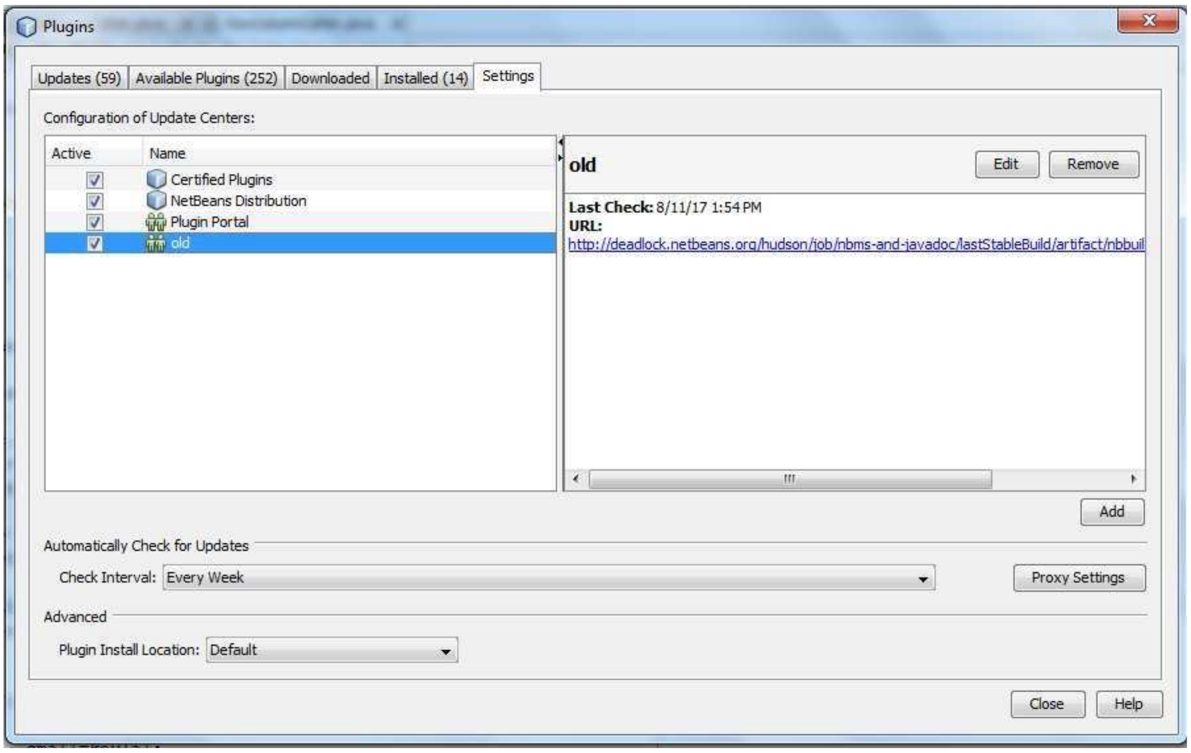
1. Install axis2 plugin



Configuration of Update Centers:

Active	Name
✓	Certified Plugins
✓	NetBeans Distribution
✓	Plugin Portal
✓	old

old

Last Check: 8/11/17 1:54 PM
URL:
http://deadlock.netbeans.org/hudson/job/nbms-and-javadoc/lastStableBuild/artifact/nbbuil

