# Ex No : 2 Implementation of SOAP and RESTful web services in Java

# Date :

# 

**Aim :**

To implement the SOAP and RESTFUL Web services in Java for our Application’s Signup Page along with its installation processes.

**Procedure:**

**SOAP:**

* Installation of Eclipse IDE for Java EE Developers.
* Installation Tomcat Apache Web Server.
* Setup Tomcat in Eclipse IDE.
* Create new Dynamic web project.
* Create a web server.

**REST:**

* Install necessary npm packages
* Install POSTMAN in Google chrome extension
* Set “GET” ,enter URL and click send
* Set “POST” and click send 6. Notified as 201 so it sends and receives the request

# Web Services:

A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.

# SOAP Web Services:

SOAP is an XML based industry standard protocol for designing and developing web services. Since it's XML based, it's platform and language independent. So our server can be based on JAVA and client can be on .

# RESTFUL Web Services:

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

**SOAP IMPLEMENTATION**

**CODE:**

**INTERFACE :-**

**package** MySoapService;

**public** **interface** RegNo **extends** java.rmi.Remote {

**public** **void** setRegNo(String regno);

**public** String getRegNo();

}

**CLASS FILE:-**

**package** MySoapService;

**public** **class** DisplayRegNo **implements** RegNo

{

**public** String regNo = **null**;

**public** **void** setRegNo(String regno)

{

**this**.regNo = regno;

}

**public** String getRegNo()

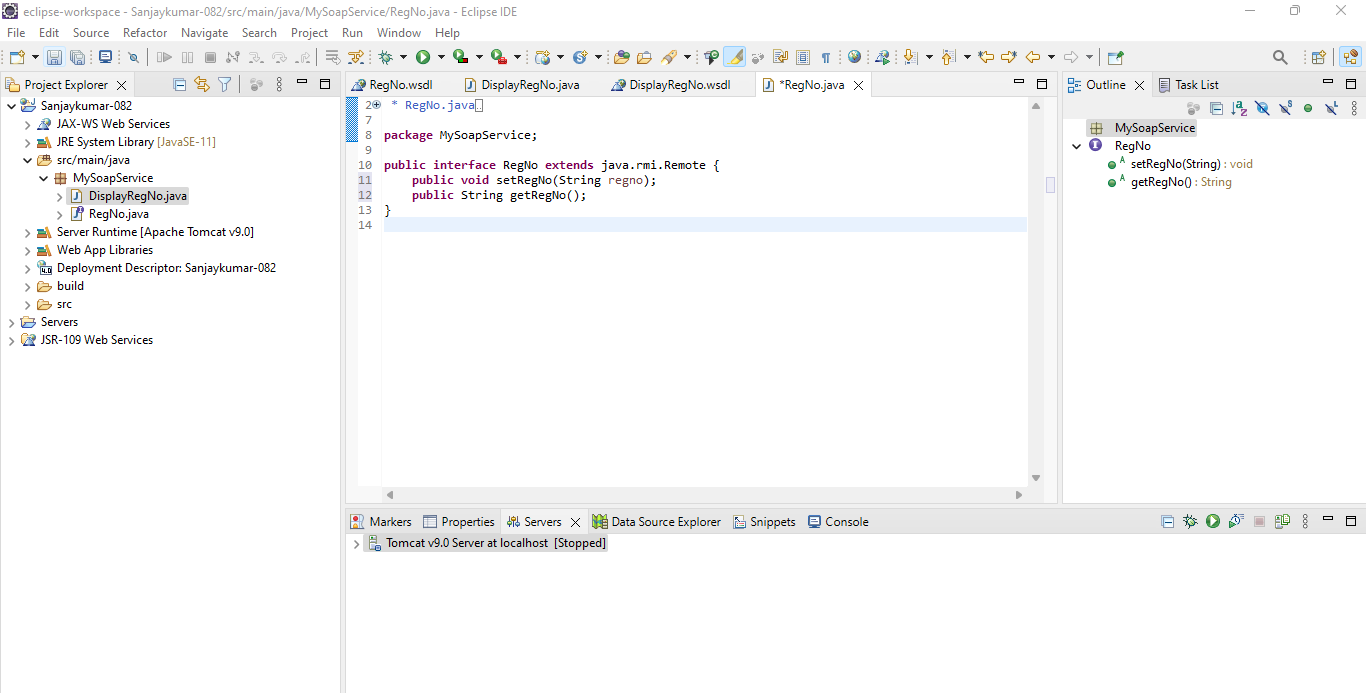
{

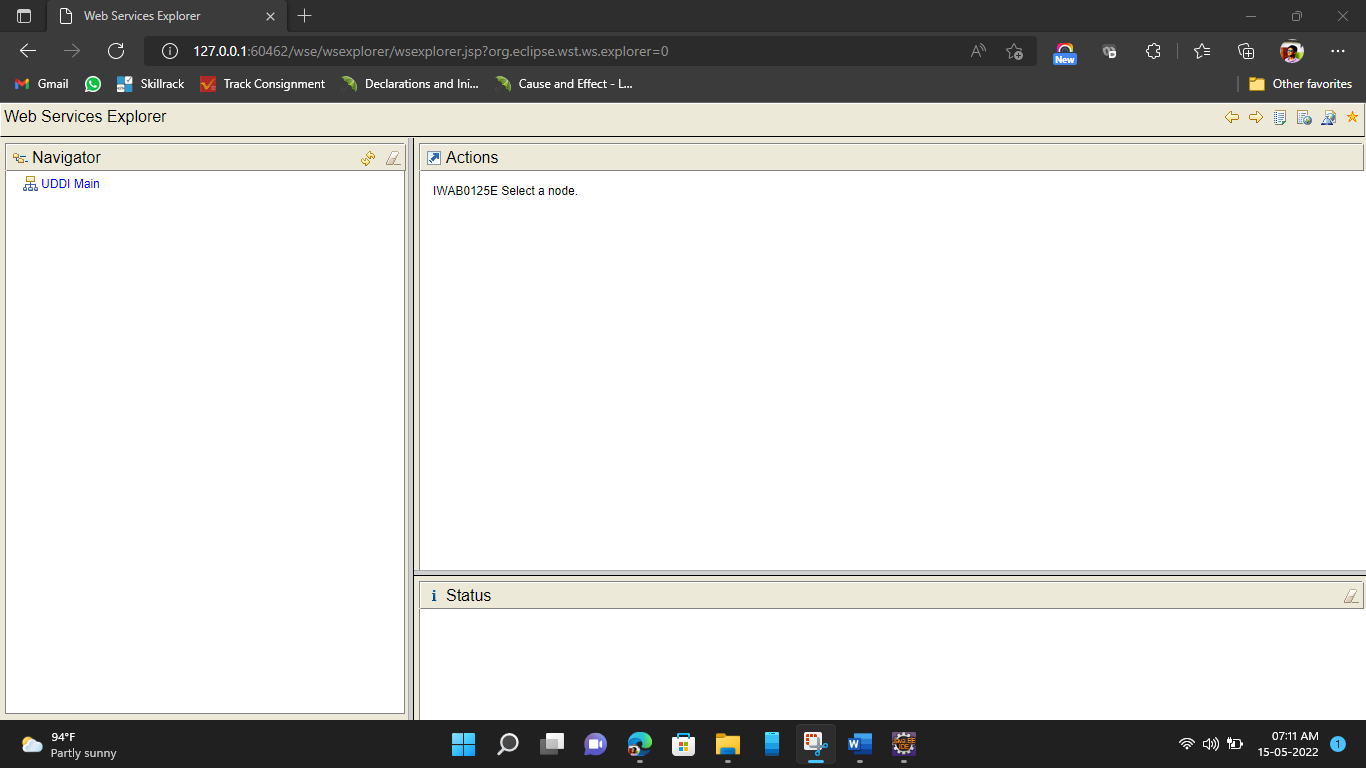
**return** regNo;

