# EXPERIMENT-1

**a) TCP Socket Programming (UNIX)**
**Aim:** Implement concurrent echo client-server application.
**Tools/ Apparatus:** Unix/Linux C Programming Environment
**Procedure:**
1. Write a server (TCP) C Program that opens a listening socket and waits to serve client
2. Write a client (TCP) C Program that connects with the server program knowing IP address and port number.
3. Get the input string from console on client and send it to server, server echoes back that string to client.

Program:

==========

**ECHOTCPSERVER.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int listensd,clientsd;
char buffer[MAXLINESIZE+1];

struct sockaddr_in servaddr;
struct sockaddr_in peeraddr;
int noBytesRead=0;
void processClient(int);
int main()
{
        //create socket
        if((listensd=socket(AF_INET,SOCK_STREAM,0))<0)
        {
                fprintf(stderr,"Cannont create socket\n");
                exit(-1);
        }

        //Initialize socket address structure
        bzero(&servaddr,sizeof(servaddr));
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);
```

```
	servaddr.sin_addr.s_addr=htonl(INADDR_ANY);  //INADDR_ANY is wildcard returns local address when not connected

	//bind socket
	if(bind(listensd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
	{
		fprintf(stderr,"Error in bind\n");
		exit(-1);
	}

	//make socket listening socket
	if(listen(listensd,5)<0)
	{
		fprintf(stderr,"Error in listen\n");
		exit(-1);
	}

	//wait for client connection
	for(;;)
	{
		clientsd=accept(listensd,(struct sockaddr*)NULL,NULL);
		if(fork()==0)
		{
			int len = sizeof(peeraddr);
			int n=getpeername(clientsd,(struct sockaddr*)&peeraddr,&len);
			char ip[MAXLINESIZE];
			if(n==-1)
			{
				fprintf(stderr,"Peer Call Error!");
				exit(-1);
			}
			const char* res=inet_ntop(AF_INET,&peeraddr.sin_addr,ip,MAXLINESIZE);
			fprintf(stdout,"IP:%s & Port: %d\n",ip,peeraddr.sin_port);
			//close listening socket in child, so that reference count remains one. child serves the client, it doesn't need listening sockt to do this
			close(listensd);

			//server client
			processClient(clientsd);

			//close connected socket
			close(clientsd);
			exit(0);
		}
		close(clientsd);
	}
	return 0;
```

```
}

void processClient(int clientsd)
{
	while((noBytesRead=read(clientsd,buffer,sizeof(buffer)))>0){
		fprintf(stdout,"%s\n",buffer);
		write(clientsd,buffer,noBytesRead);
	}
}
```

==========

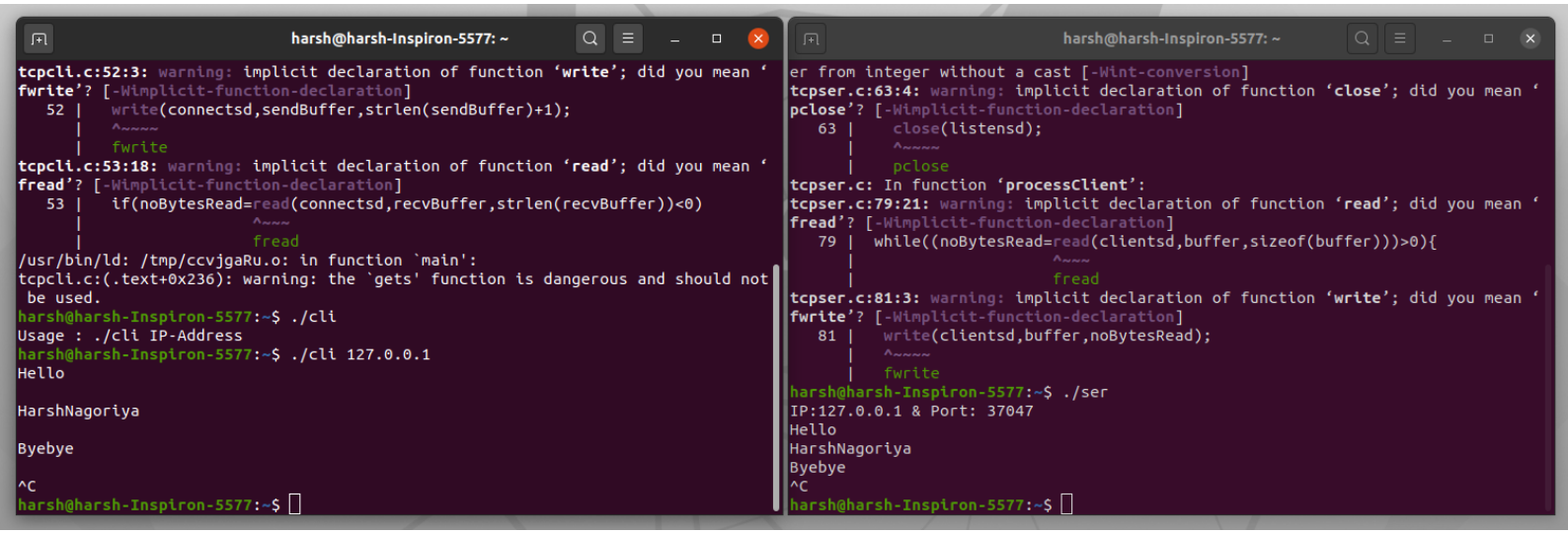
**ECHOTCPCLIENT.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int main(int argc, char** argv)
{
	int connectsd;

	char sendBuffer[MAXLINESIZE+1];
	char recvBuffer[MAXLINESIZE+1];

	struct sockaddr_in servaddr;
	int noBytesRead=0;

	if(argc!=2)
	{
		fprintf(stderr,"Usage : %s IP-Address\n",argv[0]);
		exit(-1);
	}

	if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0)
	{
		fprintf(stderr,"Cannot create socket\n");
		exit(-1);
	}

	bzero(&servaddr,sizeof(servaddr));
```

```
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);

        if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0)
        {
                fprintf(stderr,"Error in inet_pton");
                exit(-1);
        }

        if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
        {
                fprintf(stderr,"Error in connect");
                exit(-1);
        }

        for(;gets(sendBuffer)!=NULL;)
        {
                write(connectsd,sendBuffer,strlen(sendBuffer)+1);
                if(noBytesRead=read(connectsd,recvBuffer,strlen(recvBuffer))<0)
                        exit(0);
                fprintf(stdout,"%s\n",recvBuffer);
        }
        return 0;
}
```

==========



**b) UDP Socket Programming (UNIX)**
**Aim:** Implement concurrent day-time client-server application.
**Tools/ Apparatus:** Unix/Linux C Programming Environment
**Procedure:**
1. Write a server(UDP) C Program that waits in recvfrom
2. Write a client(UDP) C Program that calls sendto to send string to server program knowing IP address and port number.
3. Server replies current date and time (using time, and ctime calls) to client.

Program:

==========

**DAYTIMESERVER.C**

==========

```
 #include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

int main()
{

        struct sockaddr_in sa;
        int sockfd, coontfd;
        char str[1025];
        time_t tick;
        sockfd = socket(AF_INET, SOCK_STREAM, 0);
        if (sockfd < 0)
        {
                printf("Error in creating socket\n");
                exit(0);
        }
        else
        {
                printf("Socket Created\n");
        }
        printf("Socket created\n");
        bzero(&sa, sizeof(sa));

                        memset(str, '0', sizeof(str));
        sa.sin_family = AF_INET;
        sa.sin_port = htons(5600);
```

```
        sa.sin_addr.s_addr = htonl(INADDR_ANY);
        if (bind(sockfd, (struct sockaddr*)&sa, sizeof(sa))<0)
        {
                printf("Bind Error\n");
        }
        else
                printf("Binded\n");
        listen(sockfd, 10);
        while(1)
        {
                coontfd = accept(sockfd, (struct sockaddr*)NULL ,NULL); // Accept a request from
client
                printf("Accepted\n");

      tick = time(NULL);

      snprintf(str, sizeof(str), "%.24s\r\n", ctime(&tick)); // read sys time and write to buffer

      printf("sent\n");
      printf("%s\n", str);
                write(coontfd, str, strlen(str)); // send buffer to client
        }
        close(sockfd); // close the socket
        return 0;
}
```

==========

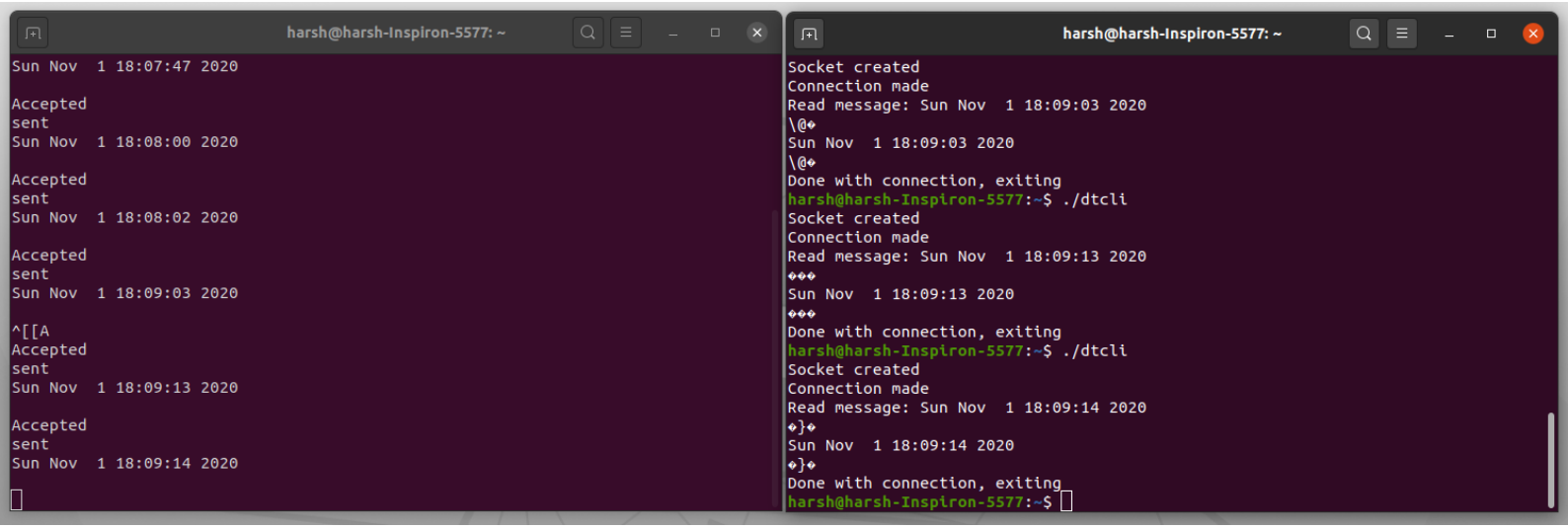
**DAYTIMECLIENT.C**

==========

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>
int main()
{
        struct sockaddr_in sa;
        int n, sockfd;
        char buff[1025];
        sockfd = socket(PF_INET, SOCK_STREAM, 0);
        if (sockfd < 0) {
                printf("Error in creation\n");
                exit(0);
        }
```

