-generate:', 'Compiling 1 source file to C:\Users\Dinesh\D', 'compile-single:', 'run-single:', 'The Sum of two numbers : 6 and 12 is 18', and 'BUILD SUCCESSFUL (total time: 0 seconds)'. The window has a scroll bar and a 'Output' button at the bottom.