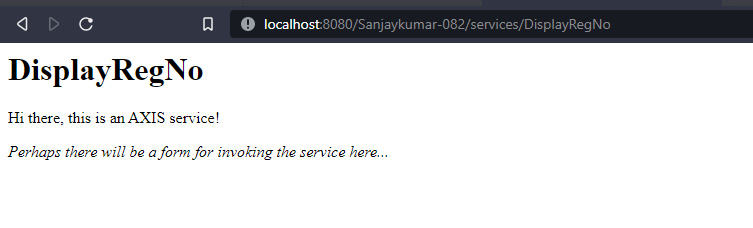
}

}

**OUTPUT:**

****



****

**REST IMPLEMENTATION**

**CODE:**

const express = require('express');

const bp = require('body-parser');

const app = express();

app.use(express.json());

app.use(bp.json());

app.use(bp.urlencoded({extended:false}));

var teams = [

{

team\_id: 1,

name: "Staraptors",

size: 5,

subject: "18ES690",

members: ["Sanjaykumar S", "Jeswin W", "Aravinth S", "Jeyaganesh A V", "Rahul Hariesh B"]

},

{

team\_id: 2,

name: "Tech Tuners",

size: 3,

subject: "18ES690",

members: ["Velmurugan M", "Jaga Ganesh D", "Lokkeshwaran S"]

}

]

var server = app.listen(4846, ()=>{

console.log("Server is listening at 4846!");

})

//GET

//get all

app.get('/teams', (req, res)=>{

res.send(teams);

})

//get particular team

app.get('/team', (req, res)=>{

const id = req.query.id;

for(let i=0; i<teams.length; i++)

{

if(teams[i].team\_id == id)

{

var team = teams[i];

break;

}

}

res.send(team);

})

//DELETE

app.delete('/remove\_team',(req, res)=>{

const id = req.query.id;

for(let i=0; i<teams.length; i++)

{

if(teams[i].team\_id == id)

{

var team = teams[i];

break;

}

}

teams.splice(team,1);

res.send(teams);

})

//PUT

app.put('/update\_teamname', (req, res)=>{

const id = req.query.id;

const team\_name = req.query.team\_name;

for(let i = 0; i<teams.length; i++)

{

if(teams[i].team\_id == id)

{

teams[i]["name"] = team\_name;

}

}

res.send(teams);

})

//POST

app.post('/team\_members', (req, res)=>{

const id = req.body.id;

console.log(id)

for(let i=0; i<teams.length; i++)

{

if(teams[i].team\_id == id)

{

var team = teams[i];

break;