```
        else
                printf("Socket created\n");

        bzero(&sa, sizeof(sa));
        sa.sin_family = AF_INET;
        sa.sin_port = htons(5600);
        if (connect(sockfd, (struct sockaddr_in*)&sa, sizeof(sa)) < 0)
        {
                printf("Connection failed\n");
                exit(0);
        }
        else
                printf("Connection made\n");

        if ( n = read(sockfd, buff, sizeof(buff))) {
        printf("Read message: %s\n", buff);
                printf("%s\n", buff);
      printf("Done with connection, exiting\n");

                exit(0); }
        else
        {
      printf("Read Error\n");
        }
        close(sockfd);
        return 0;
}
```

==========



# EXPERIMENT-2

**Configuring Socket options**
**Aim:** Configure following options on server socket and tests them: SO_KEEPALIVE, SO_LINGER, SO_SNDBUF, SO_RCVBUF, TCP_NODELAY
**Tools/ Apparatus:** Unix/Linux C Programming Environment
**Procedure:**
1) Write a server(TCP) C Program that sets the socket options using setsockopt on server one by one and displays the information using getsockopt.

Program:

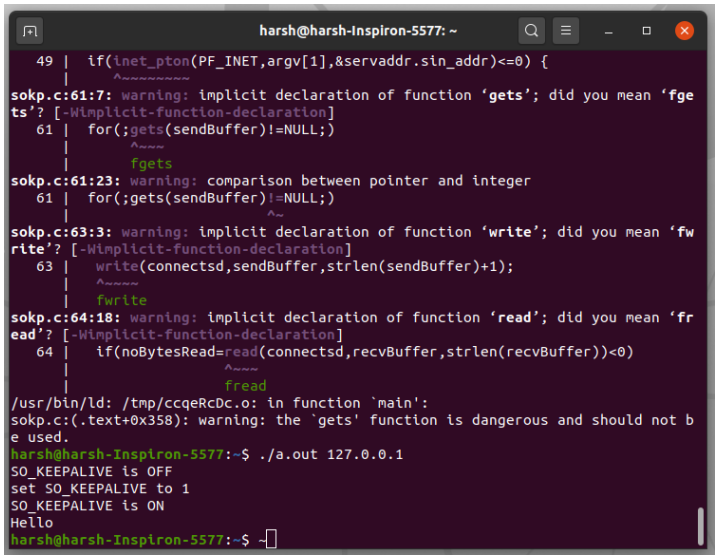
==========

**SOKEEPALIVE.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5877

int main(int argc, char** argv)
{
        int connectsd;
        char sendBuffer[MAXLINESIZE+1];
        char recvBuffer[MAXLINESIZE+1];

        struct sockaddr_in servaddr;
        int noBytesRead=0;

        if(argc!=2)
        {
                fprintf(stderr,"Usage : %s IP-Address\n",argv[0]);
                exit(-1);
        }

        if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0)
        {
                fprintf(stderr,"Cannot create socket\n");
                exit(-1);
        }

  int res = 0, sendbuff, optlen;
```

```
 optlen = sizeof(sendbuff);
 res = getsockopt(connectsd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff, &optlen);
  printf("SO_KEEPALIVE is %s\n", (sendbuff ? "ON" : "OFF"));
 sendbuff = 1;
 printf("set SO_KEEPALIVE to %d\n", sendbuff);
 res = setsockopt(connectsd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff, sizeof(sendbuff));

 // Get flag
 optlen = sizeof(sendbuff);
 res = getsockopt(connectsd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff, &optlen);
 printf("SO_KEEPALIVE is %s\n", (sendbuff ? "ON" : "OFF"));
        bzero(&servaddr,sizeof(servaddr));
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);

        if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0) {
                fprintf(stderr,"Error in inet_pton");
                exit(-1);
        }
        if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
        {

                fprintf(stderr,"Error in connect");
                exit(-1);
        }
        for(;gets(sendBuffer)!=NULL;)
        {
                write(connectsd,sendBuffer,strlen(sendBuffer)+1);
                if(noBytesRead=read(connectsd,recvBuffer,strlen(recvBuffer))<0)
                        exit(0);
                fprintf(stdout,"%s\n",recvBuffer);
        }
        return 0;
}
```



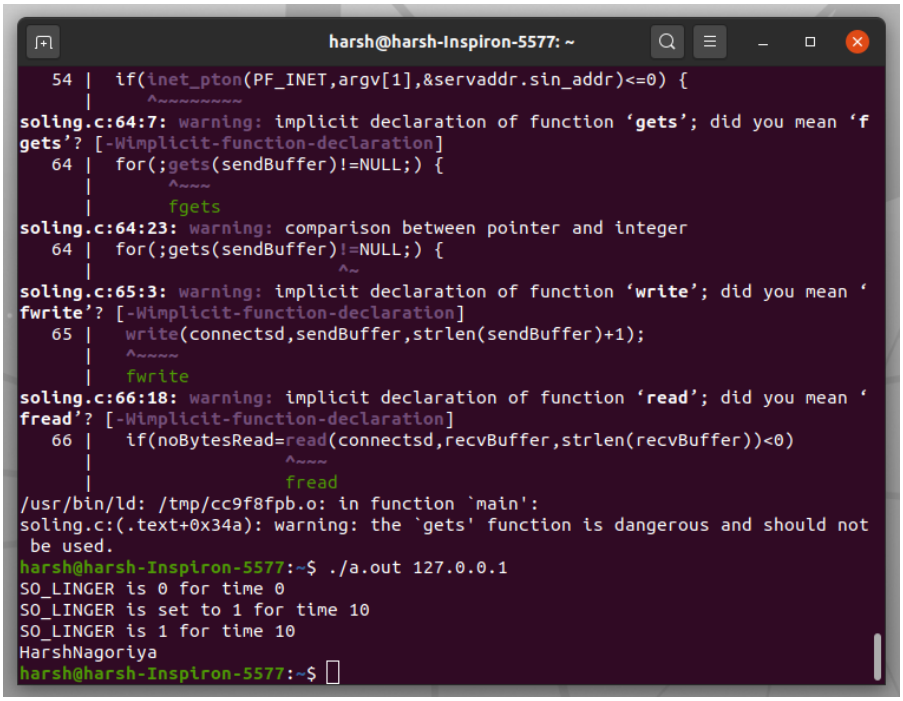
==========

**SOLINGER.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555
int main(int argc, char** argv)
{
        int connectsd;
        char sendBuffer[MAXLINESIZE+1];
        char recvBuffer[MAXLINESIZE+1];
        struct sockaddr_in servaddr;
        int noBytesRead=0;
        if(argc!=2) {
                fprintf(stderr,"Usage : %s IP-Address\n",argv[0]);
                exit(-1);
        }

        if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0) {
                fprintf(stderr,"Cannot create socket\n");
                exit(-1);
        }
 struct linger {
   int l_onoff;
   int l_linger;
 } l;
 socklen_t optlen;
 int res = 0;
 // Get buffer size
 optlen = sizeof(l);
 res = getsockopt(connectsd, SOL_SOCKET, SO_LINGER, &l, &optlen);

 // printf("send buffer size = %d\n", sendbuff);
 printf("SO_LINGER is %d for time %d\n", l.l_onoff, l.l_linger);

 // Set buffer size
 l.l_onoff = 1;
 l.l_linger = 10;
 printf("SO_LINGER is set to %d for time %d\n", l.l_onoff, l.l_linger);
 res = setsockopt(connectsd, SOL_SOCKET, SO_LINGER, &l, sizeof(l));
```

```
  // Get buffer size
  optlen = sizeof(l);
  res = getsockopt(connectsd, SOL_SOCKET, SO_LINGER, &l, &optlen);
  // printf("send buffer size = %d\n", sendbuff);
  printf("SO_LINGER is %d for time %d\n", l.l_onoff, l.l_linger);
        bzero(&servaddr,sizeof(servaddr));
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);
        if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0) {
                fprintf(stderr,"Error in inet_pton");
                exit(-1);
        }
        if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0) {

                fprintf(stderr,"Error in connect");
                exit(-1);
        }
        for(;gets(sendBuffer)!=NULL;) {
                write(connectsd,sendBuffer,strlen(sendBuffer)+1);
                if(noBytesRead=read(connectsd,recvBuffer,strlen(recvBuffer))<0)
                        exit(0);
                fprintf(stdout,"%s\n",recvBuffer);
        }
        return 0;
}
```



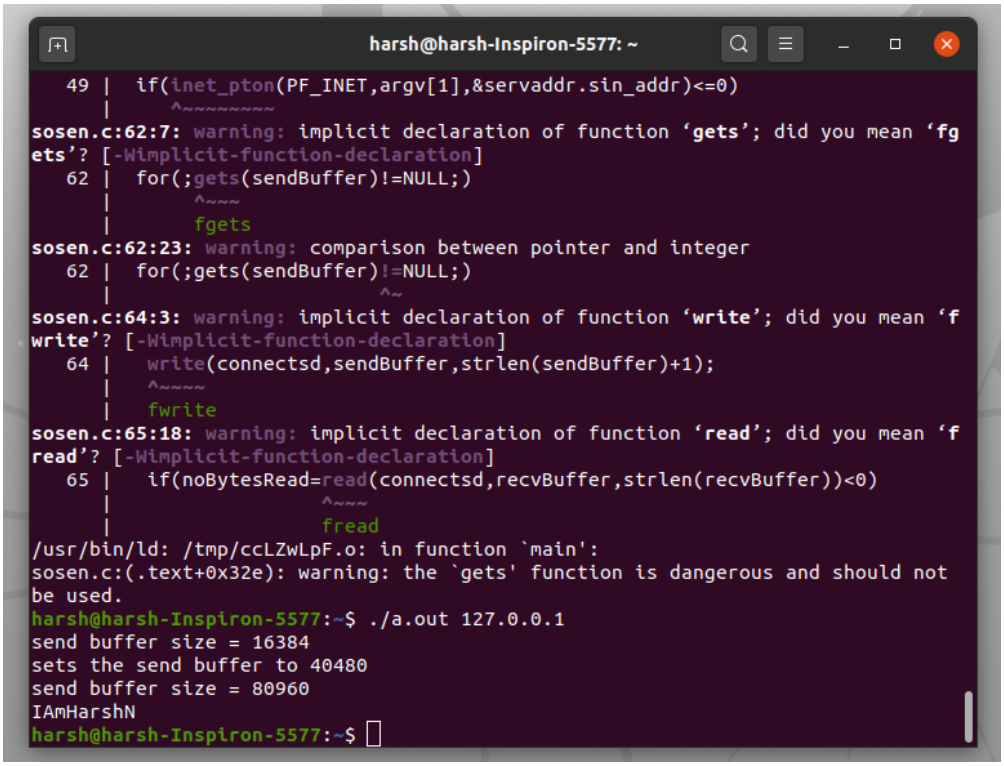
==========

**SOSENDBUF.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int main(int argc, char** argv)
{
        int connectsd;
        char sendBuffer[MAXLINESIZE+1];
        char recvBuffer[MAXLINESIZE+1];
        struct sockaddr_in servaddr;
        int noBytesRead=0;
        if(argc!=2) {
                fprintf(stderr,"Usage : %s IP-Address\n",argv[0]);
                exit(-1);
        }
        if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0) {
                fprintf(stderr,"Cannot create socket\n");
                exit(-1);
        }

  int sendbuff;
  socklen_t optlen;
  int res = 0;
  // Get buffer size
  optlen = sizeof(sendbuff);
  res = getsockopt(connectsd, SOL_SOCKET, SO_SNDBUF, &sendbuff, &optlen);
  printf("send buffer size = %d\n", sendbuff);

  // Set buffer size
  sendbuff = 40480;
  printf("sets the send buffer to %d\n", sendbuff);
  res = setsockopt(connectsd, SOL_SOCKET, SO_SNDBUF, &sendbuff, sizeof(sendbuff));

  // Get buffer size
  optlen = sizeof(sendbuff);
  res = getsockopt(connectsd, SOL_SOCKET, SO_SNDBUF, &sendbuff, &optlen);
  printf("send buffer size = %d\n", sendbuff);
        bzero(&servaddr,sizeof(servaddr));
```

```
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);

        if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0)
        {
                fprintf(stderr,"Error in inet_pton");
                exit(-1);
        }

        if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
        {
                fprintf(stderr,"Error in connect");
                exit(-1);
        }

        for(;gets(sendBuffer)!=NULL;)
        {
                write(connectsd,sendBuffer,strlen(sendBuffer)+1);
                if(noBytesRead=read(connectsd,recvBuffer,strlen(recvBuffer))<0)
                        exit(0);
                fprintf(stdout,"%s\n",recvBuffer);
        }
        return 0;
}
```



