Abstract

A web service provides a crucial communication mechanism between two programs while ensuring platform independency. The web service can be used by any consumer process without restricting itself to specific programming platform or language. The project aims at designing and implementing one such web service for Grade Point Average (GPA) calculation. In this project, the service provider provides methods for recoding data regarding courses from the consumer and calculating the GPA based on the information provided by the client. Four services are implemented which include initializeCourses, setCourse, assignGradePoint and calculateGPA. The project comprises of implementation of the Service and the Client in Java using Eclipse Mars IDE, Apache Tomcat Server V7 and Apache Axis2 Service. Generation of WSDL and SOAP XML requests is handled internally by the Axis service. Further, the web service is tested thoroughly using two techniques: Testing Web Service Client provided readily by Eclipse IDE and SoapUI, a functional testing tool for SOAP protocol.

-Raj Palkar: https://in.linkedin.com/in/rajpalkar

Related Theory

Web Service:

A Web service basically is a collection of open protocols that is used to exchange data between applications[1]. The use of open protocols enables Web services to be platform independent. Software that are written in different programming languages and that run on different platforms can use Web services to exchange data over computer networks such as the Internet[1].

It is common knowledge that developed applications vary widely in the programming languages used for creating them and hence, they are not able to communicate and exchange data effectively. A combination of open protocols and standards like XML, SOAP and WSDL helps web services in overcoming these challenges and makes communication possible. A Web service uses XML to tag data, SOAP to transfer a message and finally WSDL to describe the availability of services[1].

SOAP:

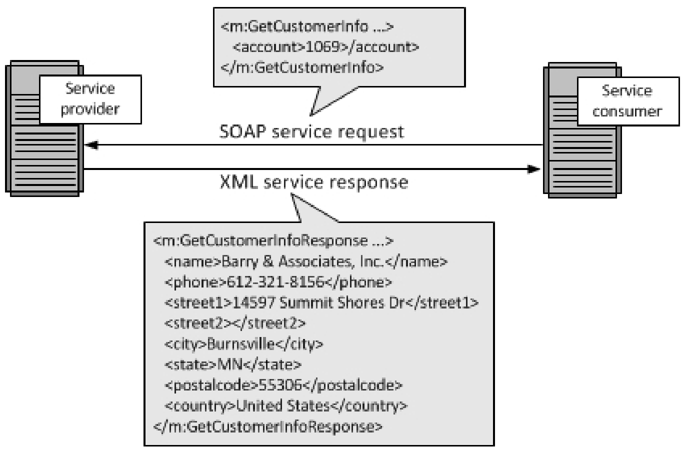
SOAP is an acronym for Simple Object Access Protocol. It is actually an application communication protocol which specifies the format for sending and receiving messages. SOAP is based on XML and hence, it ensure platform independency[2].

Following are the two components of a SOAP envelope

- An optional header providing information on authentication, encoding of data, or how a recipient of a SOAP message should process the message[3].
- The body that contains the message. These messages can be defined using the WSDL specification[3].

HTTP is generally used by SOAP, but it can also utilize protocols like SMTP. SOAP can be used to exchange complete documents or for a remote procedure call[3].

Following is the interacting between the client and service provider:



```
<m:GetCustomerInfo ...>
  <account>1069>/account>
</m:GetCustomerInfo>
```

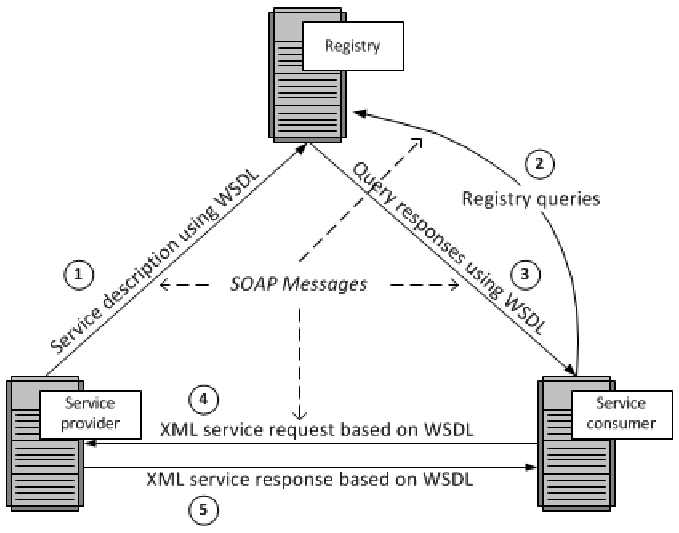
```
<m:GetCustomerInfoResponse ...>
  <name>Barry & Associates, Inc.</name>
  <phone>612-321-8156</phone>
  <street1>14597 Summit Shores Dr</street1>
  <street2></street2>
  <city>Burnsville</city>
  <state>MN</state>
  <postalcode>55306</postalcode>
  <country>United States</country>
</m:GetCustomerInfoResponse>
```

WSDL:

WSDL stands for Web Services Description Language. WSDL is basically responsible for indicating what methods are available in a Web Service and how a client can use those services based on the specifications. As the name suggests, it describes the used how to access and use the methods provided by the service. It is an XML-based language for describing Web services as stated by W3C recommendation[4].

The following figure illustrates the use of WSDL. Service provider is on the left side and the service consumer is on the right one. The steps listed below help in summarizing the interactions and use of WSDL.

- A service provider describes its service using WSDL. This definition is published to a repository of services[5].
- A service consumer issues one or more queries to the repository to locate a service and determine how to communicate with that service[5].
- Part of the WSDL provided by the service provider is passed to the service consumer so that the consumer gains knowledge regarding the services[5].
- The service consumer sends a request to the service provider using WSDL[5].
- The service provider outputs the response to the service consumer[5].


Registry
1
Service description using WSDL
2
Registry queries
Query responses using WSDL
3
SOAP Messages
Service provider
Service consumer
4
XML service request based on WSDL
XML service response based on WSDL
5

Tools Used

- Eclipse Mars IDE

  Eclipse is an integrated development environment (IDE)[9]. It contains a base workspace and an extensible plug-in system for customizing the environment[9]. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers[9]. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules[9].

- SoapUI - 5.2.1[6]

  SoapUI is the world leading Open Source Functional Testing Tool[6], mainly it is used for API testing [6]. SoapUI supports multiple protocols such as SOAP, REST, HTTP, JMS, AMF and JDBC[6]. SoapUI enables you to create advanced Performance Tests very quickly and run Automated Functional Tests[6].

- Apache Tomcat Server 7[7]

  Apache Tomcat™ is an open source software implementation of the Java Servlet[7], JavaServer Pages[7], Java Expression Language[7] and Java WebSocket technologies[7]. Apache Tomcat is developed in an open and participatory environment and released under the Apache License version 2[7]. Apache Tomcat[7] is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project[7].

- Apache Axis2[8] Web Service (version 1.64)

  Apache Axis2[8] is a core engine for Web services. It is a complete re-design and re-write of the widely used Apache Axis SOAP stack[8]. Implementations of Axis2 are available in Java and C[8]. Axis2 provides the capability to add Web services interfaces to Web applications[8].

- Java 8

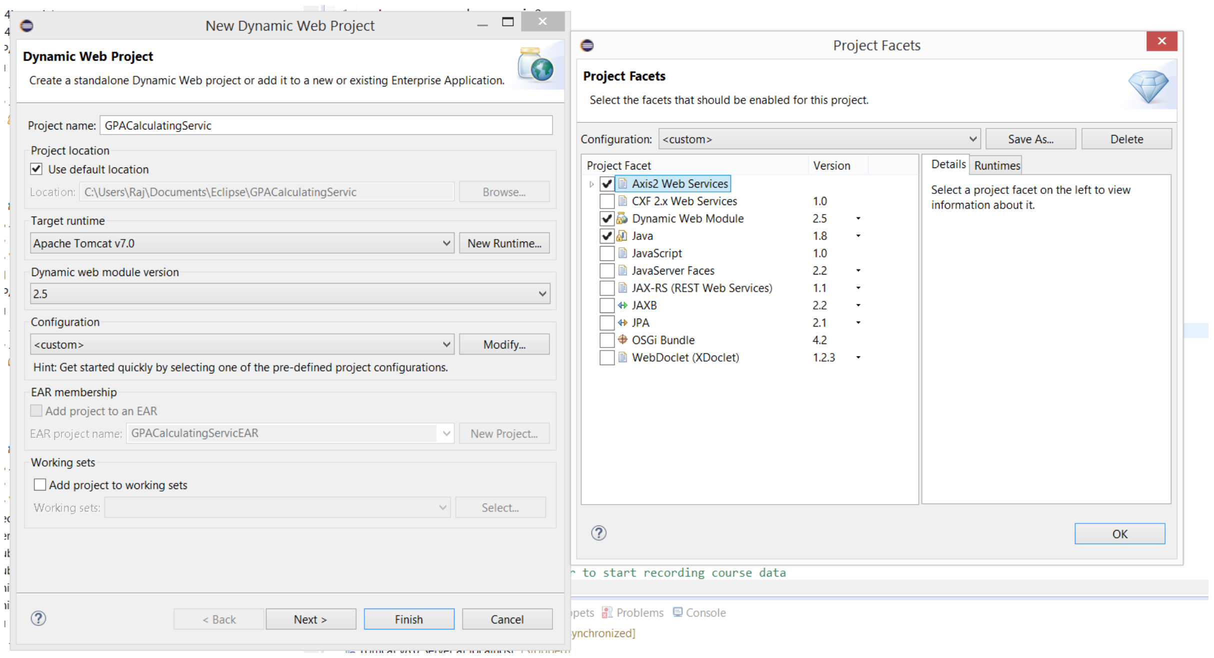
- Google Chrome Browser

- Windows 8

Implementation

1. The project starts out with creation of a dynamic web project **GPACalculatingService** in Eclipse IDE, with runtime target as Apache Tomcat v7.0 and dynamic web module version 2.5 in order to customize the project facets to be used.

We select the Java, Dynamic Web Module and Axis2 Web Services from the available project facets.


New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: GPACalculatingServic

Project location
☑ Use default location
Location: C:\Users\Raj\Documents\Eclipse\GPACalculatingServic

Target runtime: Apache Tomcat v7.0

Dynamic web module version: 2.5

Configuration: <custom>

Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR membership
☐ Add project to an EAR
EAR project name: GPACalculatingServiceEAR

Working sets
☐ Add project to working sets
Working sets:

Project Facets

Project Facets

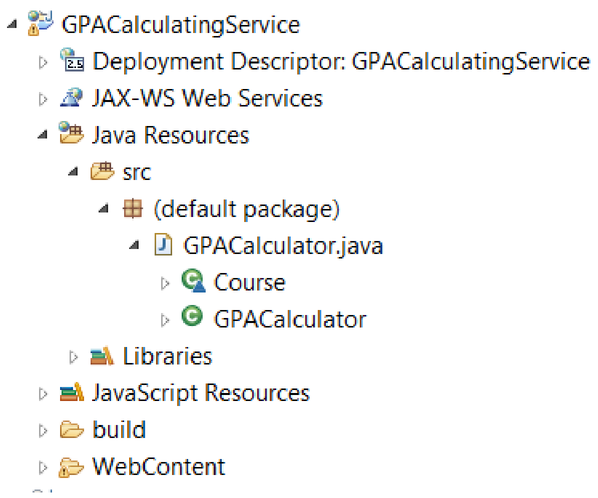
Select the facets that should be enabled for this project.

Configuration: <custom>

Project Facet	Version
☑ Axis2 Web Services	
☐ CXF 2.x Web Services	1.0
☑ Dynamic Web Module	2.5
☑ Java	1.8
☐ JavaScript	1.0
☐ JavaServer Faces	2.2
☐ JAX-RS (REST Web Services)	1.1
☐ JAXB	2.2
☐ JPA	2.1
☐ OSGi Bundle	4.2
☐ WebDoclet (XDoclet)	1.2.3

Details: Select a project facet on the left to view information about it.