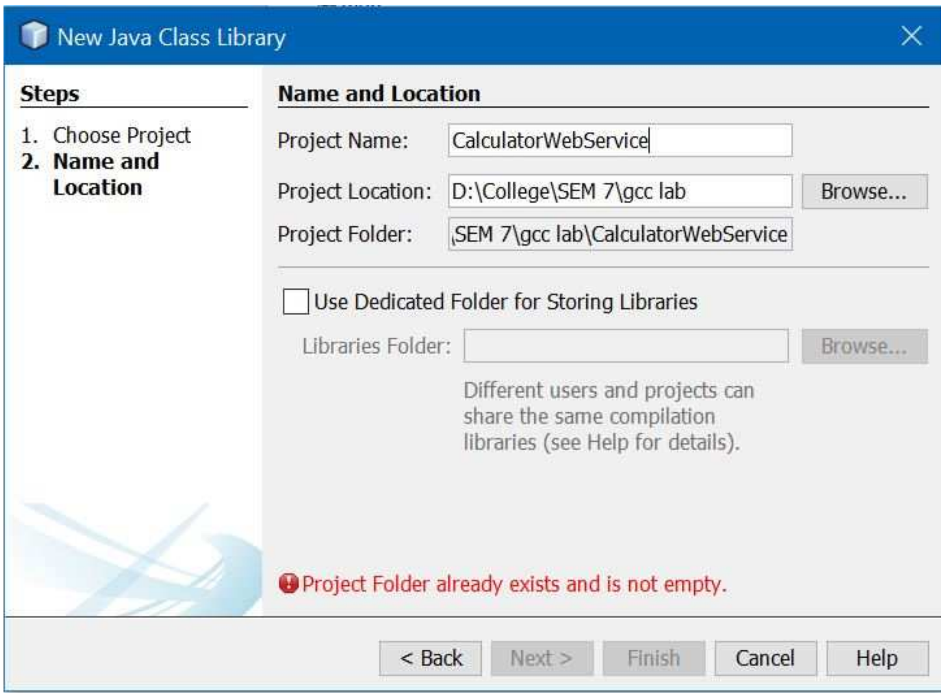
Automatically Check for Updates

Check Interval: Every Week

Advanced

Plugin Install Location: Default

2. Create a Web Service



Name and Location

Project Name: CalculatorWebService

Project Location: D:\College\SEM 7\gcc lab

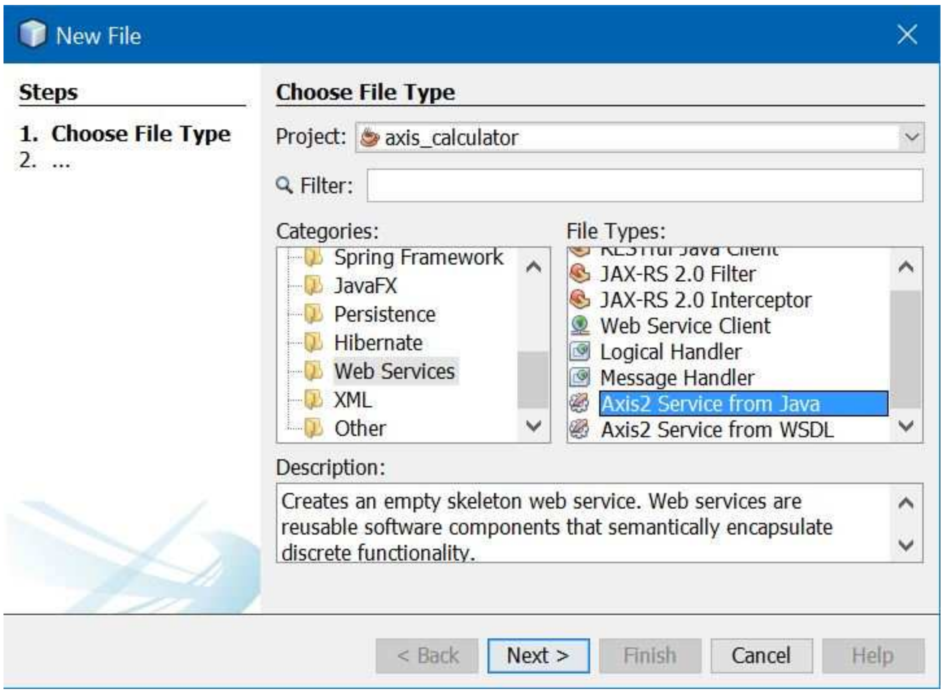
Project Folder: SEM 7\gcc lab\CalculatorWebService

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Project Folder already exists and is not empty.