==========

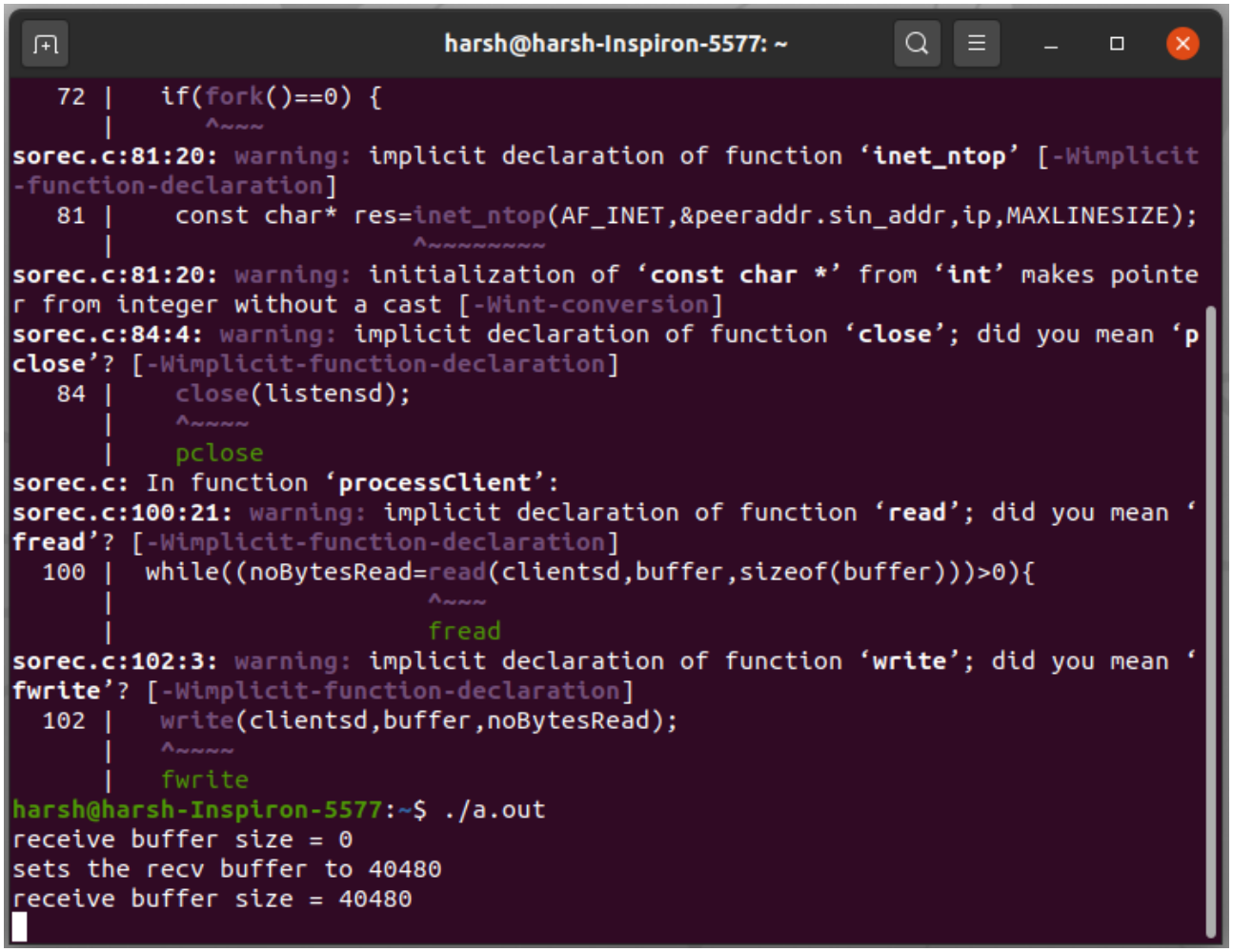
**SORECBUF.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int listensd,clientsd;
char buffer[MAXLINESIZE+1];

struct sockaddr_in servaddr;
struct sockaddr_in peeraddr;
int noBytesRead=0;
//
void processClient(int);

int main()
{
        //create socket
        if((listensd=socket(AF_INET,SOCK_STREAM,0))<0)
        {
                fprintf(stderr,"Cannont create socket\n");
                exit(-1);
        }

  int sockfd, recvbuff;
  socklen_t optlen;

  int res = 0;

  // Get buffer size
  optlen = sizeof(recvbuff);
  res = getsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, &optlen);

  printf("receive buffer size = %d\n", recvbuff);

  // Set buffer size
  recvbuff = 40480;

  printf("sets the recv buffer to %d\n", recvbuff);
  res = setsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, sizeof(recvbuff));
```

```
  // Get buffer size
  optlen = sizeof(recvbuff);
  res = getsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, &optlen);

  printf("receive buffer size = %d\n", recvbuff);

        //Initialize socket address structure
        bzero(&servaddr,sizeof(servaddr));
        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);

        servaddr.sin_addr.s_addr=htonl(INADDR_ANY);   //INADDR_ANY is wildcard returns
local address when not connected

        //bind socket
        if(bind(listensd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0) {
                fprintf(stderr,"Error in bind\n");
                exit(-1);
        }

        //make socket listening socket
        if(listen(listensd,5)<0) {
                fprintf(stderr,"Error in listen\n");
                exit(-1);
        }
        //wait for client connection
        for(;;) {
                clientsd=accept(listensd,(struct sockaddr*)NULL,NULL);
                if(fork()==0) {
                        int len = sizeof(peeraddr);
                        int n=getpeername(clientsd,(struct sockaddr*)&peeraddr,&len);
                        char ip[MAXLINESIZE];
                        if(n==-1) {
                                fprintf(stderr,"Peer Call Error!");
                                exit(-1);
                        }

                        const char*
res=inet_ntop(AF_INET,&peeraddr.sin_addr,ip,MAXLINESIZE);
                        fprintf(stdout,"IP:%s & Port: %d\n",ip,peeraddr.sin_port);
                        //close listening socket in child, so that reference count remains one. child
serves the client, it doesn't need listening sockt to do this
                        close(listensd);

                        //server client
                        processClient(clientsd);

                        //close connected socket
```

```
                        close(clientsd);
                        exit(0);
                }
                close(clientsd);
        }
        return 0;
}

void processClient(int clientsd)
{
        while((noBytesRead=read(clientsd,buffer,sizeof(buffer)))>0){
                fprintf(stdout,"%s\n",buffer);
                write(clientsd,buffer,noBytesRead);
        }
}
```



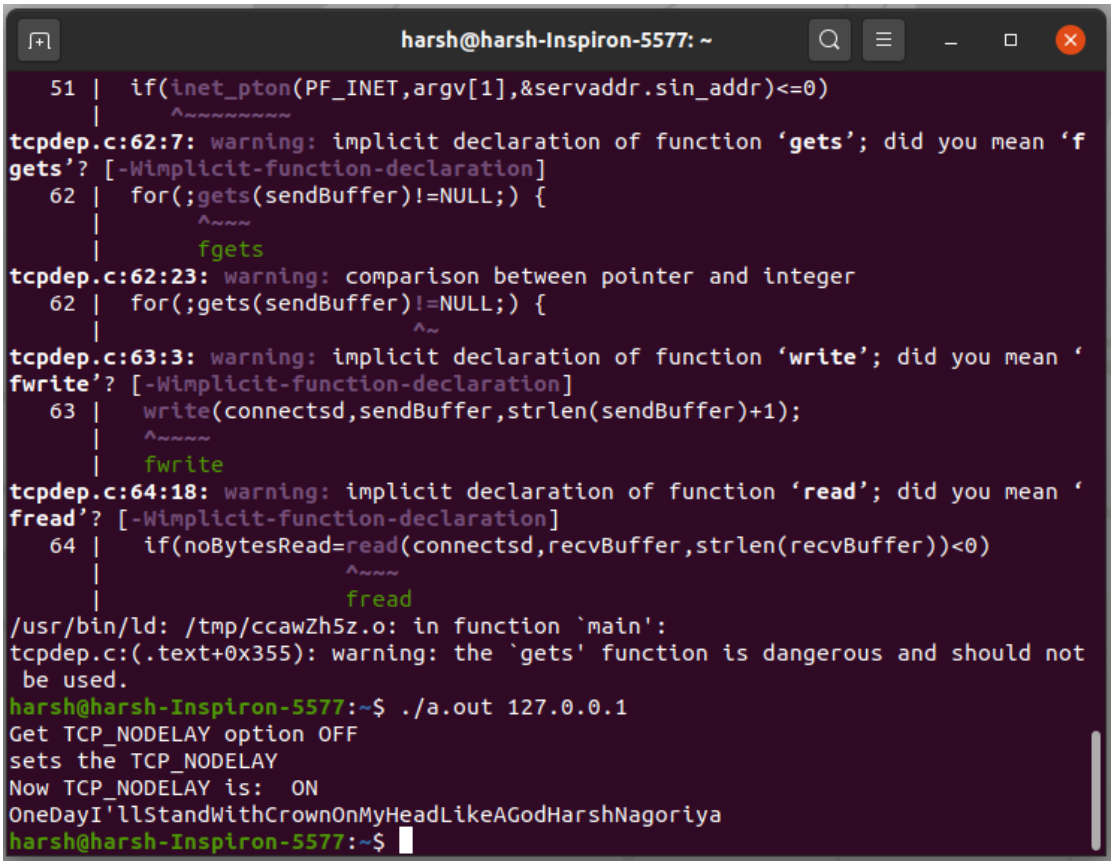
==========

**TCPNODELAY.C**

==========

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>
#include <netinet/tcp.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int main(int argc, char** argv)
{
        int connectsd;
        char sendBuffer[MAXLINESIZE+1];
        char recvBuffer[MAXLINESIZE+1];
        struct sockaddr_in servaddr;
        int noBytesRead=0;

        if(argc!=2) {
                fprintf(stderr,"Usage : %s IP-Address\n",argv[0]);
                exit(-1);
        }

        if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0) {
                fprintf(stderr,"Cannot create socket\n");
                exit(-1);
        }

  int flag;
  socklen_t optlen;
  int res = 0;
  optlen = sizeof(flag);
  res = getsockopt(connectsd, IPPROTO_TCP, TCP_NODELAY, &flag, &optlen);
  printf("Get TCP_NODELAY option %s\n", (flag ? "ON" : "OFF"));

  flag = 1;
  printf("sets the TCP_NODELAY\n");
  res = setsockopt(connectsd, IPPROTO_TCP, TCP_NODELAY, &flag, sizeof(int));
  // test now
  flag = 0;
  optlen = sizeof(flag);
  res = getsockopt(connectsd, IPPROTO_TCP, TCP_NODELAY, &flag, &optlen);
  printf("Now TCP_NODELAY is:  %s\n", (flag ? "ON" : "OFF"));
```

```
        bzero(&servaddr,sizeof(servaddr));

        servaddr.sin_family=AF_INET;
        servaddr.sin_port=htons(SERV_PORT);

        if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0)
        {
                fprintf(stderr,"Error in inet_pton");
                exit(-1);
        }

        if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0) {
                scanf("%s", x);
                fprintf(stderr,"Error in connect");
                exit(-1);
        }
        for(;gets(sendBuffer)!=NULL;) {
                write(connectsd,sendBuffer,strlen(sendBuffer)+1);
                if(noBytesRead=read(connectsd,recvBuffer,strlen(recvBuffer))<0)
                        exit(0);
                fprintf(stdout,"%s\n",recvBuffer);
        }
        return 0;
}
```



# EXPERIMENT-3

**Aim:** Data Representation and Data Validation: XML Schema and XML instance document, JSON.
**Tools/ Apparatus:**GUI-IDE Tool NetBeans 6.0
**Procedure:**
1. Design a schema for student list. A student has information such as name, semester, roll no, email-ids, phone-nos, etc.
2. Write an XML instance document for the designed schema and validate this instance document against the schema.