2. Next step comprises of creation of a java class file which shall provide the web service functions that can be accessed by the clients. Thus, file **GPACalculator.java** is created in the Java Resources/src folder of **GPACalculatingService**.


```
GPACalculatingService
  Deployment Descriptor: GPACalculatingService
  JAX-WS Web Services
  Java Resources
    src
      (default package)
        GPACalculator.java
          Course
          GPACalculator
    Libraries
  JavaScript Resources
  build
  WebContent
```

3. Now, the methods to be provided by the Web Service are implemented in the **GPACalulator.java** file. Following is the code of **GPACalculator.java**

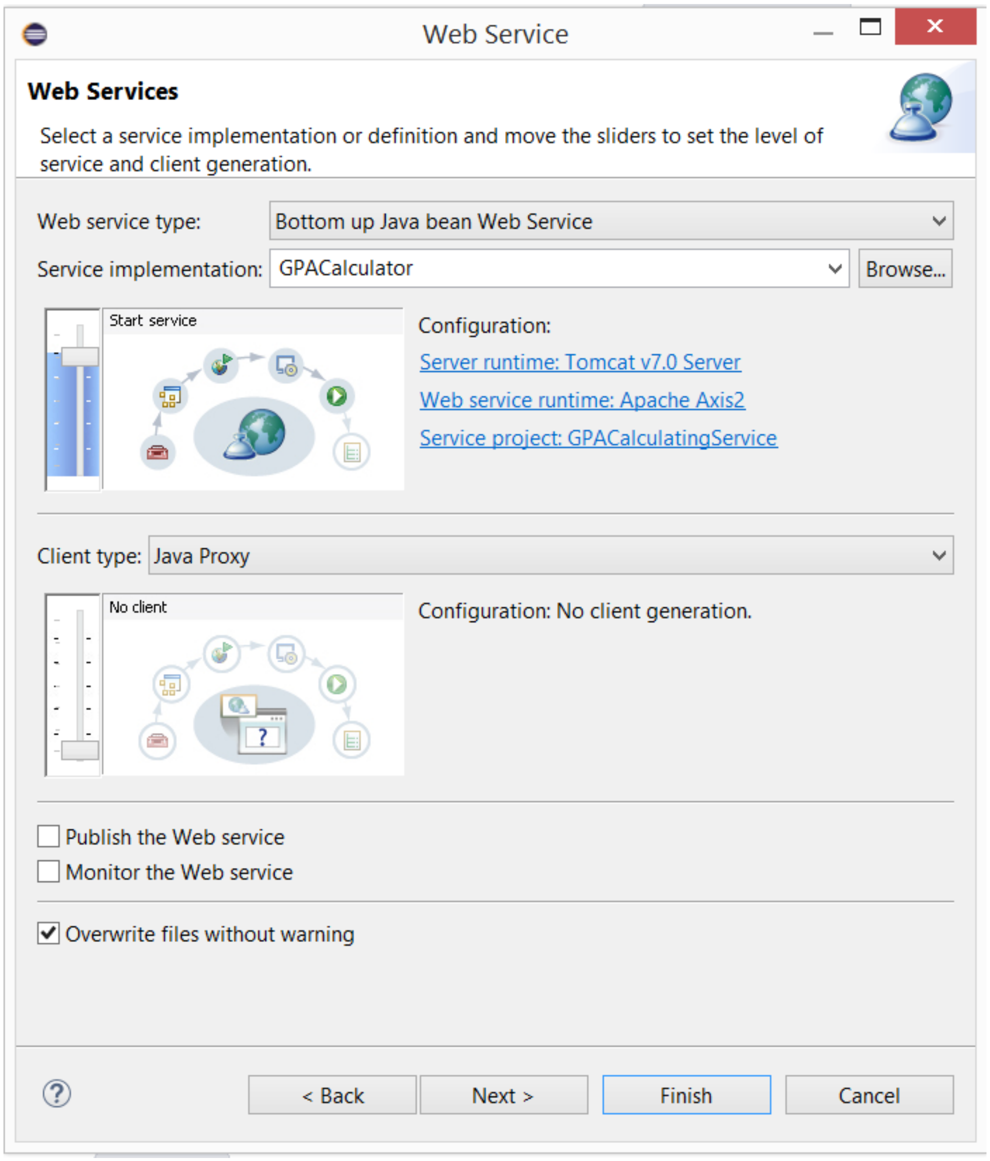
```java
public class GPACalculator {
	/*
	 * Initialize default values for class variables
	 */
	static int courseCount = 0;
	static Course courses[];
	double GPA = 0;

	/*
	 *  Function for calculating GPA
	 *  Requirement - Course data to be already recorded using getCourse() Method
	 */
	public double calculateGPA() {

		int endSemCATotalMarks = 100;

		GPA = 0;
		int totalCredits = 0;
		int totalCxG = 0;

		/*
		 *  For each of the course, calculate the CxG
		 *  where C - credits and
		 *  G - grade points earned for the course
		 */
		for(int i=0; i<courseCount; i++) {

			/*
			 *  Calculate the theory score as
			 *  End Semester Score * 0.6 + Continuous Assessment Score
			 */
			double endSemCAScore = courses[i].endSemesterMarks*0.6 +
						courses[i].continuousAssessment;

			/*
			 *   Based on the theory score, calculate the respective grade point
obtained for the course
			 */
			courses[i].endSemCAGradePoint =
assignGradePoint(endSemCAScore, endSemCATotalMarks);
			courses[i].endSemCACxG =
courses[i].endSemCAGradePoint*courses[i].theoryCredits;

			/*
			 *	Add the Term work and Practical/Oral Score
```

```
                         */
                        double pracsTWScore = (double)courses[i].practicalOralMarksScored
+
                                        (double)courses[i].termWorkMarksScored;
                        int pracsTWTotalMarks = courses[i].practicalOralTotalMarks +
                                        courses[i].termWorkTotalMarks;

                        /*
                         *  Based on the TW and Practical/Oral score, calculate the respective
grade point obtained for the course
                         */
                        courses[i].termWorkPracticalGradePoint                          =
assignGradePoint(pracsTWScore, pracsTWTotalMarks);
                        courses[i].termWorkPracticalCxG                                 =
courses[i].termWorkPracticalGradePoint *
                                        courses[i].termWorkPracticalCredits;

                        /*
                         *  Sum up the total credits taken and total CxG scored
                         */
                        totalCxG        +=      courses[i].termWorkPracticalCxG          +
courses[i].endSemCACxG;
                        totalCredits     +=     courses[i].termWorkPracticalCredits      +
courses[i].theoryCredits;

                }

                /*
                 * Final GPA is the division of CxG by Total Credits
                 */
                GPA = (double)totalCxG/(double)totalCredits;

                return GPA;
        }

        /*
         * Function to assign grade point for particular score
         */
        public int assignGradePoint(double marksScored, int totalMarks) {

                /*
                 *  Denotes the grading scheme followed by KJSCE
                 *
                 */
                double percentage = marksScored/(double)totalMarks;

                if(percentage >= 0.85)
                        return 10;
                else if(percentage >= 0.75)
                        return 9;
```

```java
		else if(percentage >= 0.70)
			return 8;
		else if(percentage >= 0.60)
			return 7;
		else if(percentage >= 0.55)
			return 6;
		else if(percentage >= 0.50)
			return 5;
		else if(percentage >= 0.45)
			return 4;

		return 0;

	}

	/*
	 *  Method to add details of a specific course to the recorded data
	 *  Requirement - Initialization using initializeCourses() method
	 */
	public String setCourse(String courseName, int theoryCredits, int termWorkPracticalCredits,
			int endSemesterMarks, int continuousAssessment, int termWorkMarksScored,
			int termWorkTotalMarks, int practicalOralMarksScored, int practicalOralTotalMarks) {

		/*
		 * Restrain from adding courses more than indicated while initialization
		 */
		if(courseCount >= courses.length) {
			return "More courses cannot be added";
		}

		/*
		 *  Call constructor to record course data
		 */
		Course c = new Course(courseName, theoryCredits, termWorkPracticalCredits,
				endSemesterMarks, continuousAssessment, termWorkMarksScored,
				termWorkTotalMarks, practicalOralMarksScored, practicalOralTotalMarks);

		courses[courseCount] = c;

		/*
		 * Increment the count of courses
		 */
		courseCount++;
		/*
		 *  Return suitable message
		 */
		return "Added subject " + courseName + ".";

	}

	/*
	 *  Initialization in order to start recording course data
	 */
	public void initializeCourses(int noOfCourses) {
		courseCount = 0;
		courses = new Course[noOfCourses];
	}

}

class Course {
	/*
	 * Attributes to store course details provided by client
	 */
	String courseName;
	int theoryCredits;
	int termWorkPracticalCredits;
	int endSemesterMarks;
	int continuousAssessment;
	int termWorkMarksScored;
	int termWorkTotalMarks;
	int practicalOralMarksScored;
	int practicalOralTotalMarks;

	/*
	 * Attributes to store calculated results
	 */
	int endSemCAGradePoint;
	int endSemCACxG;
	int termWorkPracticalGradePoint;
	int termWorkPracticalCxG;

	/*
	 * Constructor for initializing course information
	 */
	Course(String cn, int tc, int twpc, int es, int ca, int tws, int twt, int pracs, int pracst)
{
		this.courseName = cn;
		this.theoryCredits = tc;
		this.termWorkPracticalCredits = twpc;
		this.endSemesterMarks = es;
		this.continuousAssessment = ca;
		this.termWorkMarksScored = tws;
```

```
                this.termWorkTotalMarks = twt;
                this.practicalOralTotalMarks = pracst;
                this.practicalOralMarksScored = pracs;
        }


}
```

4. This java file is then used to implement a web service based on the dynamic web project. After the web service has been created, the code is run on a localhost server.


Web Service

Web Services

Select a service implementation or definition and move the sliders to set the level of service and client generation.

Web service type: Bottom up Java bean Web Service

Service implementation: GPACalculator

Start service

Configuration:
Server runtime: Tomcat v7.0 Server
Web service runtime: Apache Axis2
Service project: GPACalculatingService

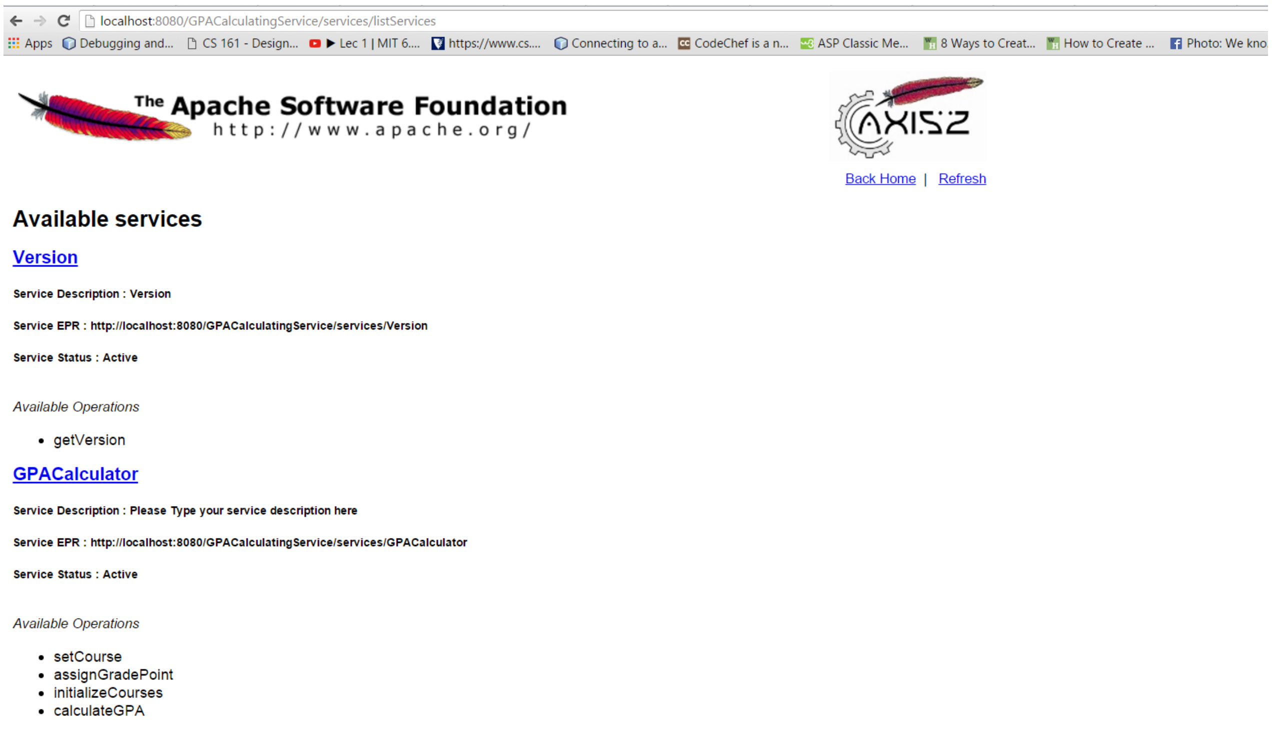
Client type: Java Proxy

No client

Configuration: No client generation.