New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: axis_calculator

Filter:

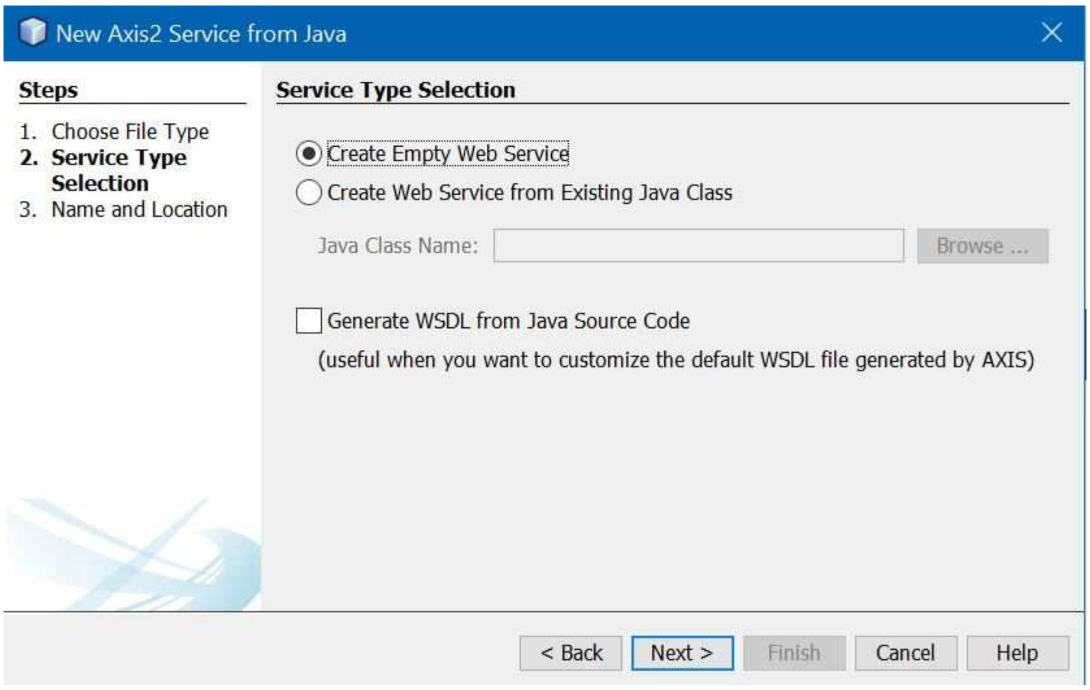
Categories:
- Spring Framework
- JavaFX
- Persistence
- Hibernate
- Web Services
- XML
- Other

File Types:
- RESTful Java Client
- JAX-RS 2.0 Filter
- JAX-RS 2.0 Interceptor
- Web Service Client
- Logical Handler
- Message Handler
- Axis2 Service from Java
- Axis2 Service from WSDL

Description:

Creates an empty skeleton web service. Web services are reusable software components that semantically encapsulate discrete functionality.

< Back | Next > | Finish | Cancel | Help



New Axis2 Service from Java

Steps

1. Choose File Type
2. **Service Type Selection**
3. Name and Location

Service Type Selection

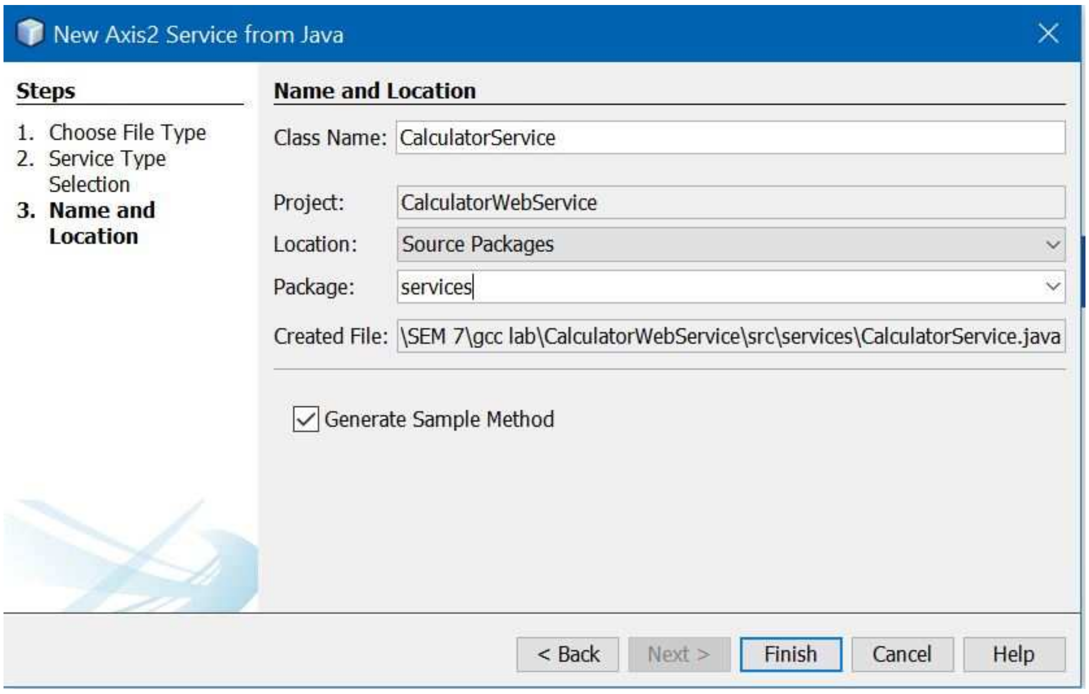
- [✓] Create Empty Web Service
- Create Web Service from Existing Java Class