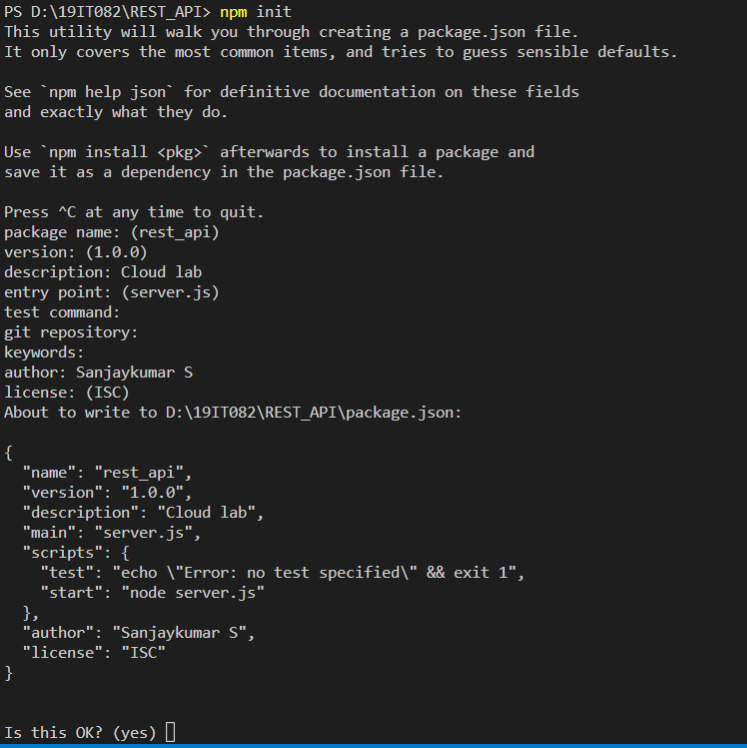
}

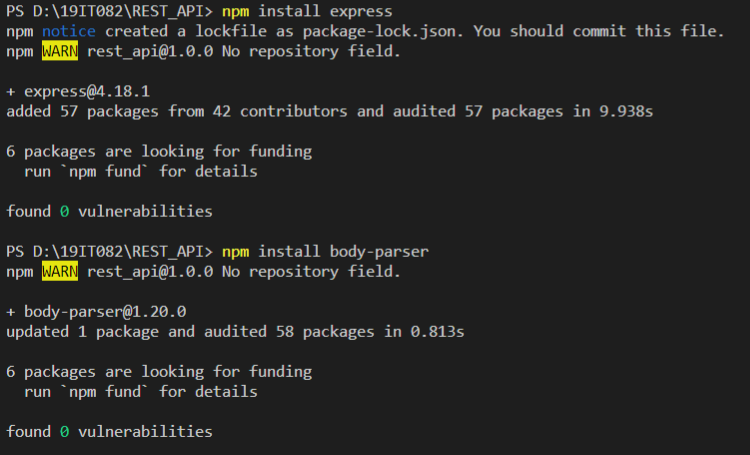
}

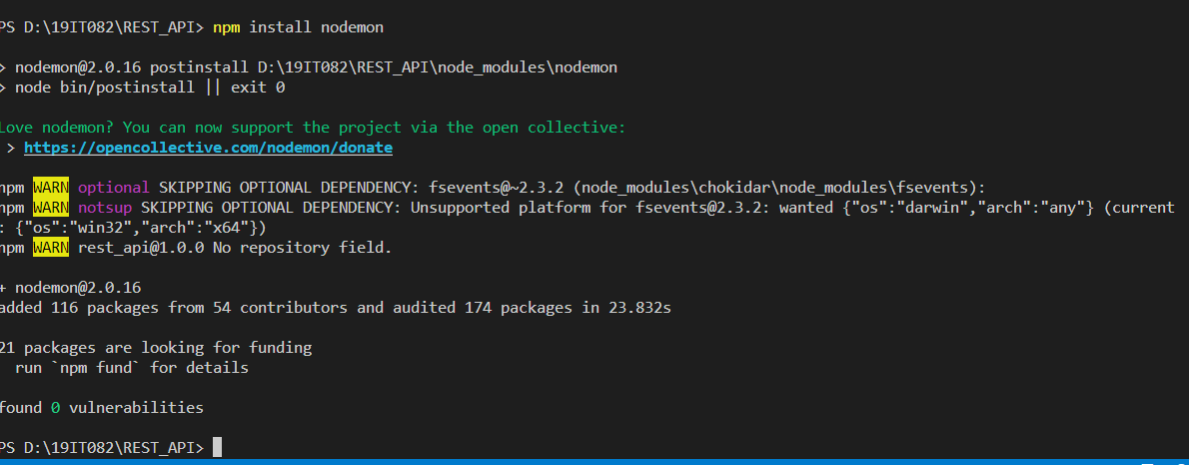
res.send(team. members);

})