==========

**student.xsd**

==========

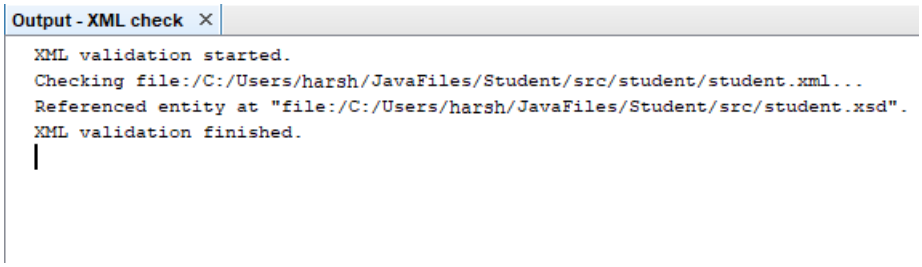
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema version="1.0"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified">
   <xs:element name="Student">
      <xs:complexType>
         <xs:sequence>

            <xs:element name="branch">
               <xs:complexType>
                  <xs:sequence>
                  <xs:element name="name" type="xs:string"/>
                  <xs:element name="age" type="xs:integer"/>
                  <xs:element name="roll" type="xs:string"/>
                  <xs:element name="email" type="xs:string"/>
                  <xs:element name="phone" type="xs:integer"/>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="sem" type="xs:integer"/>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

==========

**student.xml**

==========

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->

<?xml version="1.0" encoding="UTF-8"?>
<Student
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xsi:noNamespaceSchemaLocation='../student.xsd'>
    <branch name="IT" sem="7">
        <name>Harsh</name>
        <age>21</age>
        <roll>IT-053</roll>
        <email>contact@harshnagoriya.ml</email>
        <phone>9033350197</phone>
    </branch>
</Student>
```



# EXPERIMENT-4

**Aim:** WSDL based webservice and its monitoring: Implement ArithmeticService that implements add and subtract operations / Java based: Implement TrigonometricService that implements sin, and cos operations. Monitor SOAP request and response packets. Analyze parts of it and compare them with the operations (java functions) headers.
**Tools/ Apparatus:** Web service, BPEL Runtime Environment: GlassFish Server, GUI-IDE
**Tool:** NetBeans 6.0, WSMonitor

==========

**Trigo.java**

==========

```
package src;

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

@WebService(serviceName = "Trigo")
public class Trigo {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }
    /**
     * Web service operation
     */
    @WebMethod(operationName = "cos")
    public double cos(@WebParam(name = "degree") double degree) {
        //TODO write your implementation code here:
        double radian = Math.toRadians(degree);
        double sin = Math.cos(radian);
        return radian;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "sin")
    public double sin(@WebParam(name = "degree") float degree) {
        //TODO write your implementation code here:
   double radian = Math.toRadians(degree);
        double sin = Math.sin(radian);
        return radian;
```

```
    }
}
```

==========

**Trigoclient.java**

==========

```
package trigoclient;

public class TrigoClient {
    public static void main(String[] args) {
        // TODO code application logic here
        double sinAns = sin(60);
        double cosAns = cos(45);
        System.out.println("sin(60)= " + sinAns);
        System.out.println("cos(45)= " + cosAns);

    }
    private static double sin(float degree) {
        src.Trigo_Service service = new src.Trigo_Service();
        src.Trigo port = service.getTrigoPort();
        return port.sin(degree);
    }
    private static double cos(double degree) {
        src.Trigo_Service service = new src.Trigo_Service();
        src.Trigo port = service.getTrigoPort();
        return port.cos(degree);
    }
}
```

**Method parameter(s)**

| Type | Value |
|---|---|
| float | 45 |

**Method returned**

double : "**0.7853981633974483**"

**SOAP Request**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:sin xmlns:ns2="http://src/">
            <degree>45.0</degree>
        </ns2:sin>
    </S:Body>
</S:Envelope>
```

**SOAP Response**

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:sinResponse xmlns:ns2="http://src/">
            <return>0.7853981633974483</return>
        </ns2:sinResponse>
    </S:Body>
</S:Envelope>
```

**cos Method invocation**

---

**Method parameter(s)**

| Type | Value |
|---|---|
| double | 30 |

---

**Method returned**

double : "**0.5235987755982988**"

---

**SOAP Request**

---

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:cos xmlns:ns2="http://src/">
            <degree>30.0</degree>
        </ns2:cos>
    </S:Body>
</S:Envelope>
```

---

**SOAP Response**

---

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:cosResponse xmlns:ns2="http://src/">
            <return>0.5235987755982988</return>
        </ns2:cosResponse>
    </S:Body>
</S:Envelope>
```

==========

# WSDL File

==========

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11 JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11 JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. -->
- <definitions name="Trigo" targetNamespace="http://src/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://src/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  - <types>
    - <xsd:schema>
        <xsd:import schemaLocation="http://localhost:8080/TrigonometricService/Trigo?xsd=1" namespace="http://src/"/>
      </xsd:schema>
    </types>
  - <message name="sin">
      <part name="parameters" element="tns:sin"/>
    </message>
  - <message name="sinResponse">
      <part name="parameters" element="tns:sinResponse"/>
    </message>
  - <message name="cos">
      <part name="parameters" element="tns:cos"/>
    </message>
  - <message name="cosResponse">
      <part name="parameters" element="tns:cosResponse"/>
    </message>
  - <message name="hello">
      <part name="parameters" element="tns:hello"/>
    </message>
  - <message name="helloResponse">
      <part name="parameters" element="tns:helloResponse"/>
    </message>
  - <portType name="Trigo">
    - <operation name="sin">
        <input message="tns:sin" wsam:Action="http://src/Trigo/sinRequest"/>
        <output message="tns:sinResponse" wsam:Action="http://src/Trigo/sinResponse"/>
      </operation>
    - <operation name="cos">
        <input message="tns:cos" wsam:Action="http://src/Trigo/cosRequest"/>
        <output message="tns:cosResponse" wsam:Action="http://src/Trigo/cosResponse"/>
      </operation>
    - <operation name="hello">
        <input message="tns:hello" wsam:Action="http://src/Trigo/helloRequest"/>
        <output message="tns:helloResponse" wsam:Action="http://src/Trigo/helloResponse"/>
      </operation>
```

```
            <input message="tns:cos" wsam:Action="http://src/Trigo/cosRequest"/>
            <output message="tns:cosResponse" wsam:Action="http://src/Trigo/cosResponse"/>
        </operation>
      - <operation name="hello">
            <input message="tns:hello" wsam:Action="http://src/Trigo/helloRequest"/>
            <output message="tns:helloResponse" wsam:Action="http://src/Trigo/helloResponse"/>
        </operation>
    </portType>
  - <binding name="TrigoPortBinding" type="tns:Trigo">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      - <operation name="sin">
            <soap:operation soapAction=""/>
          - <input>
                <soap:body use="literal"/>
            </input>
          - <output>
                <soap:body use="literal"/>
            </output>
        </operation>
      - <operation name="cos">
            <soap:operation soapAction=""/>
          - <input>
                <soap:body use="literal"/>
            </input>
          - <output>
                <soap:body use="literal"/>
            </output>
        </operation>
      - <operation name="hello">
            <soap:operation soapAction=""/>
          - <input>
                <soap:body use="literal"/>
            </input>
          - <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
  - <service name="Trigo">
      - <port name="TrigoPort" binding="tns:TrigoPortBinding">
            <soap:address location="http://localhost:8080/TrigonometricService/Trigo"/>
        </port>
    </service>
</definitions>
```

Java DB Database Process × | GlassFish Server × | Retriever Output × | TrigoClient (run) ×

```
deps-jar:
Updating property file: C:\Users\harsh\JavaFiles\TrigoClient\build\built-jar.properties
wsimport-init:
wsimport-client-Trigo:
files are up to date
wsimport-client-generate:
Created dir: C:\Users\harsh\JavaFiles\TrigoClient\build\classes
Created dir: C:\Users\harsh\JavaFiles\TrigoClient\build\empty
Created dir: C:\Users\harsh\JavaFiles\TrigoClient\build\generated-sources\ap-source-output
Compiling 11 source files to C:\Users\harsh\JavaFiles\TrigoClient\build\classes
Copying 3 files to C:\Users\harsh\JavaFiles\TrigoClient\build\classes
compile:
run:
sin(60)= 1.0471975511965976
cos(45)= 0.7853981633974483
BUILD SUCCESSFUL (total time: 1 second)
```

## EXPERIMENT-5

**Aim:** Design and test BPEL module that composes ArithmeticService and TrignometricService.
**Tools/ Apparatus:** Web service, BPEL Runtime Environment: GlassFish Server, GUI-IDE Tool: NetBeans 6.0

==========

**Arithmetic Service WSDL**

==========

<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://src/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://src/" name="ArithmeticServiceService">
<types>
<xsd:schema>
<xsd:import namespace="http://src/" schemaLocation="ArithmeticServiceService_xsd_1.xsd"></xsd:import>
</xsd:schema>
</types>
<message name="addition">
<part name="parameters" element="tns:addition"></part>
</message>
<message name="additionResponse">
<part name="parameters" element="tns:additionResponse"></part>
</message>
<portType name="ArithmeticService">
<operation name="addition">
<input message="tns:addition"></input>
<output message="tns:additionResponse"></output>
</operation>
</portType>
<binding name="ArithmeticServicePortBinding" type="tns:ArithmeticService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"></soap:binding>
<operation name="addition">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>

```
</binding>
<service name="ArithmeticServiceService">
<port name="ArithmeticServicePort" binding="tns:ArithmeticServicePortBinding">
<soap:address
location="http://localhost:8080/MyArithmeticService/ArithmeticServiceService"></soap:address>
</port>
</service>
</definitions>
```

==========

**Arithmetic Service Wrapper WSDL**

==========

```
<definitions
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="ArithmeticServiceServiceWrapper"
targetNamespace="http://enterprise.netbeans.org/bpel/ArithmeticServiceServiceWrapper"
xmlns:tns="http://enterprise.netbeans.org/bpel/ArithmeticServiceServiceWrapper"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:ns="http://src/">
   <import location="ArithmeticServiceService.wsdl" namespace="http://src/"/>
   <plnk:partnerLinkType name="ArithmeticService4">
      <plnk:role name="ArithmeticServiceRole" portType="ns:ArithmeticService"/>
   </plnk:partnerLinkType>
   <plnk:partnerLinkType name="ArithmeticServiceLinkType">
      <plnk:role name="ArithmeticServiceRole" portType="ns:ArithmeticService"/>
   </plnk:partnerLinkType>
</definitions>
```

==========

**Arithmetic Service XSD**

==========

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><xs:schema
xmlns:tns="http://src/" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0"
targetNamespace="http://src/">