☐ Publish the Web service
☐ Monitor the Web service
☑ Overwrite files without warning

5. Following screenshots demonstrate the successful deployment of the web services on the localhost server:



Available services

Version

Service Description : Version

Service EPR : http://localhost:8080/GPACalculatingService/services/Version

Service Status : Active

Available Operations

- getVersion

GPACalculator

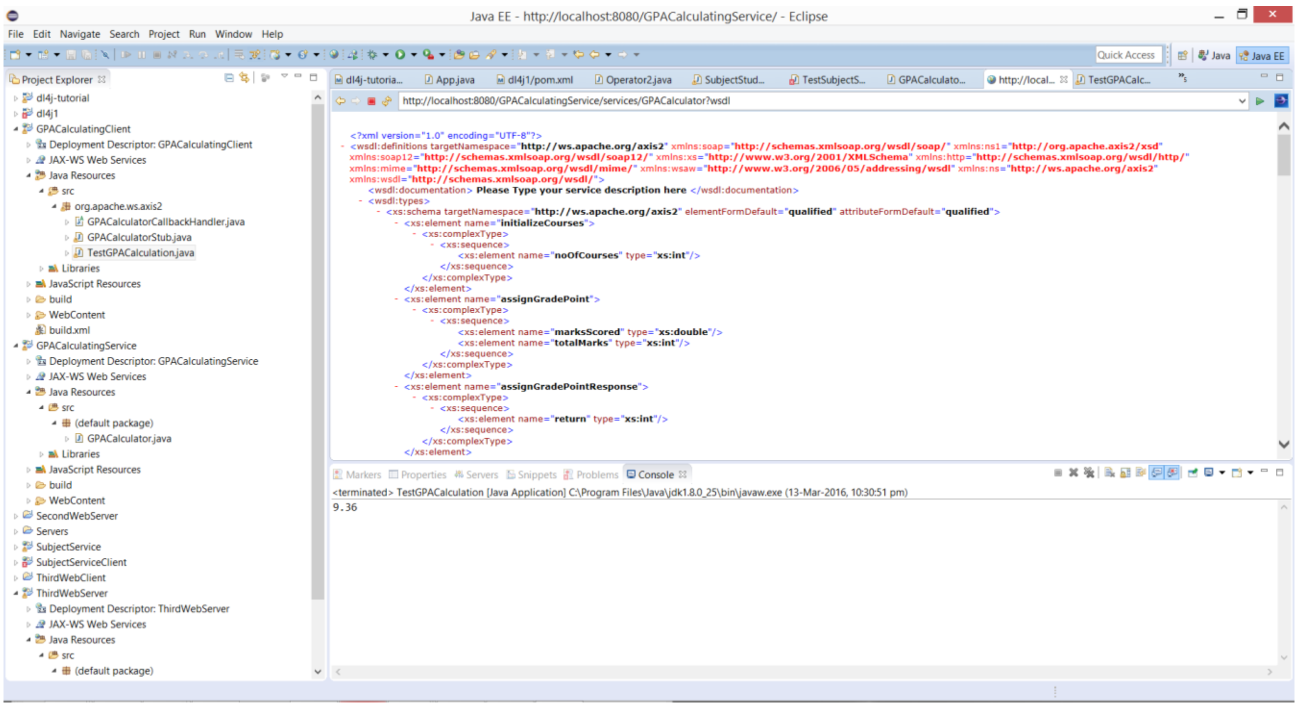
Service Description : Please Type your service description here

Service EPR : http://localhost:8080/GPACalculatingService/services/GPACalculator

Service Status : Active

Available Operations

- setCourse
- assignGradePoint
- initializeCourses
- calculateGPA



```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="http://ws.apache.org/axis2" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:ns1="http://org.apache.axis2/xsd"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:ns="http://ws.apache.org/axis2"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation> Please Type your service description here </wsdl:documentation>
  - <wsdl:types>
    - <xs:schema targetNamespace="http://ws.apache.org/axis2" elementFormDefault="qualified" attributeFormDefault="qualified">
      - <xs:element name="initializeCourses">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="noOfCourses" type="xs:int"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="assignGradePoint">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="marksScored" type="xs:double"/>
              <xs:element name="totalMarks" type="xs:int"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      - <xs:element name="assignGradePointResponse">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element name="return" type="xs:int"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
```

```
<terminated> TestGPACalculation [Java Application] C:\Program Files\Java\jdk1.8.0_25\bin\javaw.exe (13-Mar-2016, 10:30:51 pm)
9.36
```

6. The deployment of the web service provides us the WSDL document which can be used for understanding and accessing the methods provided by the web service. Following is the WSDL created for **GPACalculatingService**:

```
<?xml version="1.0" encoding="UTF-8"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns="http://ws.apache.org/axis2"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://ws.apache.org/axis2">
  <wsdl:documentation>
            Please Type your service description here
      </wsdl:documentation>
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://ws.apache.org/axis2">
      <xs:element name="initializeCourses">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="noOfCourses" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="setCourse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="courseName" nillable="true"
type="xs:string"/>
            <xs:element name="theoryCredits" type="xs:int"/>
            <xs:element name="termWorkPracticalCredits" type="xs:int"/>
            <xs:element name="endSemesterMarks" type="xs:int"/>
            <xs:element name="continuousAssessment" type="xs:int"/>
            <xs:element name="termWorkMarksScored" type="xs:int"/>
            <xs:element name="termWorkTotalMarks" type="xs:int"/>
            <xs:element name="practicalOralMarksScored" type="xs:int"/>
            <xs:element name="practicalOralTotalMarks" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="setCourseResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="calculateGPA">
        <xs:complexType>
          <xs:sequence/>
        </xs:complexType>
```

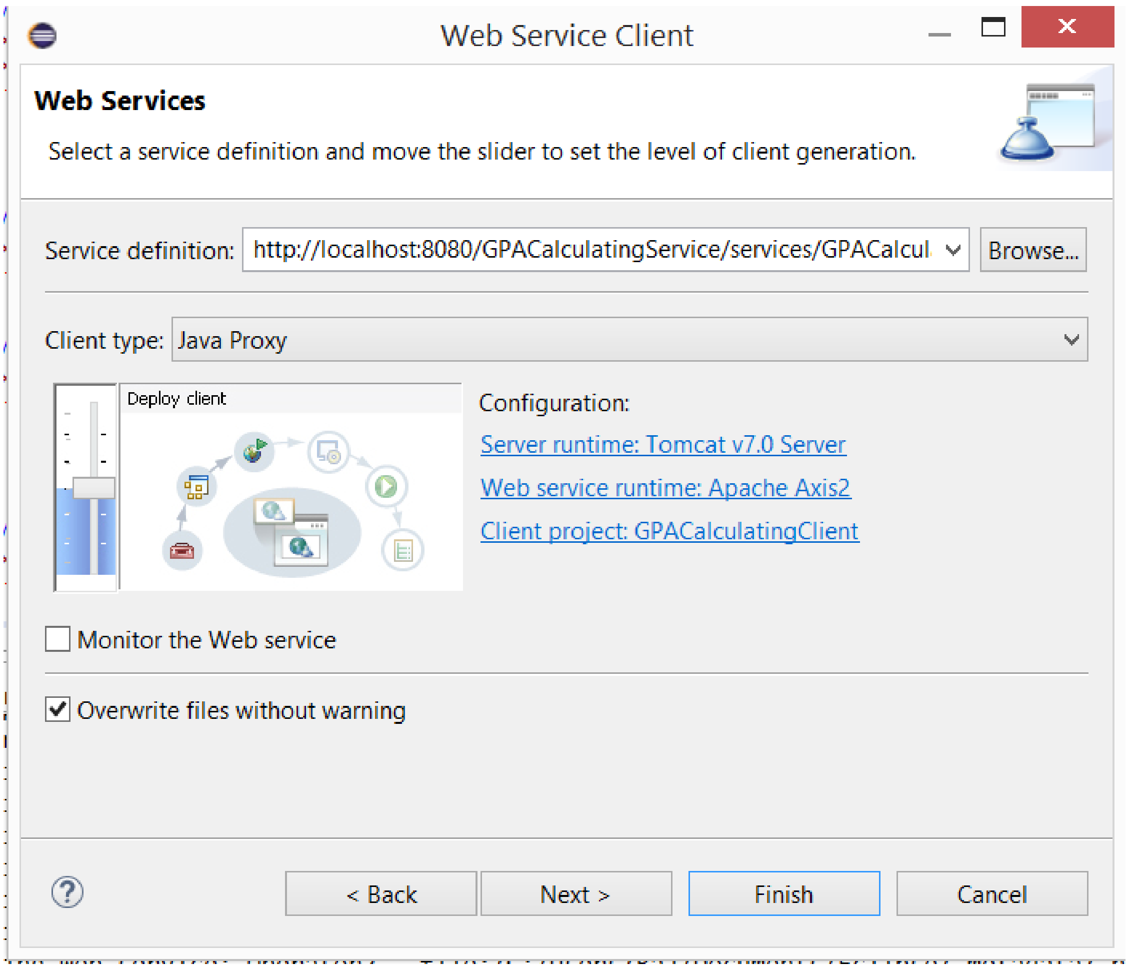
```
            </xs:element>
            <xs:element name="calculateGPAResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="return" type="xs:double"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="assignGradePoint">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="marksScored" type="xs:double"/>
                        <xs:element name="totalMarks" type="xs:int"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="assignGradePointResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="return" type="xs:int"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="setCourseRequest">
        <wsdl:part name="parameters" element="ns:setCourse"/>
    </wsdl:message>
    <wsdl:message name="setCourseResponse">
        <wsdl:part name="parameters" element="ns:setCourseResponse"/>
    </wsdl:message>
    <wsdl:message name="assignGradePointRequest">
        <wsdl:part name="parameters" element="ns:assignGradePoint"/>
    </wsdl:message>
    <wsdl:message name="assignGradePointResponse">
        <wsdl:part name="parameters" element="ns:assignGradePointResponse"/>
    </wsdl:message>
    <wsdl:message name="initializeCoursesRequest">
        <wsdl:part name="parameters" element="ns:initializeCourses"/>
    </wsdl:message>
    <wsdl:message name="calculateGPARequest">
        <wsdl:part name="parameters" element="ns:calculateGPA"/>
    </wsdl:message>
    <wsdl:message name="calculateGPAResponse">
        <wsdl:part name="parameters" element="ns:calculateGPAResponse"/>
    </wsdl:message>
    <wsdl:portType name="GPACalculatorPortType">
        <wsdl:operation name="setCourse">
            <wsdl:input message="ns:setCourseRequest" wsaw:Action="urn:setCourse"/>
```

```
      <wsdl:output message="ns:setCourseResponse"
wsaw:Action="urn:setCourseResponse"/>
    </wsdl:operation>
    <wsdl:operation name="assignGradePoint">
      <wsdl:input message="ns:assignGradePointRequest"
wsaw:Action="urn:assignGradePoint"/>
      <wsdl:output message="ns:assignGradePointResponse"
wsaw:Action="urn:assignGradePointResponse"/>
    </wsdl:operation>
    <wsdl:operation name="initializeCourses">
      <wsdl:input message="ns:initializeCoursesRequest"
wsaw:Action="urn:initializeCourses"/>
    </wsdl:operation>
    <wsdl:operation name="calculateGPA">
      <wsdl:input message="ns:calculateGPARequest"
wsaw:Action="urn:calculateGPA"/>
      <wsdl:output message="ns:calculateGPAResponse"
wsaw:Action="urn:calculateGPAResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="GPACalculatorSoap11Binding"
type="ns:GPACalculatorPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="setCourse">
      <soap:operation soapAction="urn:setCourse" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="assignGradePoint">
      <soap:operation soapAction="urn:assignGradePoint" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="initializeCourses">
      <soap:operation soapAction="urn:initializeCourses" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
    </wsdl:operation>
    <wsdl:operation name="calculateGPA">
      <soap:operation soapAction="urn:calculateGPA" style="document"/>
      <wsdl:input>
```

```
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:binding                                              name="GPACalculatorSoap12Binding"
type="ns:GPACalculatorPortType">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
      <wsdl:operation name="setCourse">
         <soap12:operation soapAction="urn:setCourse" style="document"/>
         <wsdl:input>
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap12:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="assignGradePoint">
         <soap12:operation soapAction="urn:assignGradePoint" style="document"/>
         <wsdl:input>
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap12:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="initializeCourses">
         <soap12:operation soapAction="urn:initializeCourses" style="document"/>
         <wsdl:input>
            <soap12:body use="literal"/>
         </wsdl:input>
      </wsdl:operation>
      <wsdl:operation name="calculateGPA">
         <soap12:operation soapAction="urn:calculateGPA" style="document"/>
         <wsdl:input>
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap12:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:binding name="GPACalculatorHttpBinding" type="ns:GPACalculatorPortType">
      <http:binding verb="POST"/>
      <wsdl:operation name="setCourse">
         <http:operation location="setCourse"/>
         <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="assignGradePoint">
            <http:operation location="assignGradePoint"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="initializeCourses">
            <http:operation location="initializeCourses"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
        </wsdl:operation>
        <wsdl:operation name="calculateGPA">
            <http:operation location="calculateGPA"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="GPACalculator">
        <wsdl:port                                  name="GPACalculatorHttpSoap11Endpoint"
binding="ns:GPACalculatorSoap11Binding">
            <soap:address
location="http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalcula
torHttpSoap11Endpoint/"/>
        </wsdl:port>
        <wsdl:port                                  name="GPACalculatorHttpSoap12Endpoint"
binding="ns:GPACalculatorSoap12Binding">
            <soap12:address
location="http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalcula
torHttpSoap12Endpoint/"/>
        </wsdl:port>
        <wsdl:port                                            name="GPACalculatorHttpEndpoint"
binding="ns:GPACalculatorHttpBinding">
            <http:address
location="http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalcula
torHttpEndpoint/"/>
        </wsdl:port>
    </wsdl:service>
```