Java Class Name: Browse ...

- [ ] Generate WSDL from Java Source Code

(useful when you want to customize the default WSDL file generated by AXIS)

< Back | Next > | Finish | Cancel | Help



Class Name: CalculatorService

Project: CalculatorWebService

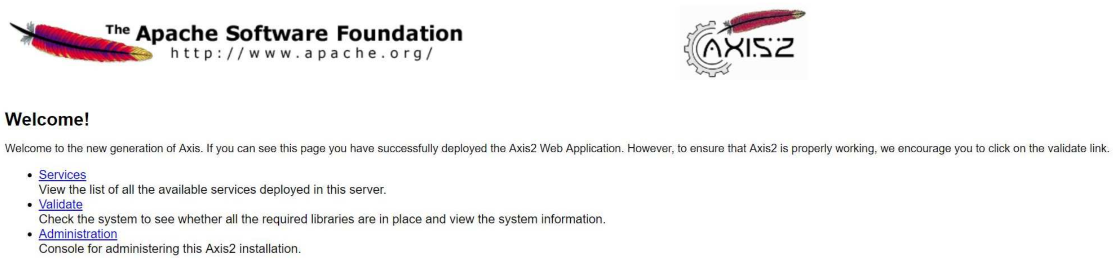
Location: Source Packages

Package: services

Created File: \SEM 7\gcc lab\CalculatorWebService\src\services\CalculatorService.java

☑ Generate Sample Method

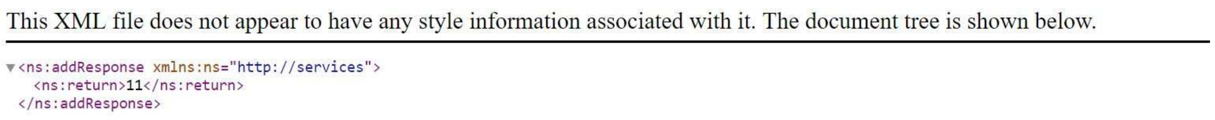
3. Deploy and test the web service



Welcome!

Welcome to the new generation of Axis. If you can see this page you have successfully deployed the Axis2 Web Application. However, to ensure that Axis2 is properly working, we encourage you to click on the validate link.