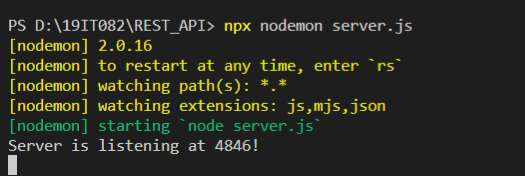
**Adding packages:-**



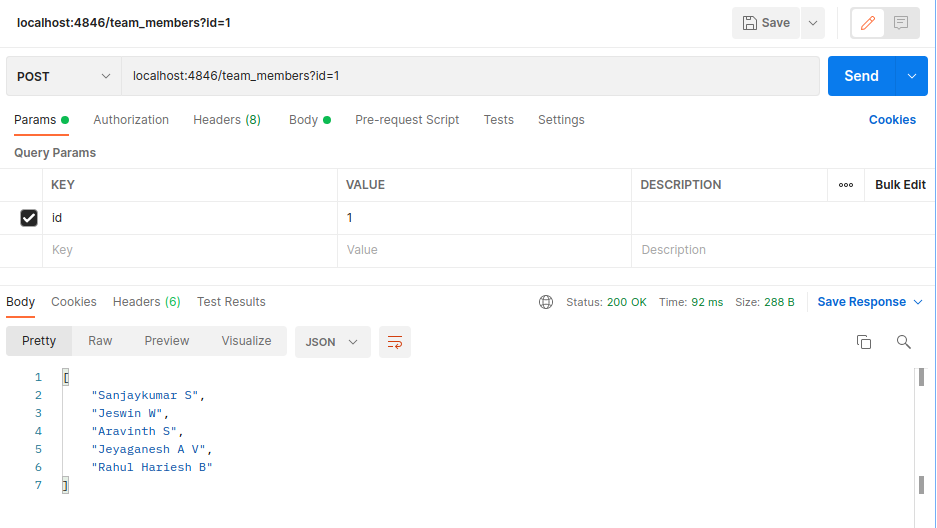




**OUTPUT:-**

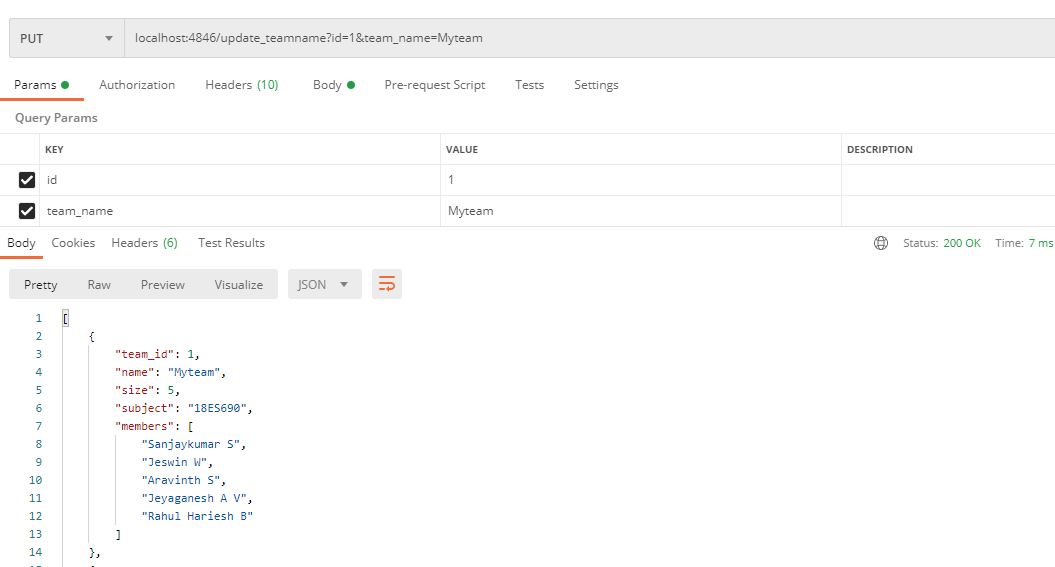


**POST**

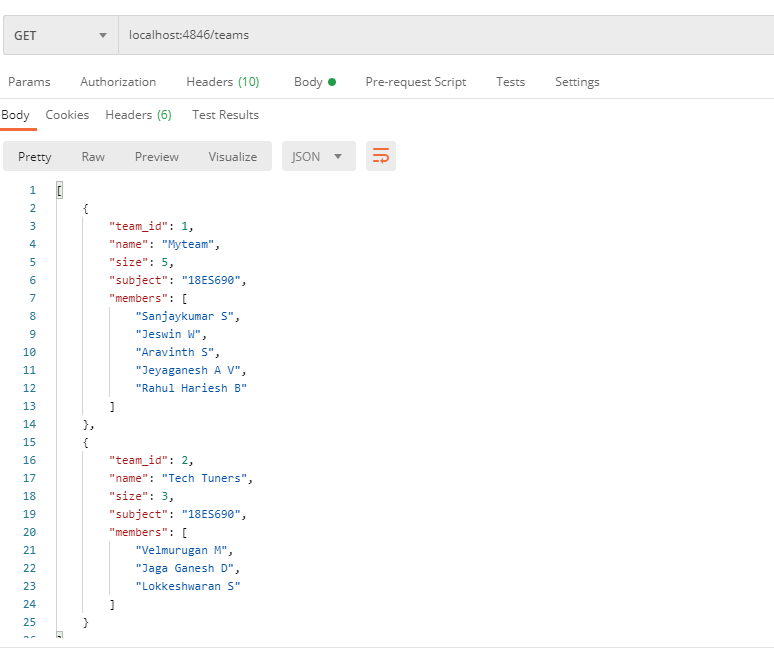


‘