</wsdl:definitions>

7. Further, a client needs to be created for testing whether the services can be accessed and used for calculating GPA on expected inputs. Hence, another dynamic web project **GPACalculatingClient** is created based on similar steps like the **GPACalculatingService** and a Web Service Client is generated for the WSDL created in the previous step.


Web Service Client

Web Services

Select a service definition and move the slider to set the level of client generation.

Service definition: http://localhost:8080/GPACalculatingService/services/GPACalcul

Client type: Java Proxy

Deploy client

Configuration:

Server runtime: Tomcat v7.0 Server

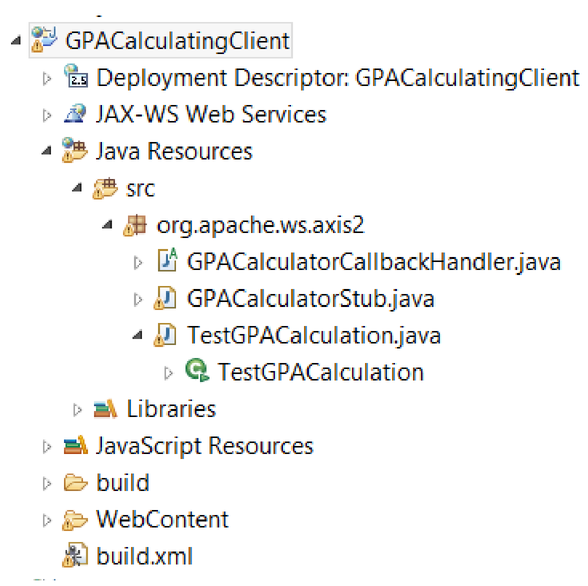
Web service runtime: Apache Axis2

Client project: GPACalculatingClient

☐ Monitor the Web service

☑ Overwrite files without warning

8. Now, we create a client java file **TestGPACalculation.java** for accessing the web service methods and using them for calculating GPA given the input data.

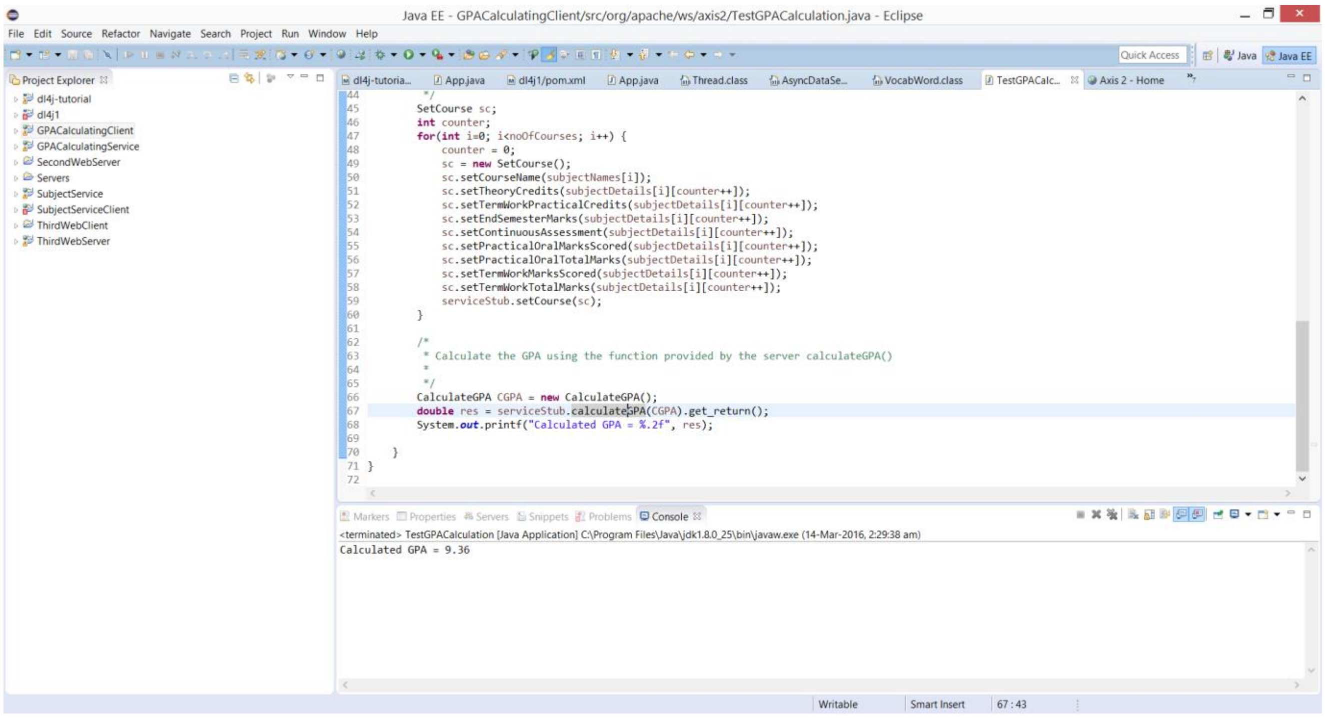

```
GPACalculatingClient
    Deployment Descriptor: GPACalculatingClient
    JAX-WS Web Services
    Java Resources
        src
            org.apache.ws.axis2
                GPACalculatorCallbackHandler.java
                GPACalculatorStub.java
                TestGPACalculation.java
                    TestGPACalculation
        Libraries
    JavaScript Resources
    build
    WebContent
    build.xml
```

9. The contents of the TestGPACalculation.java are provided below:

```
package org.apache.ws.axis2;

import org.apache.ws.axis2.GPACalculatorStub.CalculateGPA;
import org.apache.ws.axis2.GPACalculatorStub.SetCourse;
import org.apache.ws.axis2.GPACalculatorStub.InitializeCourses;

public class TestGPACalculation{

        public static void main(String args[]) throws Exception {
                // Create a stub object for using the services offered by the web server
                GPACalculatorStub serviceStub = new GPACalculatorStub();

                // Number of Courses
                int noOfCourses = 6;

                /*
                 * Subject names
                 */
                String subjectNames[] = {"DSP", "CSS", "AI", "SC", "BE Project", "NTAL"};

                /* Order for marks:
                 *  Theory Credits, Term Work and Practical Credits,
                 *  End Semester Marks, CA Marks, Practical/Oral Marks Scored
                 *  Practical/Oral Total Marks, Term Work Marks Scored,
                 *  Term Work Total Marks
                 */
                int subjectDetails[][] = { {4, 1, 81, 37,  0,  0, 23, 25},
                                           {4, 1, 70, 31, 22, 25, 24, 25},
                                           {4, 1, 83, 33, 24, 25, 24, 25},
                                           {4, 1, 72, 36, 23, 25, 23, 25},
                                           {0, 3,  0,  0, 48, 50, 47, 50},
                                           {0, 2,  0,  0, 45, 50, 20, 25} };
```

```
		/*
		 * Initialize the server to start recording course data
		 */
		InitializeCourses ic = new InitializeCourses();
		ic.setNoOfCourses(noOfCourses);
		serviceStub.initializeCourses(ic);

		/*
		 * Use the web service function getCourse() to record each course data
		 * on the server
		 */
		SetCourse sc;
		int counter;
		for(int i=0; i<noOfCourses; i++) {
			counter = 0;
			sc = new SetCourse();
			sc.setCourseName(subjectNames[i]);
			sc.setTheoryCredits(subjectDetails[i][counter++]);
			sc.setTermWorkPracticalCredits(subjectDetails[i][counter++]);
			sc.setEndSemesterMarks(subjectDetails[i][counter++]);
			sc.setContinuousAssessment(subjectDetails[i][counter++]);
			sc.setPracticalOralMarksScored(subjectDetails[i][counter++]);
			sc.setPracticalOralTotalMarks(subjectDetails[i][counter++]);
			sc.setTermWorkMarksScored(subjectDetails[i][counter++]);
			sc.setTermWorkTotalMarks(subjectDetails[i][counter++]);
			serviceStub.setCourse(sc);
		}

		/*
		 * Calculate the GPA using the function provided by the server
calculateGPA()
		 *
		 */
		CalculateGPA CGPA = new CalculateGPA();
		double res = serviceStub.calculateGPA(CGPA).get_return();
		System.out.printf("Calculated GPA = %.2f", res);

	}
}
```

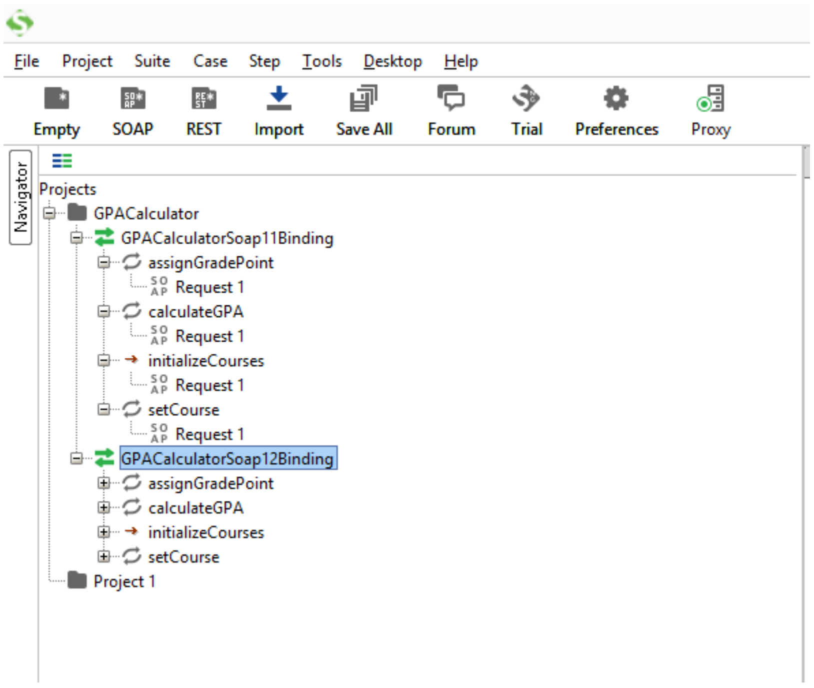
10. Correct output is obtained for the input data.