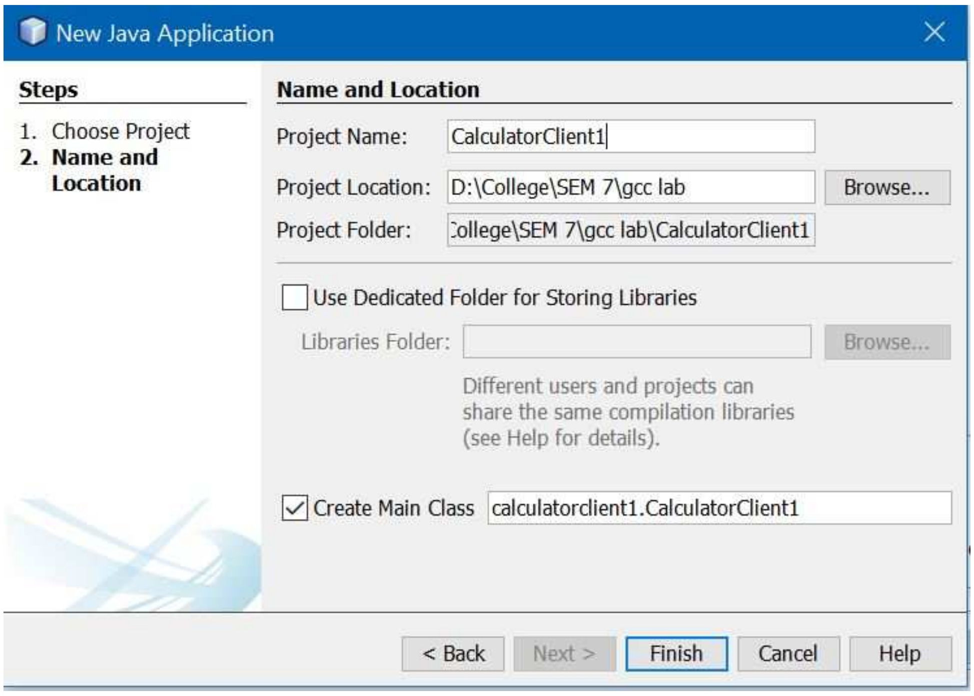
- Services
  View the list of all the available services deployed in this server.
- Validate
  Check the system to see whether all the required libraries are in place and view the system information.
- Administration
  Console for administering this Axis2 installation.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ns:addResponse xmlns:ns="http://services">
   <ns:return>11</ns:return>
 </ns:addResponse>
```

4. Create a web service Client



Project Name: CalculatorClient1

Project Location: D:\College\SEM 7\gcc lab

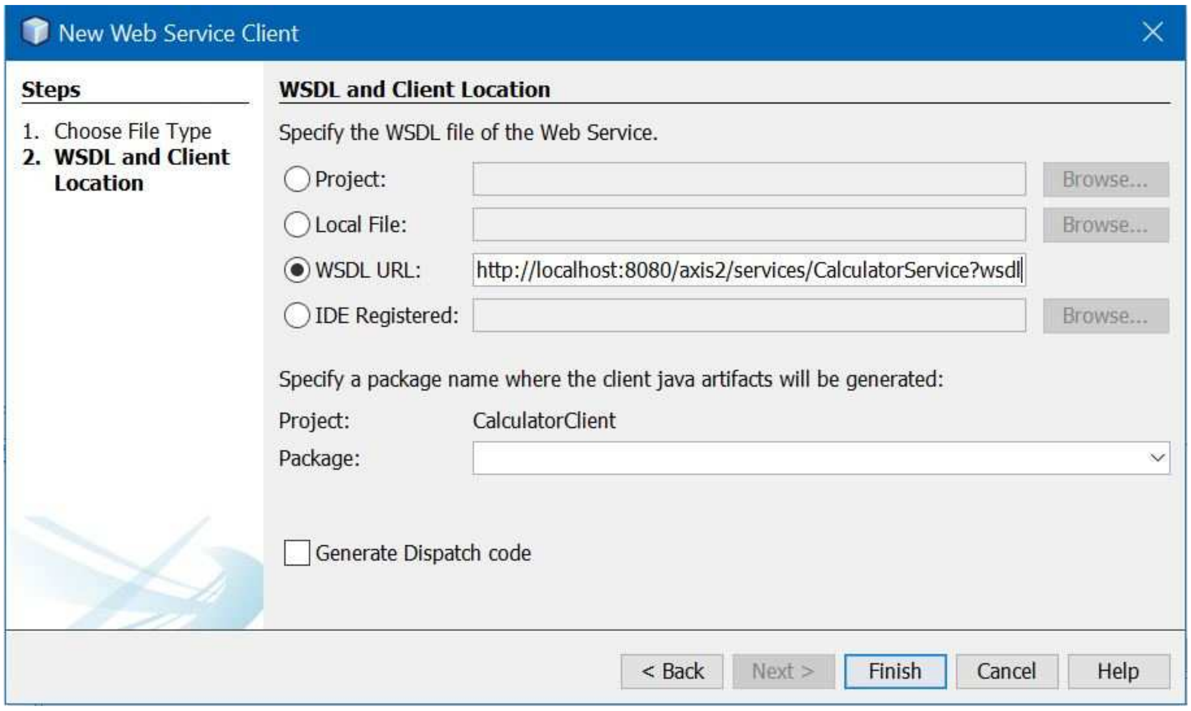
Project Folder: :ollege\SEM 7\gcc lab\CalculatorClient1

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class calculatorclient1.CalculatorClient1



Specify the WSDL file of the Web Service.

○ Project:

○ Local File:

● WSDL URL: http://localhost:8080/axis2/services/CalculatorService?wsdl

○ IDE Registered:

Specify a package name where the client java artifacts will be generated:

Project: CalculatorClient

Package:

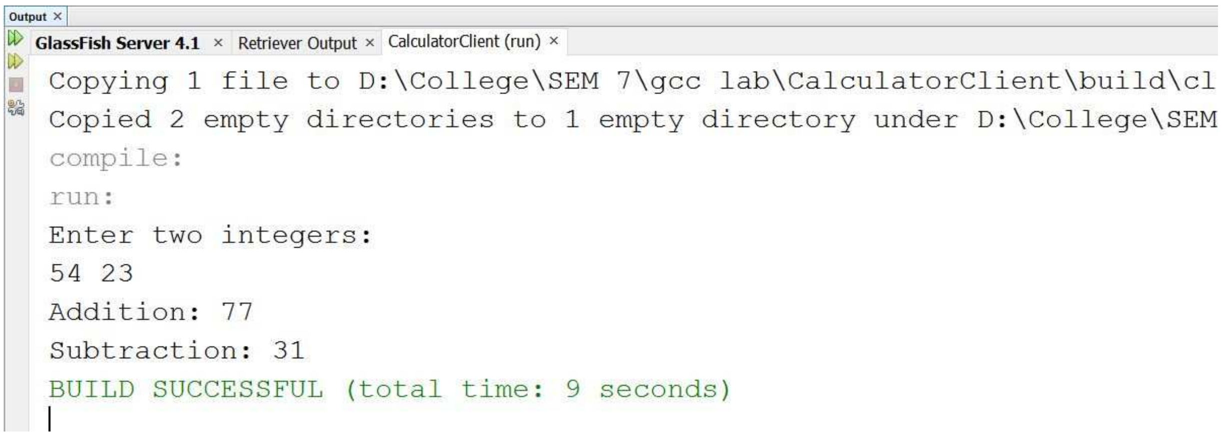
☐ Generate Dispatch code

Web Service:

```
package services;

/**
 *
 * @author Simran
 */
public class CalculatorService {

    /**
     * Sample method
     */
public String hello(String name) {
return "Hello " + name;
    }
publicint add(int x, int y) {
returnx+y;
    }

publicint sub(int x, int y) {
return x-y;
    }

}
```

Web Client:

```
packagecalculatorclient;

importjava.util.Scanner;

/**
 *
 * @author Simran
 */
public class CalculatorClient {

    /**
     * @paramargs the command line arguments
     */
public static void main(String[] args) {
        // TODO code application logic here
System.out.println("Enter two integers: ");
        Scanner in = new Scanner(System.in);
int a = in.nextInt();
int b = in.nextInt();
System.out.printf("Addition: %d\n",add(a,b));
System.out.printf("Subtraction: %d\n",sub(a,b));
    }

private static Integer add(java.lang.Integer x, java.lang.Integer y) {
services.CalculatorService service = new services.CalculatorService();
services.CalculatorServicePortType port =
service.getCalculatorServiceHttpSoap11Endpoint();
returnport.add(x, y);
    }

private static Integer sub(java.lang.Integer x, java.lang.Integer y) {
services.CalculatorService service = new services.CalculatorService();
services.CalculatorServicePortType port =
service.getCalculatorServiceHttpSoap11Endpoint();
returnport.sub(x, y);
    }

}
```



```
Copying 1 file to D:\College\SEM 7\gcc lab\CalculatorClient\build\cl
Copied 2 empty directories to 1 empty directory under D:\College\SEM
compile:
run:
Enter two integers:
54 23
Addition: 77
Subtraction: 31
BUILD SUCCESSFUL (total time: 9 seconds)
```