<xs:element name="addition" type="tns:addition"></xs:element>

<xs:element name="additionResponse" type="tns:additionResponse"></xs:element>

<xs:complexType name="addition">
<xs:sequence>
<xs:element name="input1" type="xs:double"></xs:element>
<xs:element name="input2" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
```

<xs:complexType name="additionResponse">
<xs:sequence>
<xs:element name="return" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

==========

**Scientific Services WSDL**

==========

<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://src/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://src/" name="DualScientifcServiceService">
<types>
<xsd:schema>
<xsd:import namespace="http://src/" schemaLocation="DualScientifcServiceService_xsd_1.xsd"></xsd:import>
</xsd:schema>
</types>
<message name="sin">
<part name="parameters" element="tns:sin"></part>
</message>
<message name="sinResponse">
<part name="parameters" element="tns:sinResponse"></part>
</message>
<message name="cos">
<part name="parameters" element="tns:cos"></part>
</message>
<message name="cosResponse">
<part name="parameters" element="tns:cosResponse"></part>
</message>
<portType name="DualScientifcService">
<operation name="sin">
<input message="tns:sin"></input>
<output message="tns:sinResponse"></output>
</operation>
<operation name="cos">
<input message="tns:cos"></input>
<output message="tns:cosResponse"></output>
</operation>
</portType>

```
<binding name="DualScientifcServicePortBinding" type="tns:DualScientifcService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"></soap:binding>
<operation name="sin">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="cos">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
</binding>
<service name="DualScientifcServiceService">
<port name="DualScientifcServicePort" binding="tns:DualScientifcServicePortBinding">
<soap:address
location="http://localhost:8080/MyDualScientificService/DualScientifcServiceService"></soap:add
ress>
</port>
</service>
</definitions>
```

==========

**Scientific Service Wrapper WSDL**

==========

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="DualScientifcServiceServiceWrapper"
targetNamespace="http://enterprise.netbeans.org/bpel/DualScientifcServiceServiceWrapper"
xmlns:tns="http://enterprise.netbeans.org/bpel/DualScientifcServiceServiceWrapper"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:ns="http://src/">
   <import location="DualScientifcServiceService.wsdl" namespace="http://src/"/>
   <plnk:partnerLinkType name="DualScientifcService1">
```

```
    <plnk:role name="DualScientifcServiceRole" portType="ns:DualScientifcService"/>
  </plnk:partnerLinkType>
  <plnk:partnerLinkType name="DualScientifcServiceLinkType">
    <plnk:role name="DualScientifcServiceRole" portType="ns:DualScientifcService"/>
  </plnk:partnerLinkType>
</definitions>
```

==========

**Scientific Services XSD**

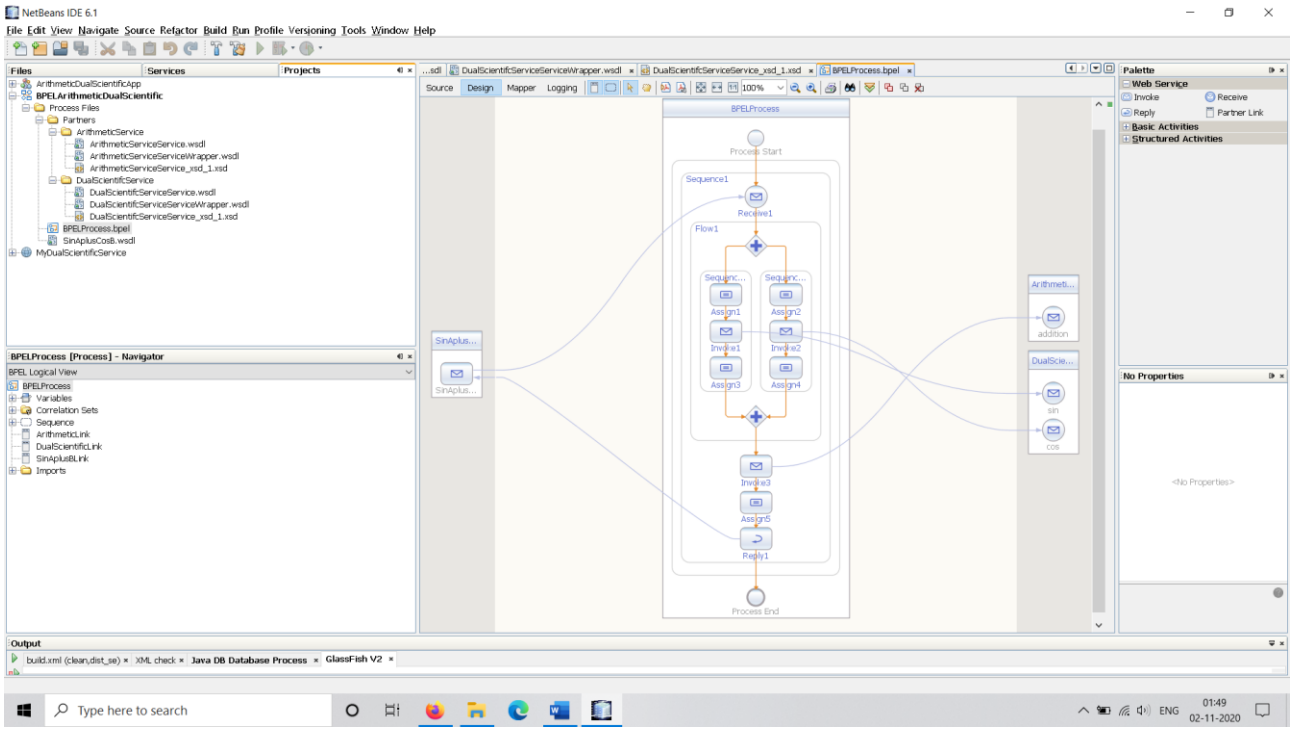
==========

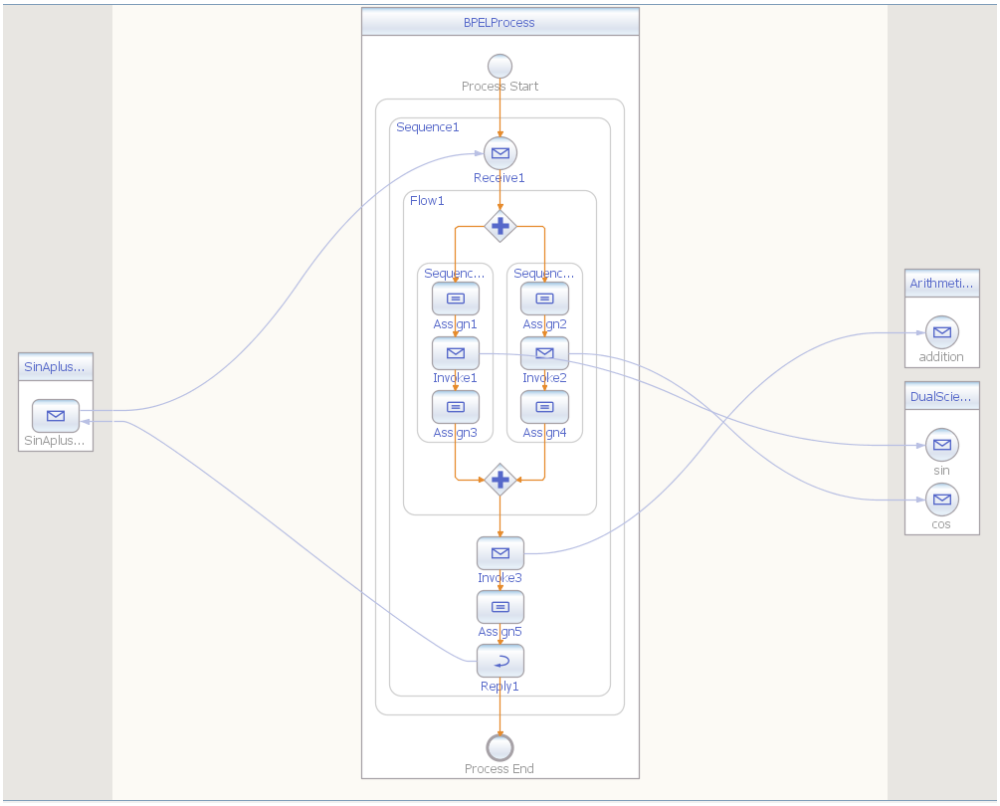
```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><xs:schema
xmlns:tns="http://src/" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0"
targetNamespace="http://src/">

<xs:element name="cos" type="tns:cos"></xs:element>

<xs:element name="cosResponse" type="tns:cosResponse"></xs:element>

<xs:element name="sin" type="tns:sin"></xs:element>

<xs:element name="sinResponse" type="tns:sinResponse"></xs:element>

<xs:complexType name="sin">
<xs:sequence>
<xs:element name="inputSin" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sinResponse">
<xs:sequence>
<xs:element name="return" type="xs:double" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="cos">
<xs:sequence>
<xs:element name="inputCos" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="cosResponse">
<xs:sequence>
<xs:element name="return" type="xs:double" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

==========

**BPEL Process**

==========

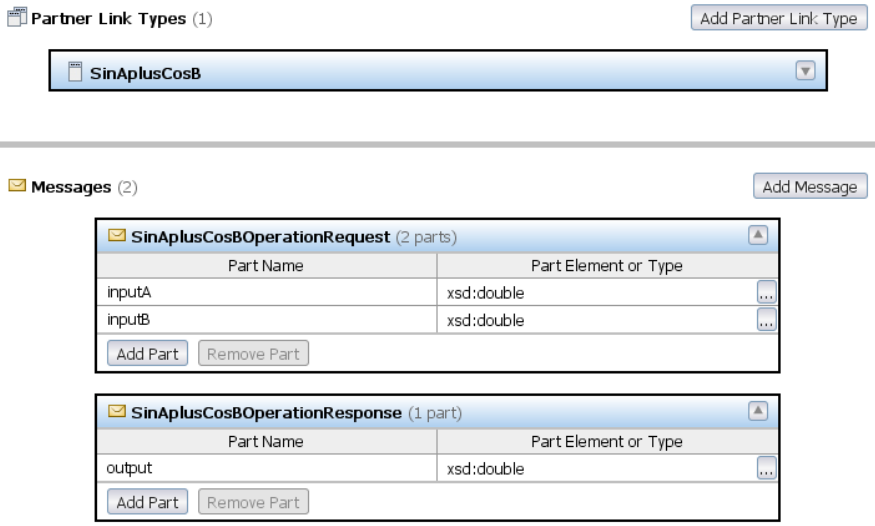


==========

**sinA plus cosB**

==========



# EXPERIMENT-6

**Aim:** Deployment of a HADOOP cluster and monitoring status of its components.
**Tools/ Apparatus:** GUI-IDE Tool NetBeans 6.0, Hadoop Common, Hadoop Distributed File System, Hadoop YARN, Hadoop MapReduce, Ambari

**Procedure:**

- Install the appropriate version of java for your Hadoop.
- ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.
- Download the appropriate hadoop file system from link given below. http://www.apache.org/dyn/closer.cgi/hadoop/common/

- Unpack the downloaded Hadoop distribution. In the distribution, edit the file etc/hadoop/hadoop-env.sh to define some parameters as follows: " #export JAVA_HOME=/usr/java/latest".
- The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
$ mkdir input
$ cp etc/hadoop/*.xml input
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples- 2.9.2.jar
grep input output 'dfs[a-z.]+'
$ cat output/*
```

- Now check that you can ssh to the localhost without a passphrase:

```
$ ssh localhost
```

- If you cannot ssh to localhost without a passphrase, execute the following commands:

```
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

Execution

- The following instructions are to run a MapReduce job locally.

1. Format the filesystem:
   $ bin/hdfs namenode -format
2. Start NameNode daemon and DataNode daemon:
   $ sbin/start-dfs.sh

- The hadoop daemon log output is written to the $HADOOP_LOG_DIR directory (defaults to $HADOOP_HOME/logs).
- Browse the web interface for the NameNode; by default it is available at: NameNode - http://localhost:50070/

- Make the HDFS directories required to execute MapReduce jobs:

$ bin/hdfs dfs -mkdir /user
$ bin/hdfs dfs -mkdir /user/<username>

- Copy the input files into the distributed filesystem:

$ bin/hdfs dfs -put etc/hadoop input

- Run some of the examples provided:

$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar grep input output 'dfs[a-z.]+'

- Examine the output files: Copy the output files from the distributed file-system to the local filesystem and examine them:

$ bin/hdfs dfs -get output output
$ cat output/*

OR

- View the output files on the distributed filesystem:

$ bin/hdfs dfs -cat output/*

- When you’re done, stop the daemons with:

$ sbin/stop-dfs.sh

# EXPERIMENT-7

**Aim:** Perform data intensive computing using map-reduce based programming on a HADOOP cluster.
**Tools/ Apparatus:** GUI-IDE Tool NetBeans 6.0, Hadoop Common, Hadoop Distributed File System, Hadoop YARN, Hadoop MapReduce, Ubuntu

**Procedure:**

Prerequisites

1. Installation and Configuration of Single node Hadoop :

2. Prepare your computer network (Decide no of nodes to set up cluster) :

3. Basic installation and configuration :
   3.1 configure etc/hosts for master and slaves nodes

```
$ sudo gedit /etc/hosts
# Add following hostname and their ip in host table
192.168.2.14 HadoopMaster
192.168.2.15 HadoopSlave1
192.168.2.16 HadoopSlave2
```

   3.2 Create hadoop as group and hduser as user in all Machines (if not created !!).