Java EE - GPACalculatingClient/src/org/apache/ws/axis2/TestGPACalculation.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
dl4j-tutorial
dl4j1
GPACalculatingClient
GPACalculatingService
SecondWebServer
Servers
SubjectService
SubjectServiceClient
ThirdWebClient
ThirdWebServer
dl4j-tutoria... App.java dl4j1/pom.xml App.java Thread.class AsyncDataSe... VocabWord.class TestGPACalc... Axis 2 - Home
Markers Properties Servers Snippets Problems Console
Writable Smart Insert 67 : 43

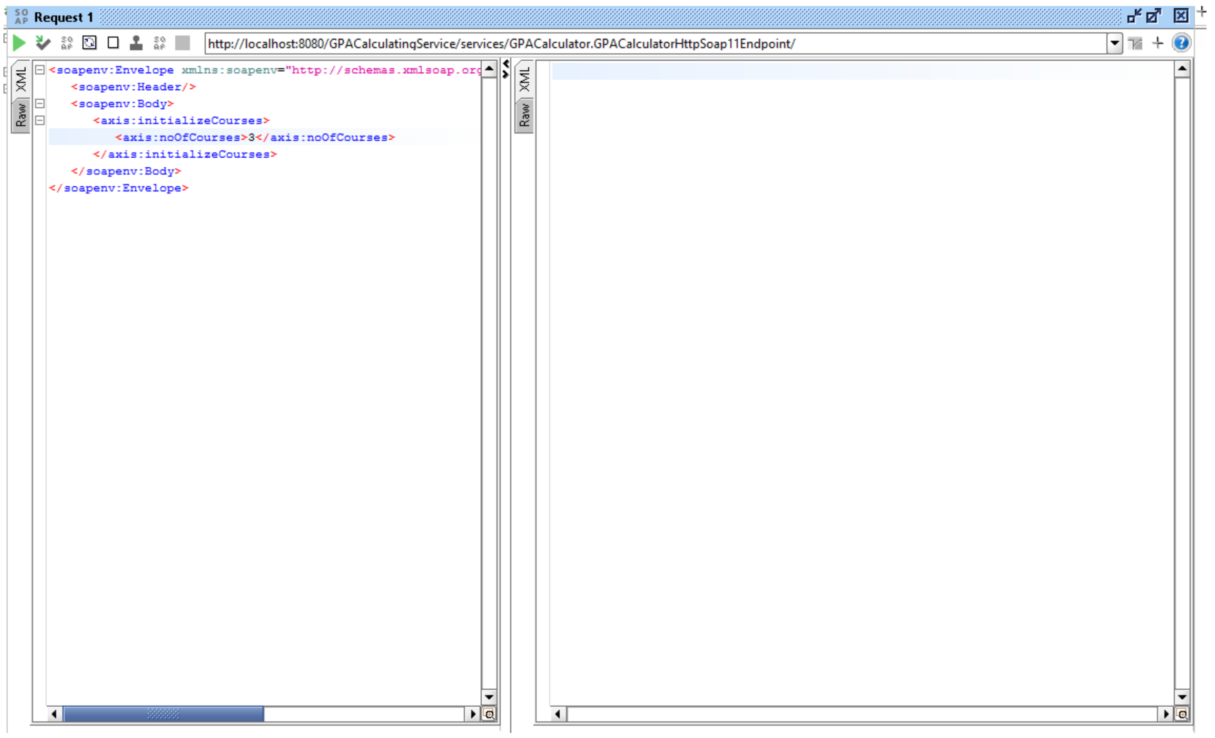
```java
         */
        SetCourse sc;
        int counter;
        for(int i=0; i<noOfCourses; i++) {
            counter = 0;
            sc = new SetCourse();
            sc.setCourseName(subjectNames[i]);
            sc.setTheoryCredits(subjectDetails[i][counter++]);
            sc.setTermWorkPracticalCredits(subjectDetails[i][counter++]);
            sc.setEndSemesterMarks(subjectDetails[i][counter++]);
            sc.setContinuousAssessment(subjectDetails[i][counter++]);
            sc.setPracticalOralMarksScored(subjectDetails[i][counter++]);
            sc.setPracticalOralTotalMarks(subjectDetails[i][counter++]);
            sc.setTermWorkMarksScored(subjectDetails[i][counter++]);
            sc.setTermWorkTotalMarks(subjectDetails[i][counter++]);
            serviceStub.setCourse(sc);
        }

        /*
         * Calculate the GPA using the function provided by the server calculateGPA()
         *
         */
        CalculateGPA CGPA = new CalculateGPA();
        double res = serviceStub.calculateGPA(CGPA).get_return();
        System.out.printf("Calculated GPA = %.2f", res);

    }
}
```

```
<terminated> TestGPACalculation [Java Application] C:\Program Files\Java\jdk1.8.0_25\bin\javaw.exe (14-Mar-2016, 2:29:38 am)
Calculated GPA = 9.36
```

11. The web service is also tested against SoapUI client. The different services offered are observed and their Soap XML requests are studied in order to check whether all the services produce the expected output on decided inputs. The screenshots below provide an overview of the testing done on SoapUI.

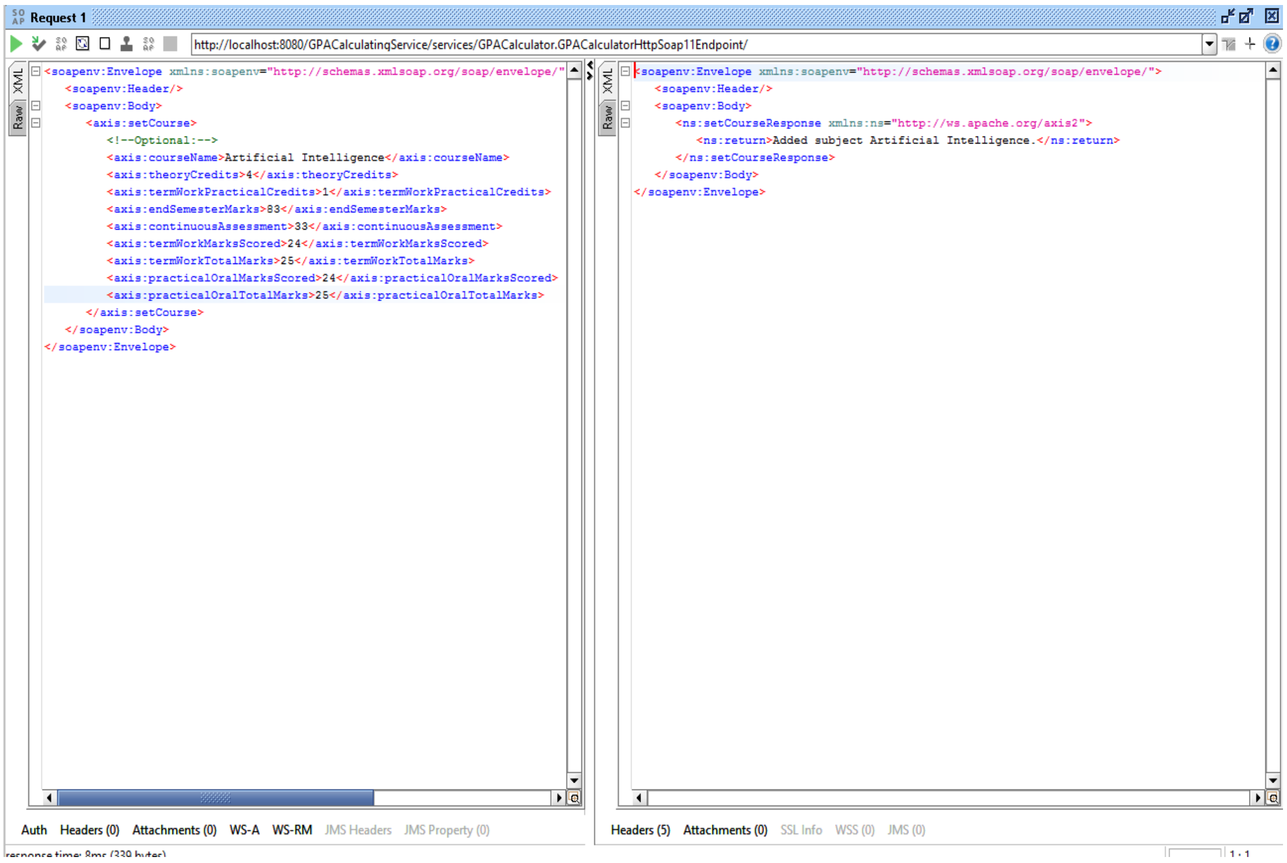

File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Navigator

```
Projects
  GPACalculator
    GPACalculatorSoap11Binding
      assignGradePoint
        Request 1
      calculateGPA
        Request 1
      initializeCourses
        Request 1
      setCourse
        Request 1
    GPACalculatorSoap12Binding
      assignGradePoint
      calculateGPA
      initializeCourses
      setCourse
  Project 1
```