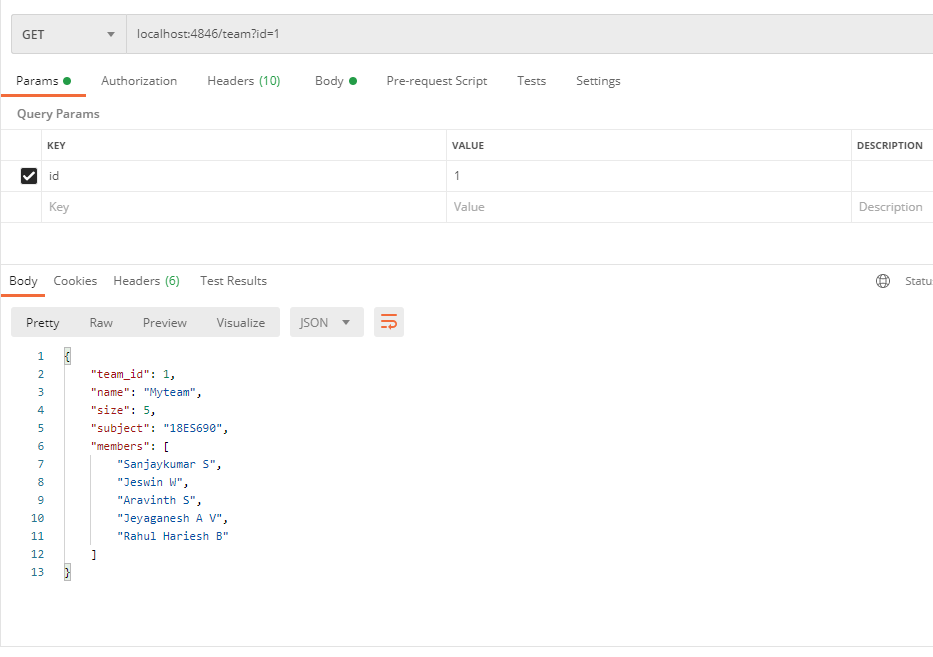
**PUT**



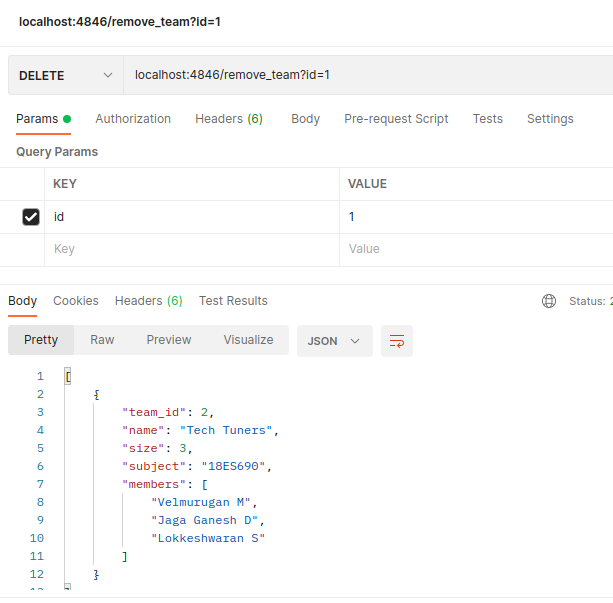
**GET**



**GET BY ID**



**DELETE:**



**RESULT:**

Thus, the SOAP and RESTful web services are implemented and the result is verified.