```
DDU@HadoopMaster:~$ sudo addgroup hadoop
DDU@HadoopMaster:~$ sudo adduser --ingroup hadoop hduser
sudo usermod -a -G sudo hduser
```

OR

Add following line in /etc/sudoers/
hduser ALL=(ALL:ALL) ALL

3.3 Install rsync for sharing hadoop source with rest all Machines, and reboot all the machine.

```
$ sudo apt-get install rsync
```

3.4 To make above changes reflected, we need to reboot all of the Machines.

```
$ sudo reboot
```

4. Applying Common Hadoop Configuration
   4.1 Update core-site.xml
   Update this file by changing hostname from localhost to HadoopMaster
   ## To edit file, fire the below given command
   hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.xml
   ## Paste these lines into <configuration> tag OR Just update it by replacing localhost with
   master
   <property>

<name>fs.default.name</name>
<value>hdfs://HadoopMaster:9000</value>
</property>

4.2 Update hdfs-site.xml

Update this file by updating repliction factor from 1 to 3.
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml
## Paste/Update these lines into <configuration> tag
<property>
<name>dfs.replication</name>
<value>3</value>
</property>

4.3 Update yarn-site.xml

Update this file by updating the following three properties by updating hostname from
localhost to HadoopMaster
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit yarn-site.xml
## Paste/Update these lines into <configuration> tag
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>HadoopMaster:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>HadoopMaster:8035</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>HadoopMaster:8050</value>
</property>

4.4 Update Mapred-site.xml

Update this file by updating and adding following properties,
## To edit file, fire the below given command
hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit mapred-site.xml
## Paste/Update these lines into <configuration> tag

<property>
<name>mapreduce.job.tracker</name>
<value>HadoopMaster:5431</value>
</property>
<property>
<name>mapred.framework.name</name>
<value>yarn</value>
</property>

4.5 Update masters

Update the directory of master nodes of Hadoop cluster

## To edit file, fire the below given command

hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit masters

## Add name of master nodes

HadoopMaster

4.6 Update slaves

Update the directory of slave nodes of Hadoop cluster

## To edit file, fire the below given command

hduser@HadoopMaster:/usr/local/hadoop/etc/hadoop$ sudo gedit slaves

## Add name of slave nodes

HadoopSlave1

HadoopSlave2

5. Copying/Sharing/Distributing Hadoop config files to rest all nodes – master/slaves

5.1 Use rsync for distributing configured Hadoop source among rest of nodes via network.

# In HadoopSlave1 machine

$ sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave1:/usr/local/hadoop/

# In HadoopSlave2 machine

$ sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave2:/usr/local/hadoop/

6 Applying Master node specific Hadoop configuration: (Only for master nodes)

6.1 Remove existing Hadoop_data folder (which was created while single node hadoop setup.)

$ sudo rm -rf /usr/local/hadoop_tmp/

6.2 : Make same (/usr/local/hadoop_tmp/hdfs) directory and create NameNode ( /usr/local/hadoop_tmp/hdfs/namenode) directory

$ sudo mkdir -p /usr/local/hadoop_tmp/

$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode

6.3 : Make hduser as owner of that directory.

$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/

7 Applying Slave node specific Hadoop configuration : (Only for slave nodes)

7.1 Remove existing Hadoop_data folder (which was created while single node hadoop setup)

$ sudo rm -rf /usr/local/hadoop_tmp/hdfs/

7.2 Creates same (/usr/local/hadoop_tmp/) directory/folder, an inside this folder again Create DataNode (/usr/local/hadoop_tmp/hdfs/namenode) directory/folder

$ sudo mkdir -p /usr/local/hadoop_tmp/

$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode

Step 7C : Make hduser as owner of that directory

sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/

8 Copying ssh key for Setting up passwordless ssh access from Master to Slave node :

hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub

hduser@HadoopSlave1

hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@HadoopSlave2

9. Format Namenonde (Run on MasterNode) :

# Run this command from Masternode
hduser@HadoopMaster: usr/local/hadoop/$ hdfs namenode -format

10. Starting up Hadoop cluster daemons : (Run on MasterNode)

Start HDFS daemons:
hduser@HadoopMaster:/usr/local/hadoop$ start-dfs.sh

11. Start MapReduce daemons:

hduser@HadoopMaster:/usr/local/hadoop$ start-yarn.sh

12. Instead both of these above command you can also use start-all.sh, but its now deprecated so its not recommended to be used for better Hadoop operations.

13. Track/Monitor/Verify Hadoop cluster : (Run on any Node)

Verify Hadoop daemons on Master and slaves(All slave):
hduser@HadoopMaster: jps

## EXPERIMENT-8

**Aim:** Create Restful Webservice and test it using Postman.
**Tools/ Apparatus:** Web service, GlassFish Server, GUI-IDE Tool:NetBeans 6.0

==========

**UserService.java**

==========

```
import java.util.Iterator;
import java.util.List;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.*;

@Path("/UserService")
public class UserService {

    UserContext userDao = new UserContext();
    private static final String SUCCESS_RESULT = "<result>success</result>";
    private static final String FAILURE_RESULT = "<result>failure</result>";

    @GET
    @Path("/users")
    @Produces(MediaType.APPLICATION_JSON) // Because we want to return list of users in
JSON format
    public List<User> getUsers() {
        return userDao.getAllUsers();
    }

    @GET
    @Path("/users/{userid}")
    @Produces(MediaType.APPLICATION_JSON)
    public User getUser(@PathParam("userid") int userid) {
        System.out.println("The ID received in GET is" + userid);
        return userDao.getUser(userid);
    }

    @POST
    @Path("/insertuser")
    @Produces(MediaType.TEXT_PLAIN)
    public String InsertUsers() {
        List<User> NewList = userDao.getAllUsers();
        User newuser = new User(3, "ABC", "XYZ");
        NewList.add(newuser);
```

```
        userDao.saveUserList(NewList);

        return "Inserted";
    }

    @POST
    @Path("/adduser")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public User AddUser(User user) {
        System.out.println("Inside Add User Method");
        List<User> NewList = userDao.getAllUsers();
        NewList.add(user);
        userDao.saveUserList(NewList);
        return user;
    }

    @PUT
    @Path("/updateuser")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public User UpdateUser(User user) {
        System.out.println("The ID received in GET is" + user.getId());
        int result = userDao.updateUser(user);
        if (result == 1) {
            System.out.println("Success in Update");
        } else {
            System.out.println("Failure in Update");
        }

        return user;
    }

    @DELETE
    @Path("/deleteuser/{userid}")
    @Produces(MediaType.TEXT_PLAIN)
    @Consumes(MediaType.APPLICATION_JSON)
    public String deleteUser(@PathParam("userid") int userid) {
        System.out.println("The ID received in DELETE is" + userid);
        int result = userDao.deleteUser(userid);
        System.out.println("Value of Result is" + result);
        if (result == 1) {
            return "SUCCESS";
        }
        return "FAILURE";
    }
}
```

==========

**User.java**

==========

```
package Rest;
import java.io.Serializable;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

//Add Java Class: User.java
@XmlRootElement(name = "user")
public class User implements Serializable {

   private static final long serialVersionUID = 1L;
   private int id;
   private String name;
   private String profession;

   public User() {
   }

   public User(int id, String name, String profession) {
      this.id = id;
      this.name = name;
      this.profession = profession;
   }
   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getProfession() {
      return profession;
   }
   public void setProfession(String profession) {
      this.profession = profession;
   }
```