http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org
   <soapenv:Header/>
   <soapenv:Body>
      <axis:initializeCourses>
         <axis:noOfCourses>3</axis:noOfCourses>
      </axis:initializeCourses>
   </soapenv:Body>
</soapenv:Envelope>
```



http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

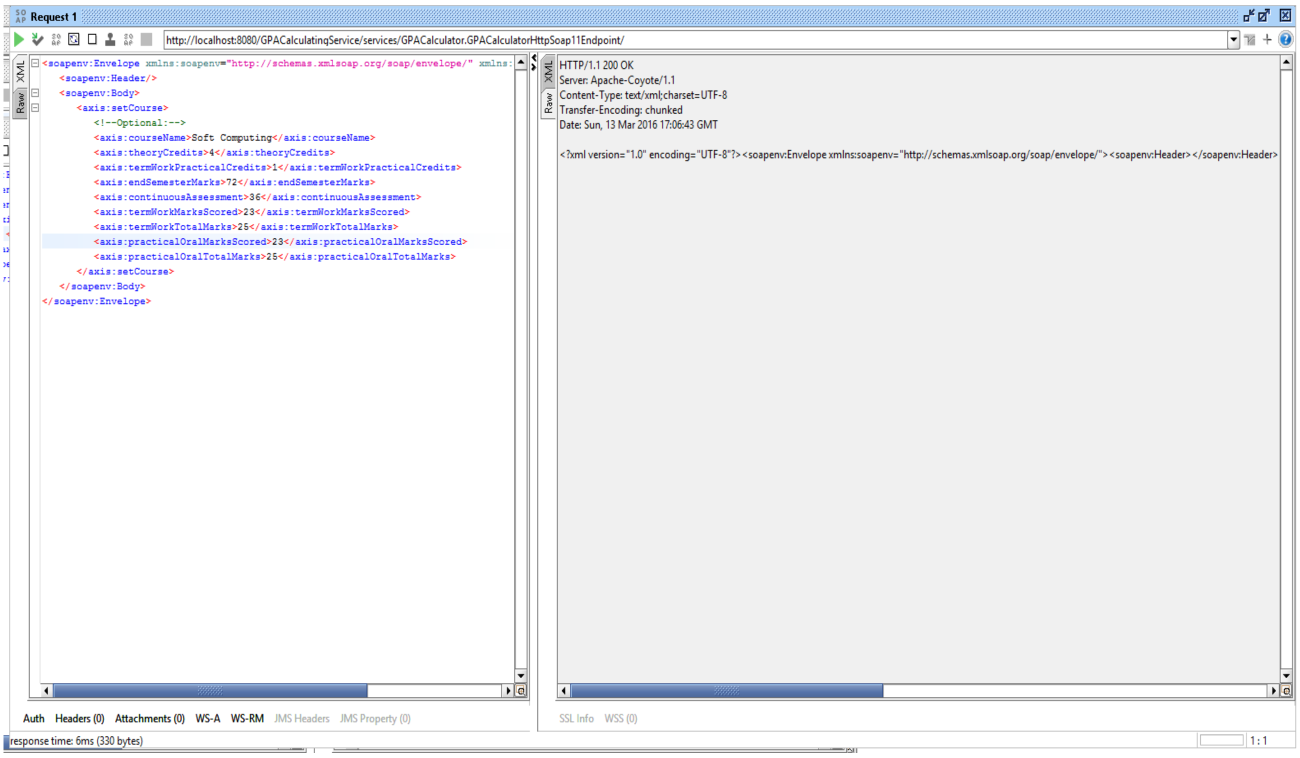
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   <soapenv:Header/>
   <soapenv:Body>
      <axis:setCourse>
         <!--Optional:-->
         <axis:courseName>Artificial Intelligence</axis:courseName>
         <axis:theoryCredits>4</axis:theoryCredits>
         <axis:termWorkPracticalCredits>1</axis:termWorkPracticalCredits>
         <axis:endSemesterMarks>83</axis:endSemesterMarks>
         <axis:continuousAssessment>33</axis:continuousAssessment>
         <axis:termWorkMarksScored>24</axis:termWorkMarksScored>
         <axis:termWorkTotalMarks>25</axis:termWorkTotalMarks>
         <axis:practicalOralMarksScored>24</axis:practicalOralMarksScored>
         <axis:practicalOralTotalMarks>25</axis:practicalOralTotalMarks>
      </axis:setCourse>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <ns:setCourseResponse xmlns:ns="http://ws.apache.org/axis2">
         <ns:return>Added subject Artificial Intelligence.</ns:return>
      </ns:setCourseResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

response time: 8ms (339 bytes)

1 : 1

-Raj Palkar: https://in.linkedin.com/in/rajpalkar



Request 1

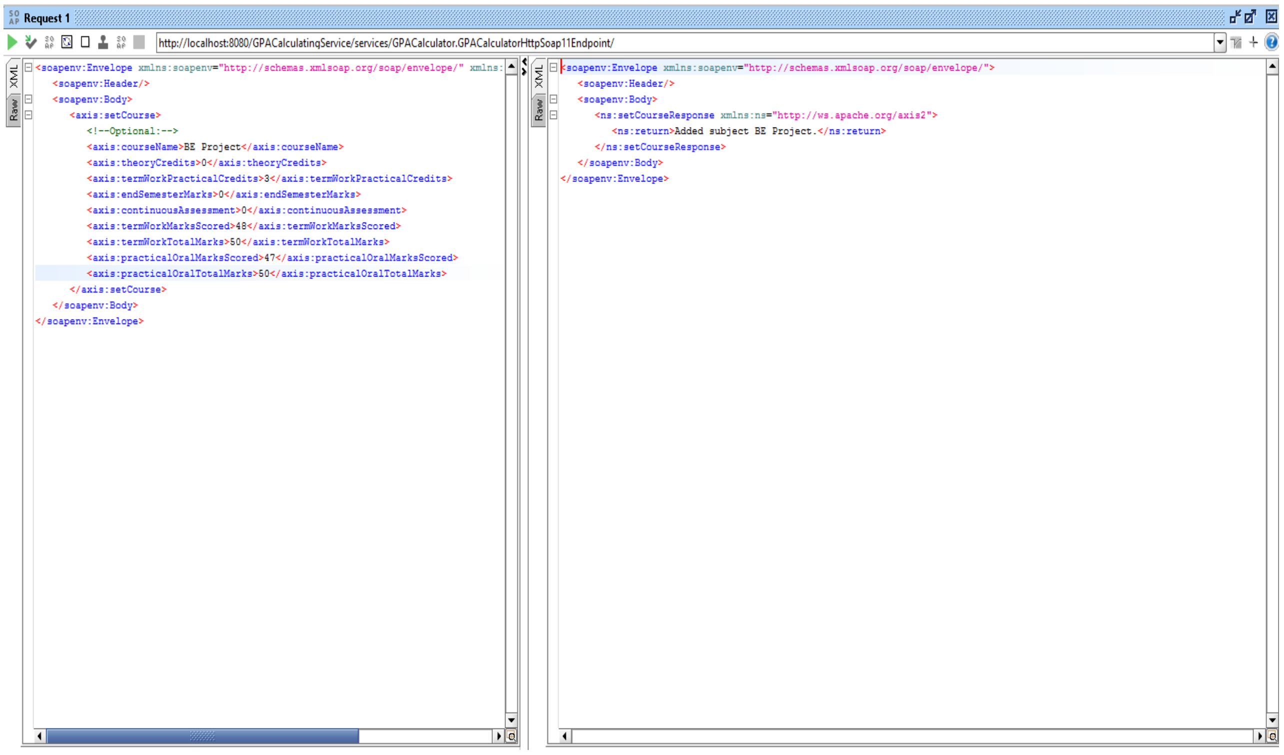
http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:
   <soapenv:Header/>
   <soapenv:Body>
      <axis:setCourse>
         <!--Optional:-->
         <axis:courseName>Soft Computing</axis:courseName>
         <axis:theoryCredits>4</axis:theoryCredits>
         <axis:termWorkPracticalCredits>1</axis:termWorkPracticalCredits>
         <axis:endSemesterMarks>72</axis:endSemesterMarks>
         <axis:continuousAssessment>36</axis:continuousAssessment>
         <axis:termWorkMarksScored>23</axis:termWorkMarksScored>
         <axis:termWorkTotalMarks>25</axis:termWorkTotalMarks>
         <axis:practicalOralMarksScored>23</axis:practicalOralMarksScored>
         <axis:practicalOralTotalMarks>25</axis:practicalOralTotalMarks>
      </axis:setCourse>
   </soapenv:Body>
</soapenv:Envelope>
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 13 Mar 2016 17:06:43 GMT

<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Header></soapenv:Header>
```

Auth Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)
SSL Info WSS (0)
1 : 1

response time: 6ms (330 bytes)



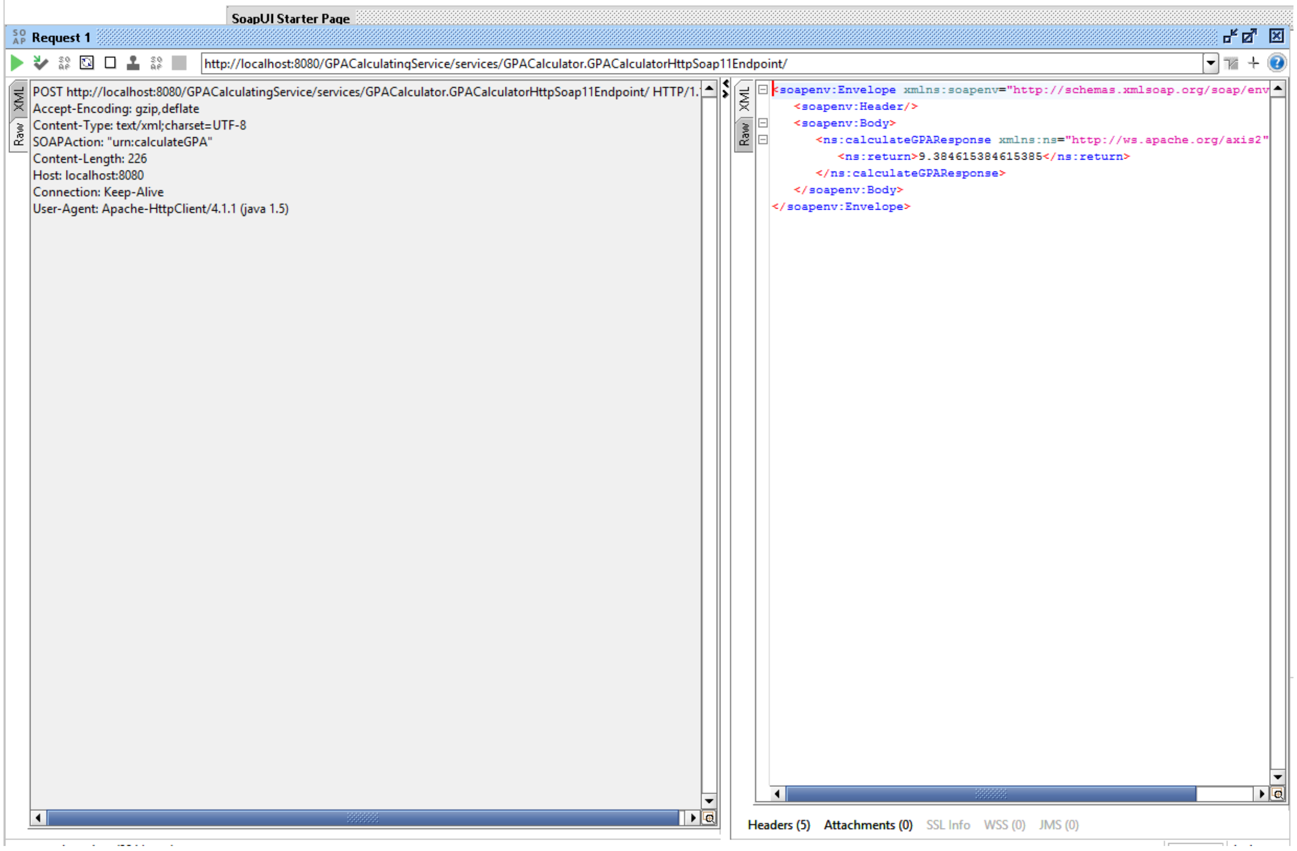
Request 1

http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:
   <soapenv:Header/>
   <soapenv:Body>
      <axis:setCourse>
         <!--Optional:-->
         <axis:courseName>BE Project</axis:courseName>
         <axis:theoryCredits>0</axis:theoryCredits>
         <axis:termWorkPracticalCredits>3</axis:termWorkPracticalCredits>
         <axis:endSemesterMarks>0</axis:endSemesterMarks>
         <axis:continuousAssessment>0</axis:continuousAssessment>
         <axis:termWorkMarksScored>48</axis:termWorkMarksScored>
         <axis:termWorkTotalMarks>50</axis:termWorkTotalMarks>
         <axis:practicalOralMarksScored>47</axis:practicalOralMarksScored>
         <axis:practicalOralTotalMarks>50</axis:practicalOralTotalMarks>
      </axis:setCourse>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <ns:setCourseResponse xmlns:ns="http://ws.apache.org/axis2">
         <ns:return>Added subject BE Project.</ns:return>
      </ns:setCourseResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