```
}
```

==========

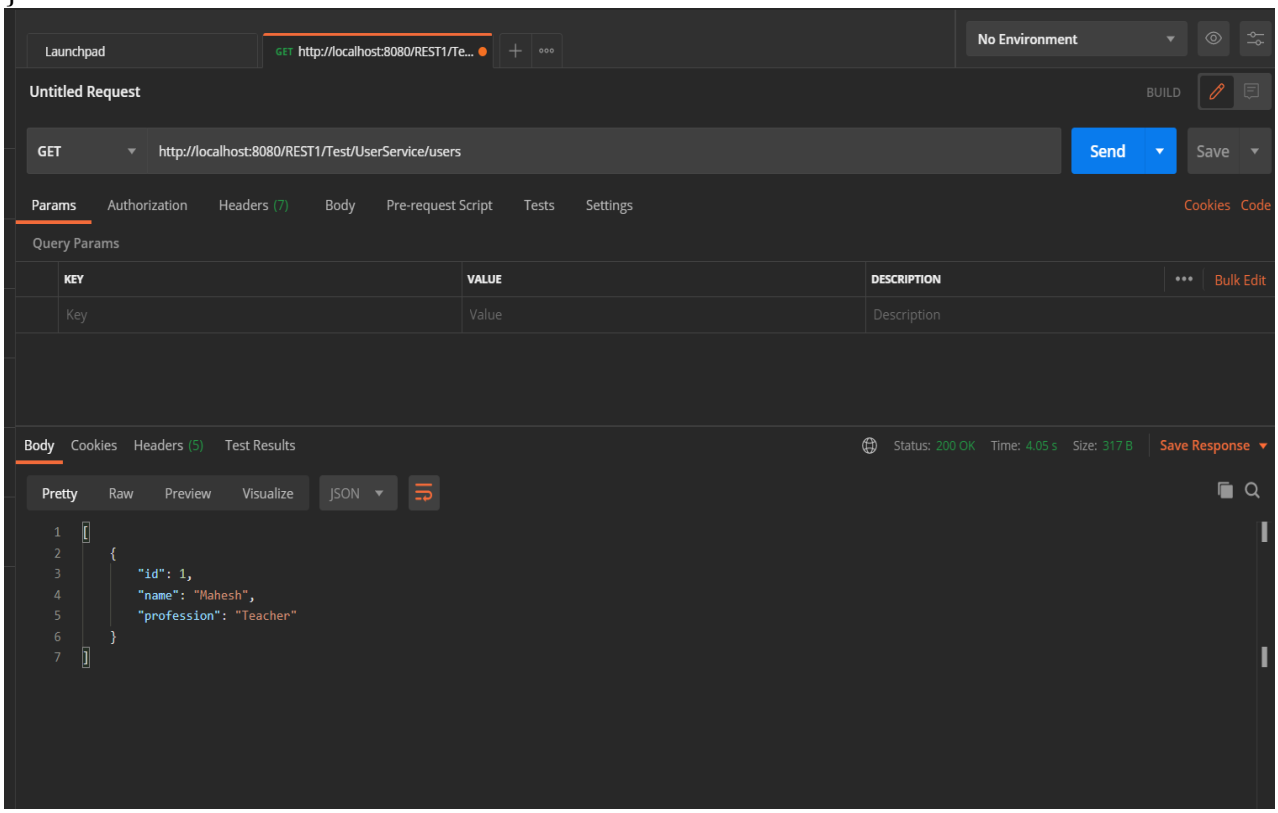
**UserContext.java**

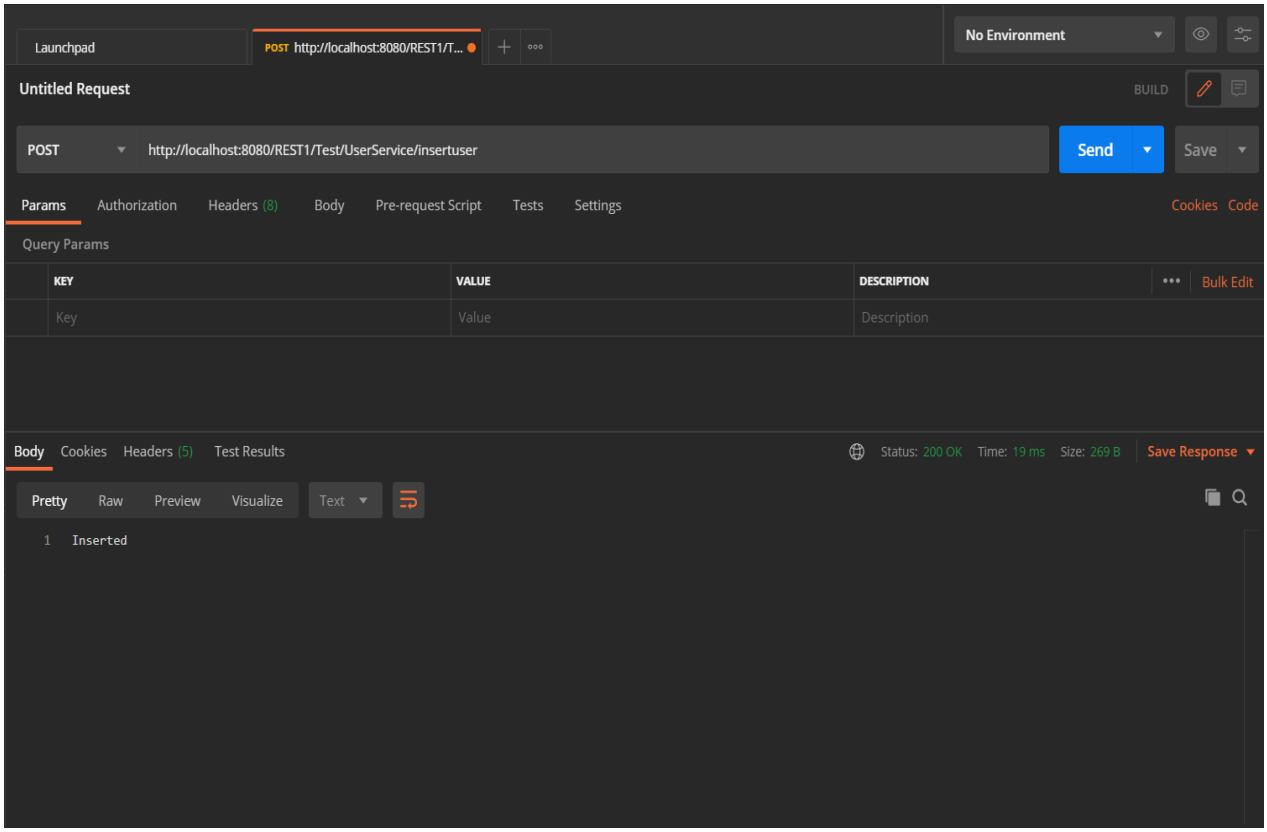
==========

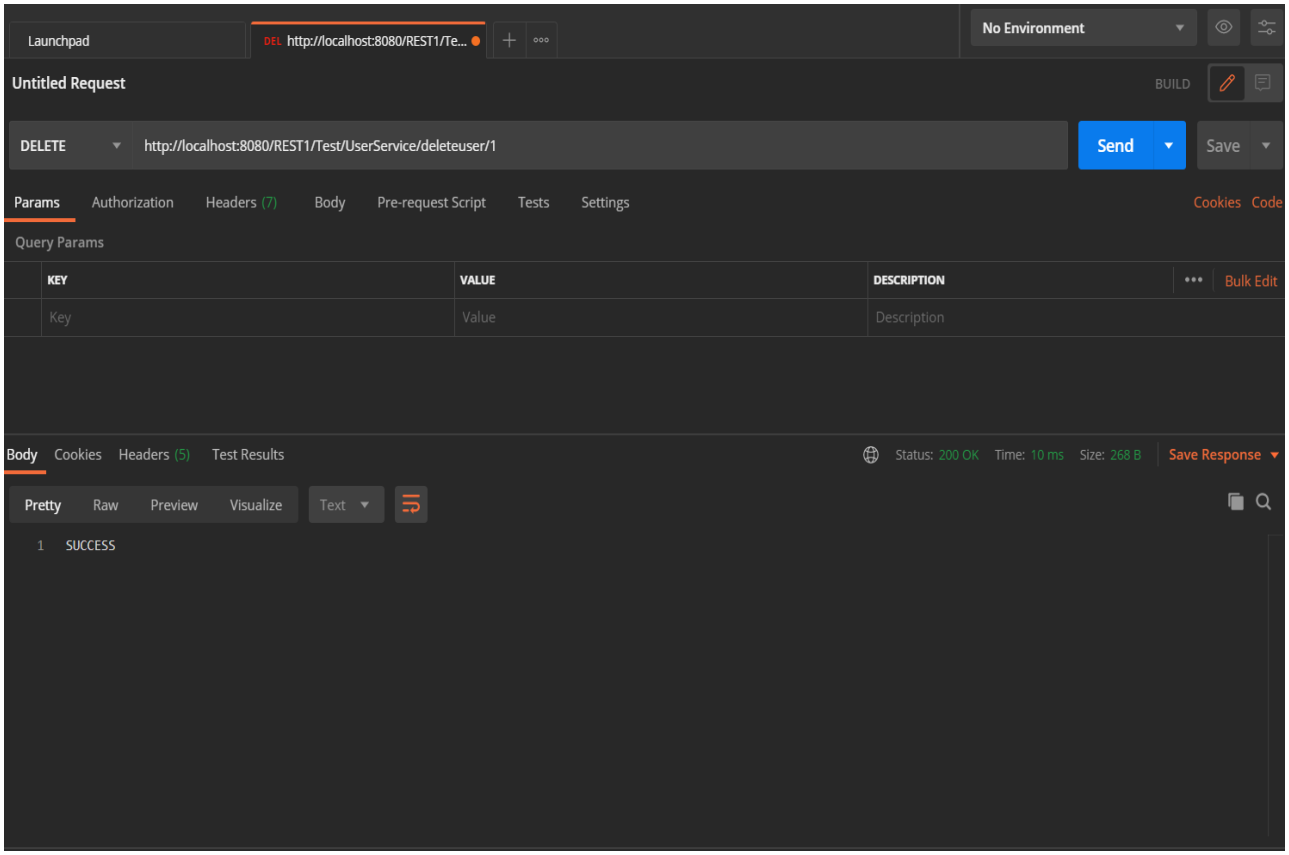
```
package Rest;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.List;

public class UserContext {

   File file = new File("Users23.dat");

   public List<User> getAllUsers() {

      List<User> userList = null;
      try {
         if (!file.exists()) {
            User user = new User(1, "Mahesh", "Teacher");
            userList = new ArrayList<User>();
            userList.add(user);
            saveUserList(userList);
         } else {
            FileInputStream fis = new FileInputStream(file);
            ObjectInputStream ois = new ObjectInputStream(fis);
            userList = (List<User>) ois.readObject();
            ois.close();
         }
      } catch (IOException e) {
         e.printStackTrace();
      } catch (ClassNotFoundException e) {
      }
      return userList;
   }

   public void saveUserList(List<User> userList1) {
      try {
         FileOutputStream fos;
         fos = new FileOutputStream(file);
         ObjectOutputStream oos = new ObjectOutputStream(fos);
         oos.writeObject(userList1);
```

```
            oos.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();

        }
    }

    public User getUser(int id) {
        List<User> users = getAllUsers();
        for (User user : users) {
            if (user.getId() == id) {
                System.out.println("getID is" + user.getId() + "AND ID is" + id);
                return user;
            }
        }
        return null;
    }

    public int deleteUser(int id) {
        System.out.println("Inside Delete User Method of User Context");
        List<User> userList = getAllUsers();
        for (User user : userList) {
            if (user.getId() == id) {
                System.out.println("Inside Delete User getID is" + user.getId() + "AND ID is" + id);
                int index = userList.indexOf(user);
                userList.remove(index);
                saveUserList(userList);
                return 1;
            }

        }
        return 0;
    }

    public int updateUser(User pUser) {
        List<User> userList = getAllUsers();
        for (User user : userList) {
            if (user.getId() == pUser.getId()) {
                int index = userList.indexOf(user);
                userList.set(index, pUser);
                saveUserList(userList);
                return 1;
            }
        }
        return 0;
    }
```

}


Launchpad
GET http://localhost:8080/REST1/Te...
No Environment
Untitled Request
BUILD
GET
http://localhost:8080/REST1/Test/UserService/users
Send
Save
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 4.05 s
Size: 317 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"name": "Mahesh",
"profession": "Teacher"
}
]


Launchpad
POST http://localhost:8080/REST1/T...
No Environment
Untitled Request
BUILD
POST
http://localhost:8080/REST1/Test/UserService/insertuser
Send
Save
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 19 ms
Size: 269 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1 Inserted


Launchpad
DEL http://localhost:8080/REST1/Te...
No Environment
Untitled Request
BUILD
DELETE
http://localhost:8080/REST1/Test/UserService/deleteuser/1
Send
Save
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 10 ms
Size: 268 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1 SUCCESS

# EXPERIMENT-9

**Aim:** Create Microservice based application using Spring Boot.
**Tools/ Apparatus**: Web service, BPEL Runtime Environment: GlassFish Server, GUI-IDE Tool: NetBeans 6.0

==========

**Movie Catelog Service**

==========

```
package io.javabrains.moviecatalogservice.resources;

import io.javabrains.moviecatalogservice.models.CatalogItem;
import io.javabrains.moviecatalogservice.models.Movie;
import io.javabrains.moviecatalogservice.models.Rating;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

import javax.websocket.server.PathParam;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.stream.Collectors;

@RestController
@RequestMapping("/catalog")
public class MovieCatalogResource {

    @RequestMapping("/{userId}")
    public List<CatalogItem> getCatalog(@PathParam("userId") String userId) {

        RestTemplate restTemplate = new RestTemplate();


        // Get all rated movieIds

        List<Rating> ratings = Arrays.asList(
                new Rating("1", 4),
                new Rating("2", 3)
        );

        // For each movie call the movie info service and get details

        return ratings.stream().map(rating -> {
            Movie movie = restTemplate.getForObject("http://localhost:8081/movies/" + rating.getMovieId(), Movie.class);
            return new CatalogItem(movie.getName(), "Murder/Mystery", rating.getRatings());
        }).collect(Collectors.toList());

        // Put them all together

    }
}
```

==========

**CatelogItems.java**

==========

```
package io.javabrains.moviecatalogservice.models;

public class CatalogItem {

    private String name;
    private String desc;
    private int ratings;

    public CatalogItem(String name, String desc, int ratings) {
        this.name = name;
        this.desc = desc;
        this.ratings = ratings;
    }

    public String getName() {
        return name;
    }

    public String getDesc() {
        return desc;
    }

    public int getRatings() {
        return ratings;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }

    public void setRatings(int ratings) {
        this.ratings = ratings;
    }
}
```

==========

**Movie.java**

==========

```
package io.javabrains.moviecatalogservice.models;

public class Movie {
    private int movieId;
    private String name;

    public Movie() {
    }

    public Movie(int movieId, String name) {
        this.movieId = movieId;
        this.name = name;
    }

    public int getMovieId() {
        return movieId;
    }

    public void setMovieId(int movieId) {
        this.movieId = movieId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

==========

**Rantings.java**

==========

```
package io.javabrains.moviecatalogservice.models;

public class Rating {
    private String movieId;
    private int ratings;

    public Rating(String movieId, int ratings) {
        this.movieId = movieId;
        this.ratings = ratings;
    }
```

```
    public String getMovieId() {
        return movieId;
    }

    public void setMovieId(String movieId) {
        this.movieId = movieId;
    }

    public int getRatings() {
        return ratings;
    }

    public void setRatings(int ratings) {
        this.ratings = ratings;
    }
}
```

==========

**Movie Info Service**

==========

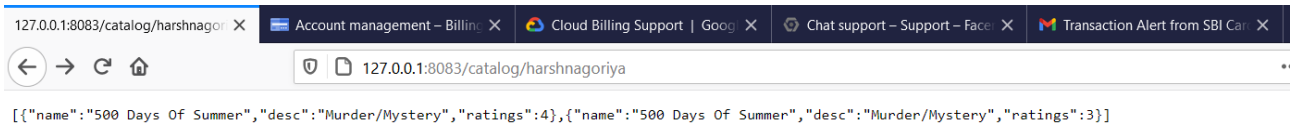
```
package io.javabrains.movieinfoserice.movieinfoservice.resources;

import io.javabrains.movieinfoserice.movieinfoservice.models.Movie;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import javax.websocket.server.PathParam;

@RestController
@RequestMapping("/movies")
public class MovieResource {

    @RequestMapping("/{movieId}")
    public Movie getMovieInfo(@PathParam("movieId") String movieId) {
        return new Movie(
                1, "500 Days Of Summer"
        );
    }
}
```

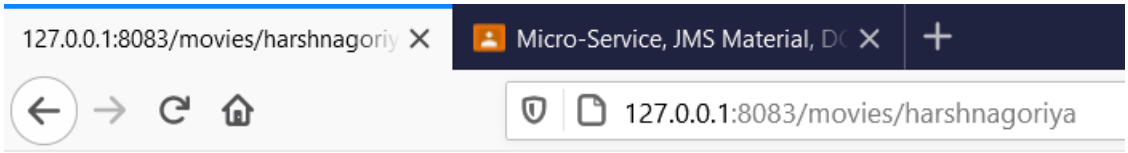
==========

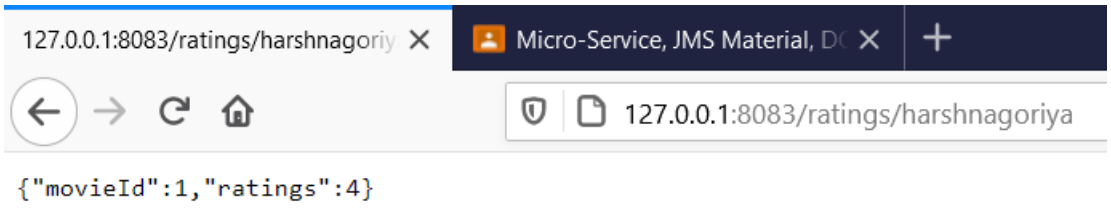
**Rating Service**

==========

```
package io.javabrains.ratingdataservice.ratingdataservie.resources;

import io.javabrains.ratingdataservice.ratingdataservie.models.Rating;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import javax.websocket.server.PathParam;

@RestController
@RequestMapping("/ratings")
public class RatingResource {
    @RequestMapping("/{movieId}")
    public Rating getRating(@PathParam("movieId") String movieId) {
        return new Rating(
                1, 4
        );
    }
}
```



# EXPERIMENT-10

**Aim:** Implementation JMS based application using Publish-Subscribe paradigm.
**Tools/ Apparatus:** Web service, BPEL Runtime Environment: GlassFish Server, GUI-IDE Tool: NetBeans 6.0, JMS, ESB: WSO2

==========

**MyListerner.java**

==========

```
package Demo;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;


public class MyListener implements MessageListener {

   @Override
   public void onMessage(Message message) {
      try{
      TextMessage msg=(TextMessage)message;
      System.out.println("following message is received:"+msg.getText());
      }
      catch(JMSException e)
      {System.out.println(e);}
   }
}
```

==========

**Reviewer.java**

==========

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.jms.*;
import javax.naming.InitialContext;

public class Receiver extends HttpServlet {

   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try{
//1) Create and start connection
         InitialContext ctx=new InitialContext();
         QueueConnectionFactory
         f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");
```

```
            QueueConnection con=f.createQueueConnection();
            con.start();
            //2) create Queue session
            QueueSession ses=con.createQueueSession(false,
            Session.AUTO_ACKNOWLEDGE);
            //3) get the Queue object
            Queue t=(Queue)ctx.lookup("myQueue");
            //4)create QueueReceiver
            QueueReceiver receiver=ses.createReceiver(t);
            //5) create listener object
            MyListener listener=new MyListener();
            //6) register the listener object with receiver
            receiver.setMessageListener(listener);
            System.out.println("Receiver1 is ready, waiting for messages...");
            System.out.println("press Ctrl+c to shutdown...");
        }
        catch(Exception e){System.out.println(e);}

    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the
left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
```

```
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

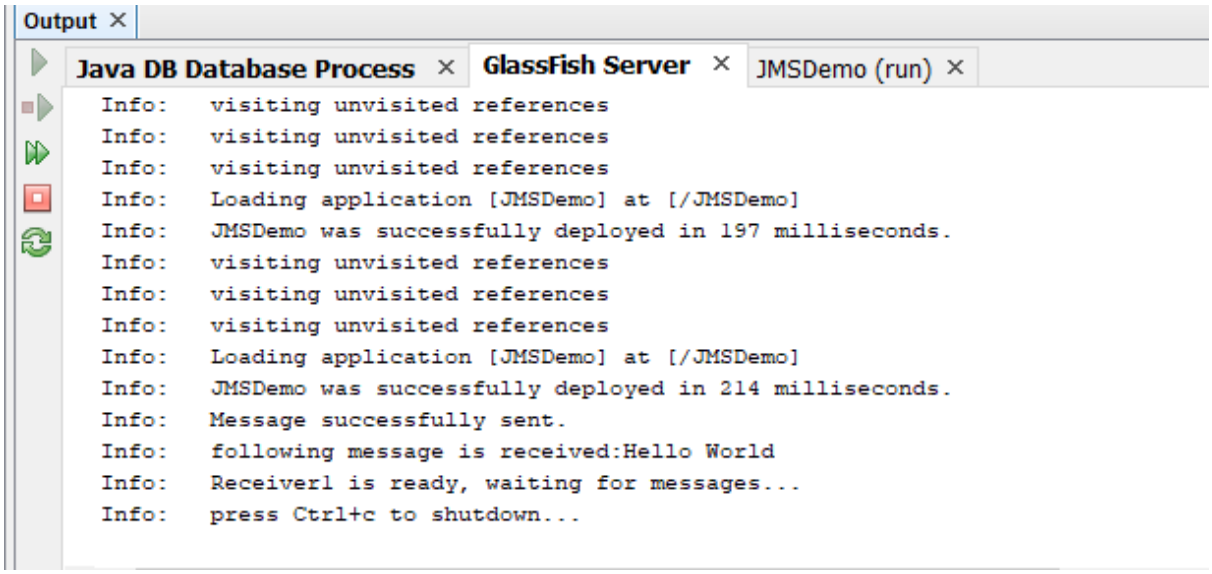
==========

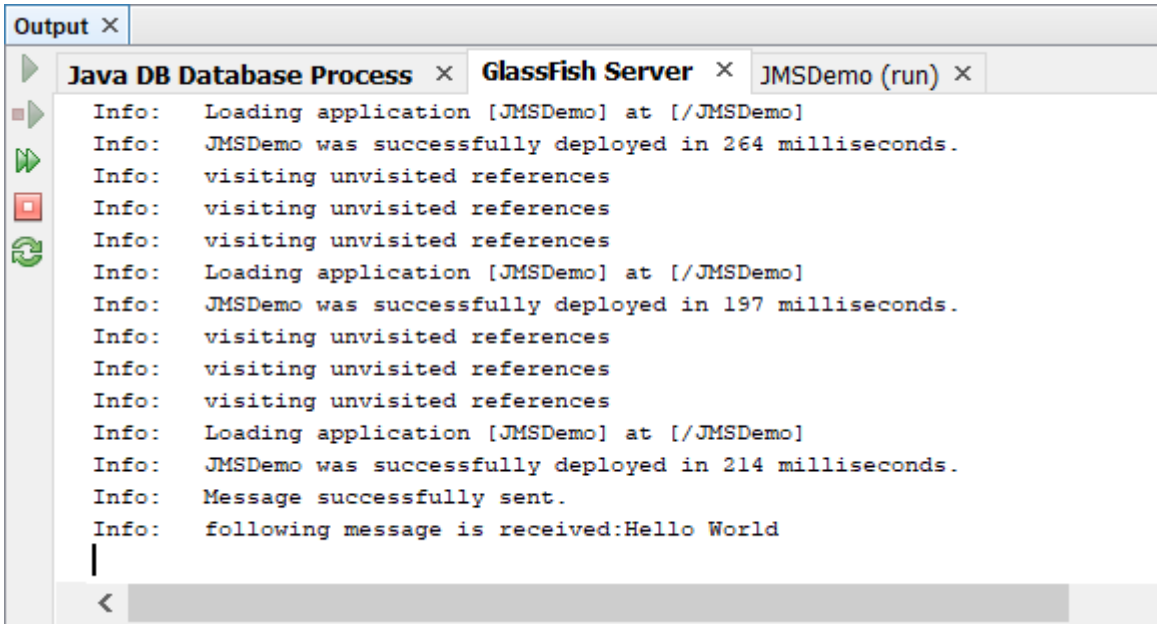
**Sender.java**

==========

```
package Demo;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.naming.*;
import javax.jms.*;

public class Sender extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        try
        { //Create and start connection
            InitialContext ctx=new InitialContext();
            QueueConnectionFactory
            f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");
            QueueConnection con=f.createQueueConnection();
            con.start();
            //2) create queue session
            QueueSession ses=con.createQueueSession(false,
            Session.AUTO_ACKNOWLEDGE);
            //3) get the Queue object
            Queue t=(Queue)ctx.lookup("myQueue");
            //4)create QueueSender object
            QueueSender sender=ses.createSender(t);
            //5) create TextMessage object
            TextMessage msg=ses.createTextMessage();
```

```
            //6) write message
            msg.setText("Hello World");
            sender.send(msg);

            System.out.println("Message successfully sent.");
        //8) connection close
            con.close();
        }
        catch(Exception e)
        {System.out.println(e);}
            }

    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```


Output
Java DB Database Process
GlassFish Server
JMSDemo (run)
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 197 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 214 milliseconds.
Info: Message successfully sent.
Info: following message is received:Hello World
Info: Receiver1 is ready, waiting for messages...
Info: press Ctrl+c to shutdown...


Output
Java DB Database Process
GlassFish Server
JMSDemo (run)
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 264 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 197 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 214 milliseconds.
Info: Message successfully sent.
Info: following message is received:Hello World