-Raj Palkar: https://in.linkedin.com/in/rajpalkar

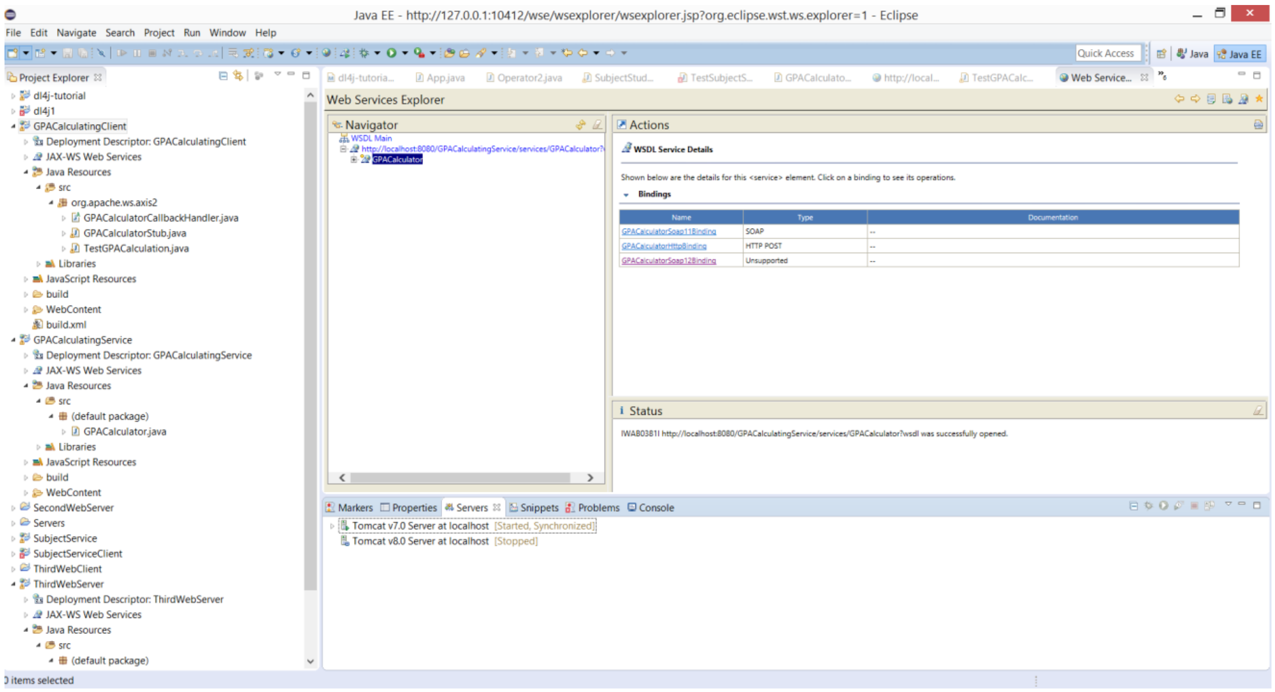


http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

```
POST http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/ HTTP/1.
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "urn:calculateGPA"
Content-Length: 226
Host: localhost:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/env
   <soapenv:Header/>
   <soapenv:Body>
      <ns:calculateGPAResponse xmlns:ns="http://ws.apache.org/axis2"
         <ns:return>9.384615384615385</ns:return>
      </ns:calculateGPAResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

12. Eclipse also allows a Testing Web Service Client and this feature also serves an exhausting service tool which provides detailed study of the web services just based on the WSDL created by the Web Service. It also helps us in understanding the contents of SOAP Request and Response Envelopes.



WSDL Service Details

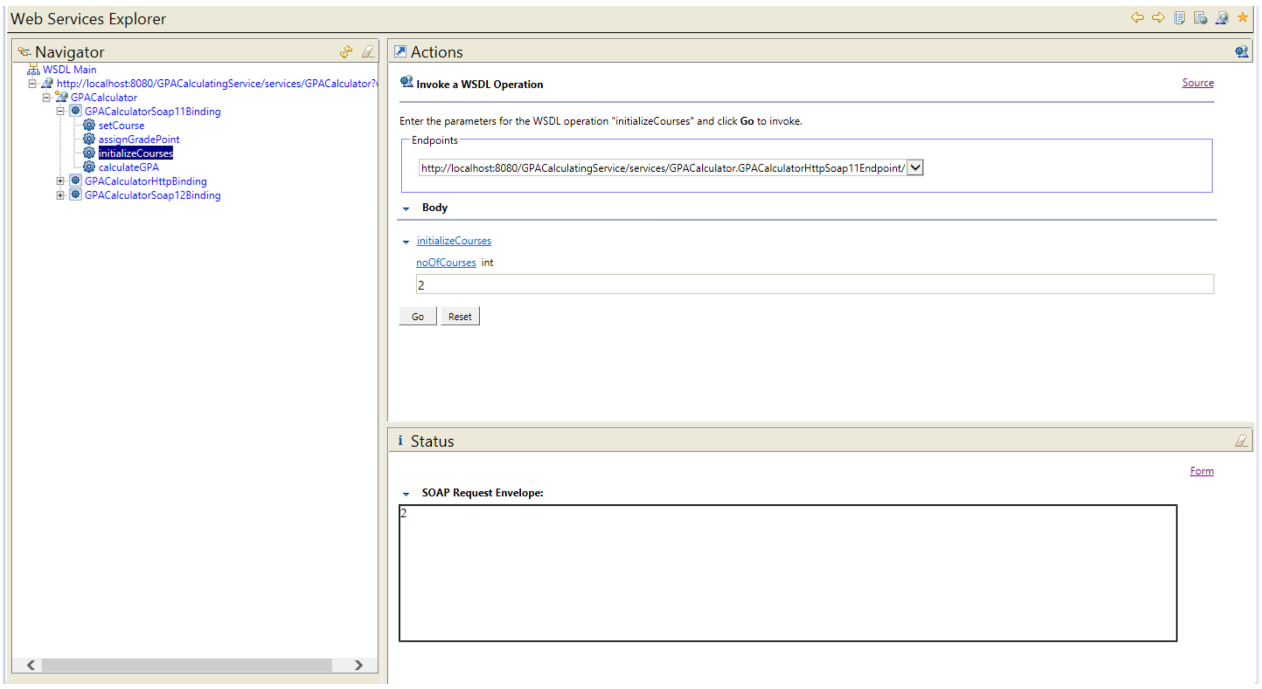
Shown below are the details for this <service> element. Click on a binding to see its operations.

Bindings

Name	Type	Documentation
GPACalculatorSoap11Binding	SOAP	--
GPACalculatorHttpBinding	HTTP POST	--
GPACalculatorSoap12Binding	Unsupported	--

IWAB0381I http://localhost:8080/GPACalculatingService/services/GPACalculator?wsdl was successfully opened.

-Raj Palkar: https://in.linkedin.com/in/rajpalkar


Web Services Explorer

Navigator
- WSDL Main
- http://localhost:8080/GPACalculatingService/services/GPACalculator?
  - GPACalculator
    - GPACalculatorSoap11Binding
      - setCourse
      - assignGradePoint
      - initializeCourses
      - calculateGPA
    - GPACalculatorHttpBinding
    - GPACalculatorSoap12Binding

Actions

Invoke a WSDL Operation

Enter the parameters for the WSDL operation "initializeCourses" and click **Go** to invoke.

Endpoints: http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

Body

initializeCourses

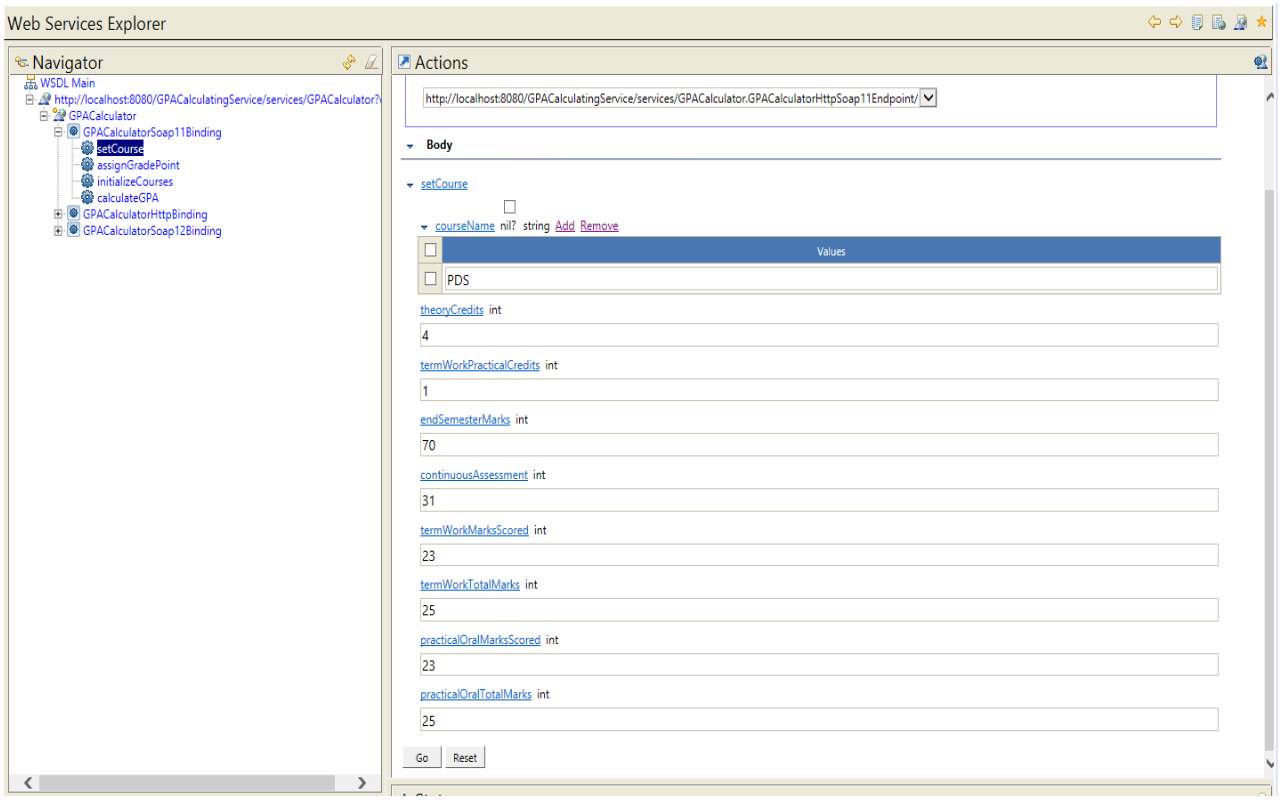
noOfCourses int

2

Status

SOAP Request Envelope:

```
2
```


Web Services Explorer

Navigator
- WSDL Main
- http://localhost:8080/GPACalculatingService/services/GPACalculator?
  - GPACalculator
    - GPACalculatorSoap11Binding
      - setCourse
      - assignGradePoint
      - initializeCourses
      - calculateGPA
    - GPACalculatorHttpBinding
    - GPACalculatorSoap12Binding

Actions

Endpoints: http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

Body

setCourse

courseName nil? string Add Remove

Values
PDS

theoryCredits int: 4

termWorkPracticalCredits int: 1

endSemesterMarks int: 70

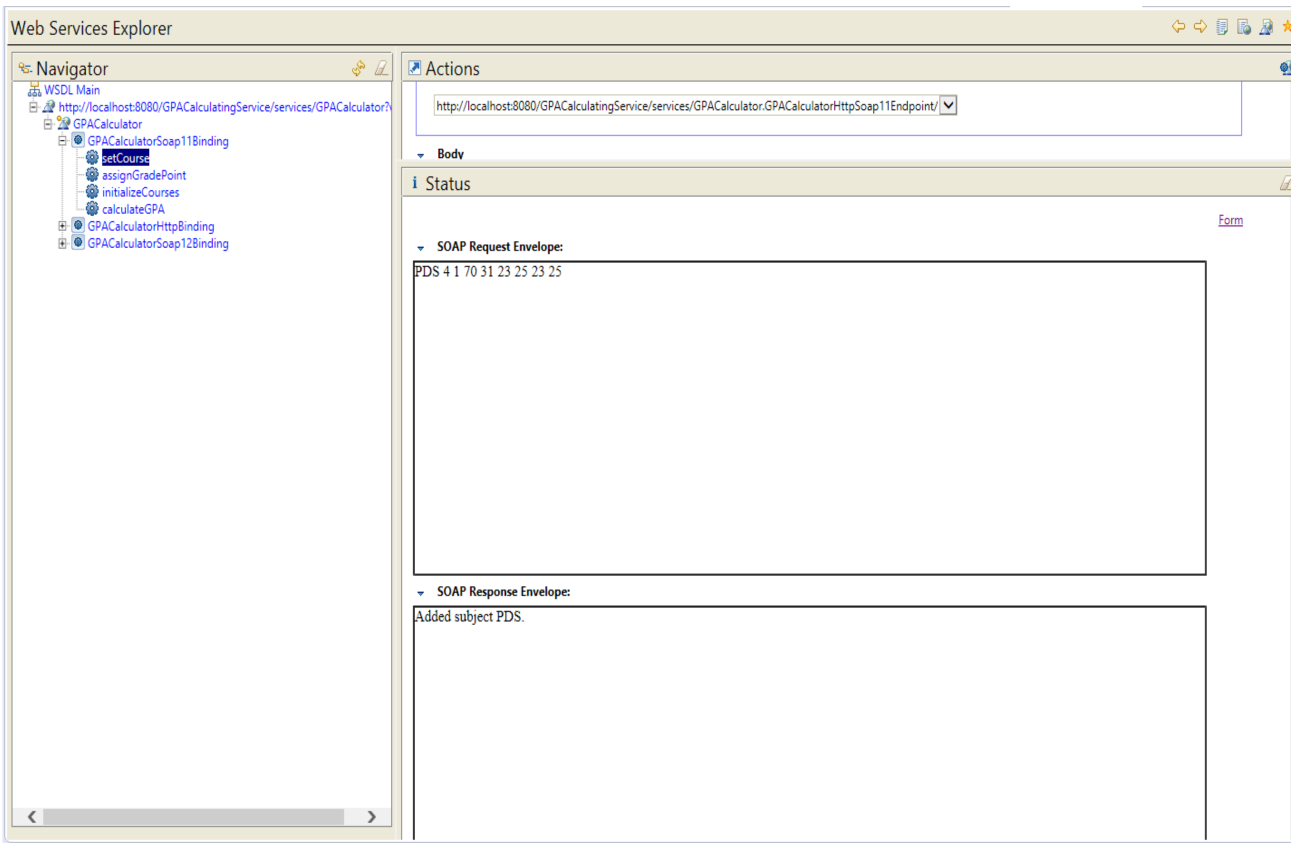
continuousAssessment int: 31

termWorkMarksScored int: 23

termWorkTotalMarks int: 25

practicalOralMarksScored int: 23

practicalOralTotalMarks int: 25



Web Services Explorer

Navigator

- WSDL Main
  - http://localhost:8080/GPACalculatingService/services/GPACalculator?
    - GPACalculator
      - GPACalculatorSoap11Binding
        - setCourse
        - assignGradePoint
        - initializeCourses
        - calculateGPA
      - GPACalculatorHttpBinding
      - GPACalculatorSoap12Binding

Actions

http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

Body

Status

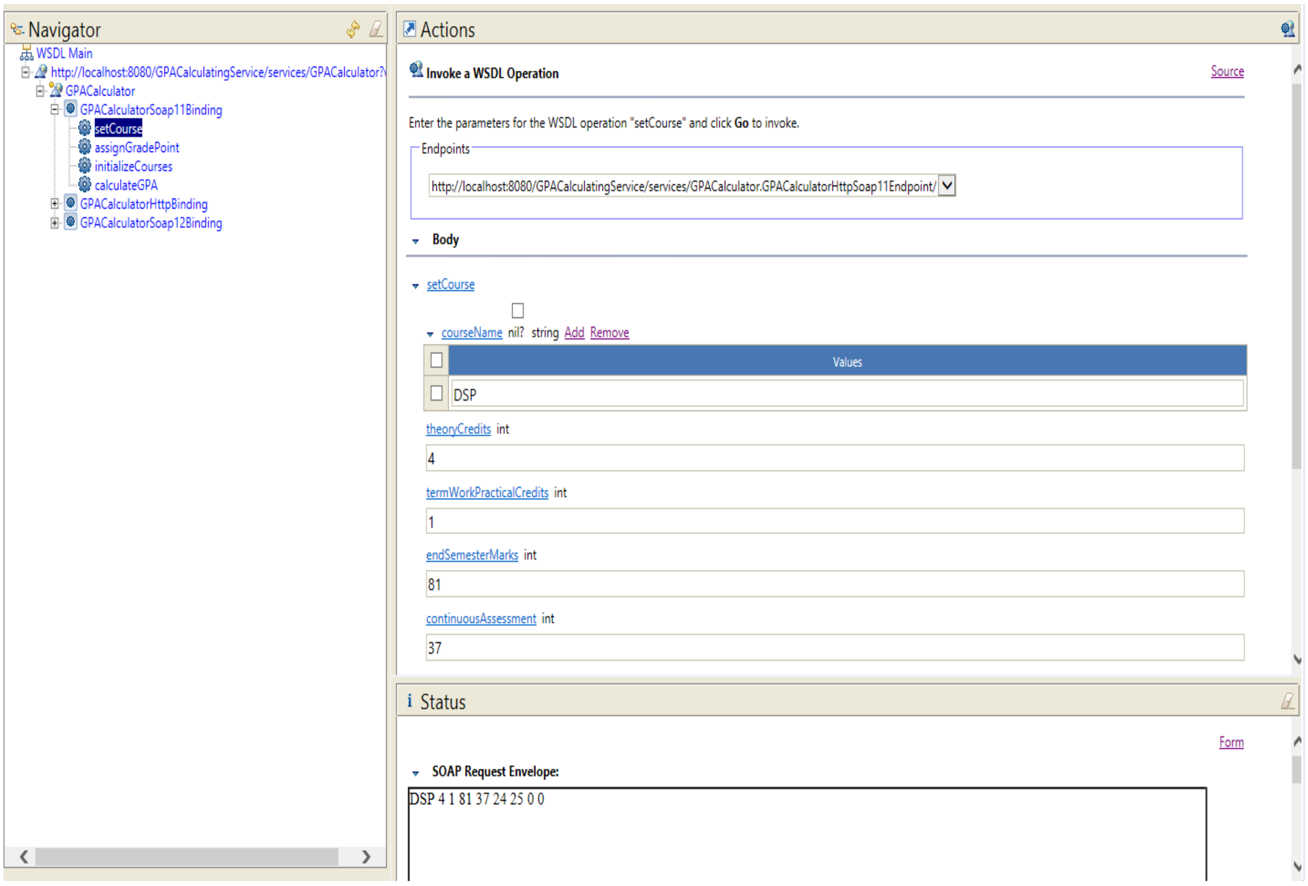
Form

SOAP Request Envelope:

```
PDS 4 1 70 31 23 25 23 25
```

SOAP Response Envelope:

```
Added subject PDS.
```



Navigator

- WSDL Main
  - http://localhost:8080/GPACalculatingService/services/GPACalculator?
    - GPACalculator
      - GPACalculatorSoap11Binding
        - setCourse
        - assignGradePoint
        - initializeCourses
        - calculateGPA
      - GPACalculatorHttpBinding
      - GPACalculatorSoap12Binding

Actions

Invoke a WSDL Operation — Source

Enter the parameters for the WSDL operation "setCourse" and click **Go** to invoke.

Endpoints

http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

Body

setCourse

courseName nil? string Add Remove

	Values
☐	DSP

theoryCredits int: 4

termWorkPracticalCredits int: 1

endSemesterMarks int: 81

continuousAssessment int: 37

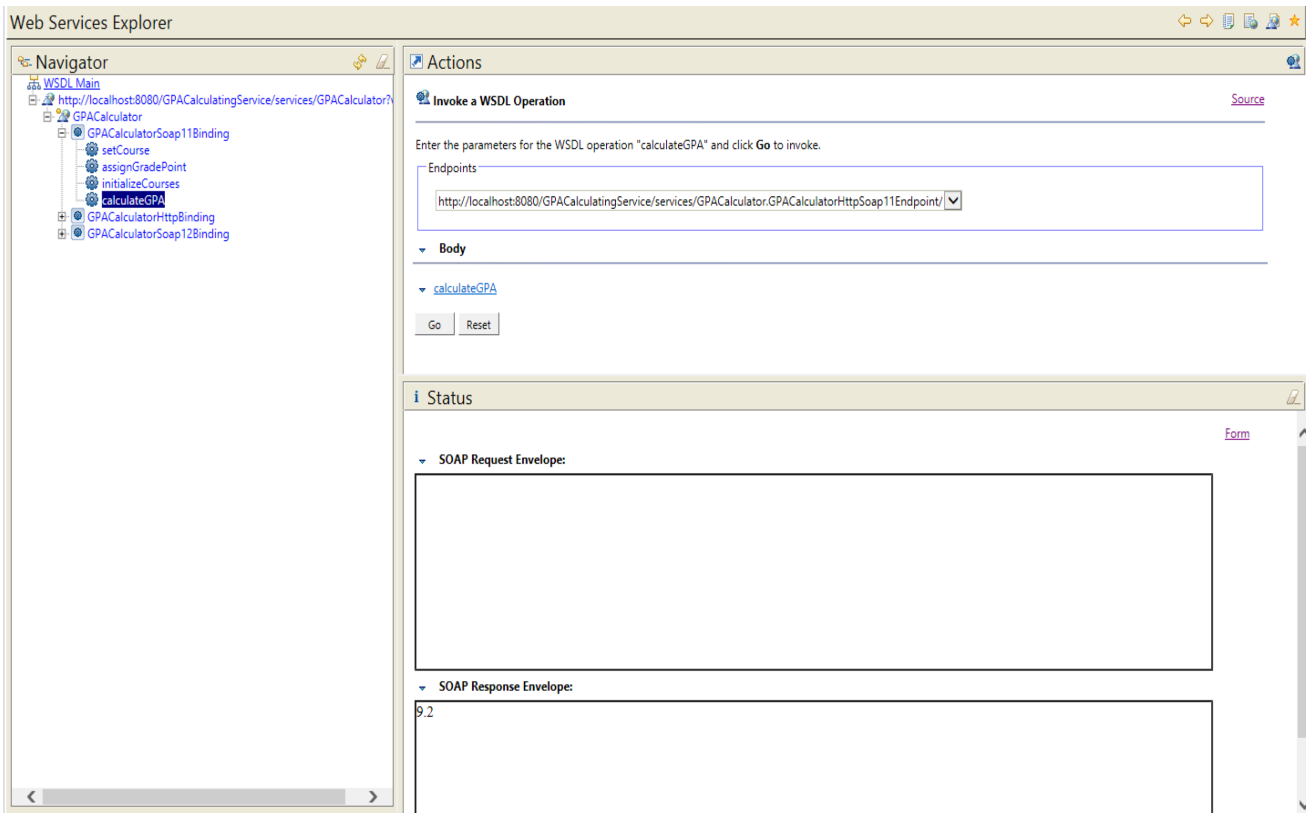
Status

Form

SOAP Request Envelope:

```
DSP 4 1 81 37 24 25 0 0
```

-Raj Palkar: https://in.linkedin.com/in/rajpalkar


Web Services Explorer

Navigator

- WSDL Main
  - http://localhost:8080/GPACalculatingService/services/GPACalculator?
    - GPACalculator
      - GPACalculatorSoap11Binding
        - setCourse
        - assignGradePoint
        - initializeCourses
        - calculateGPA
      - GPACalculatorHttpBinding
      - GPACalculatorSoap12Binding

Actions

Invoke a WSDL Operation Source

Enter the parameters for the WSDL operation "calculateGPA" and click **Go** to invoke.

Endpoints

http://localhost:8080/GPACalculatingService/services/GPACalculator.GPACalculatorHttpSoap11Endpoint/

Body

calculateGPA

Go Reset

Status

Form

SOAP Request Envelope:

SOAP Response Envelope:

```
9.2
```

-Raj Palkar: https://in.linkedin.com/in/rajpalkar

Conclusion

The project focuses on creating a web service that can provide functions for different clients without the need to worry about their implementation. The topic chosen for the web service was GPA calculation wherein the client or consumer inputs data regarding his/her scores in various courses to methods provided by the service and obtains the corresponding output. Three different kinds of clients were used for testing the service: custom program for testing the GPA calculation, Testing Web Service Client provided by Eclipse IDE and SoapUI client. The results obtained were verified with the expected output. The task of formatting returned data neatly as XML with multiple outputs has been kept as future work for the project.

References

1. Anatomy of a Web Service: XML, SOAP and WSDL for Platform-independent Data Exchange, http://www.webreference.com/authoring/web_service/index.html, Web Reference
2. XML SOAP, http://www.w3schools.com/xml/xml_soap.asp , W3Schools
3. SOAP, http://www.service-architecture.com/articles/web-services/soap.html , Service Architecture
4. XML Services, http://www.w3schools.com/xml/xml_services.asp , W3Schools
5. Web Services Explained, http://www.service-architecture.com/articles/web-services/web_services_explained.html , Service Architecture
6. SoapUI, https://www.soapui.org/
7. Apache Tomcat, https://tomcat.apache.org/index.html
8. Apache Axis2, https://en.wikipedia.org/wiki/Apache_Axis2, Wikipedia
9. Eclipse (Software), https://en.wikipedia.org/wiki/Eclipse_(software) **,** Wikipedia