|  |  |
| --- | --- |
| **Roll No.:** A059 | **Name:** Chinmay Parikh |
| **Prg/Yr/Sem:** B.Tech(I.T.)/4th /7 | **Batch:** A3 |
| **Date of Experiment:** 13/9/2014 | **Date of Submission:** 29/10/2014 |

**Aim:** Creating a Web Service which can add & display the records of a student in the exam table.

**Scenario:** Creating a Web Service which can add & display the records of a student in the exam table using Derby Database.

**Detailed Steps for creating a web service & web service client in Net Beans 7.0.**

Connecting database:

• Open the Netbeans software

• Navigate to the services tab

• Open database folder and select the derby database Right Click and select start the server

• Select the sample database or you can create one

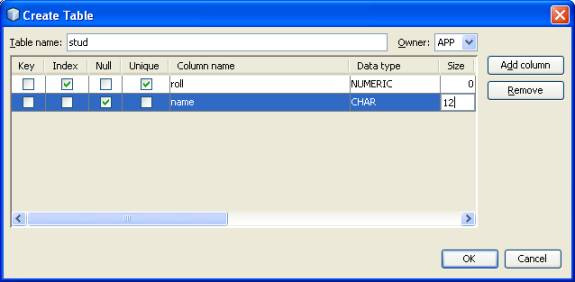
• Right Click and Select Connect. It will connect to the sample database

• In this dialog box give the table name as stud

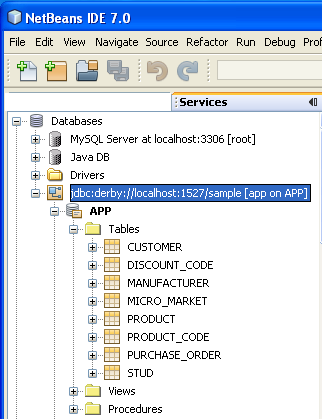
• Now add the columns

• Add the two columns

• Roll with data type numeric and Name with data type char



It creates a stud table a sample database



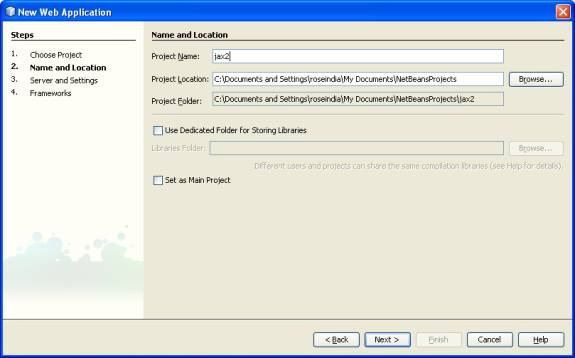
Creating the Java Web Project:

1. Click File and then New Project.

2. For New Project: On the Categories side, choose Java Web and on the Projects side, select WebApplication.

3. Click Next.

4. For Name and Location, under Project Name, enter Jax2.

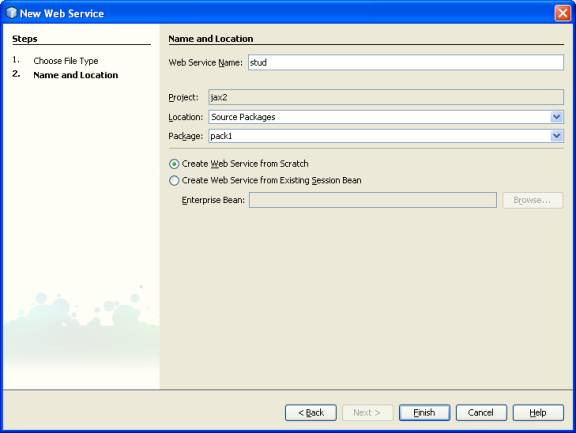


Create a web service

• Right Click on the project

• Select New->Web Service

• Type name as stud and package as pack1

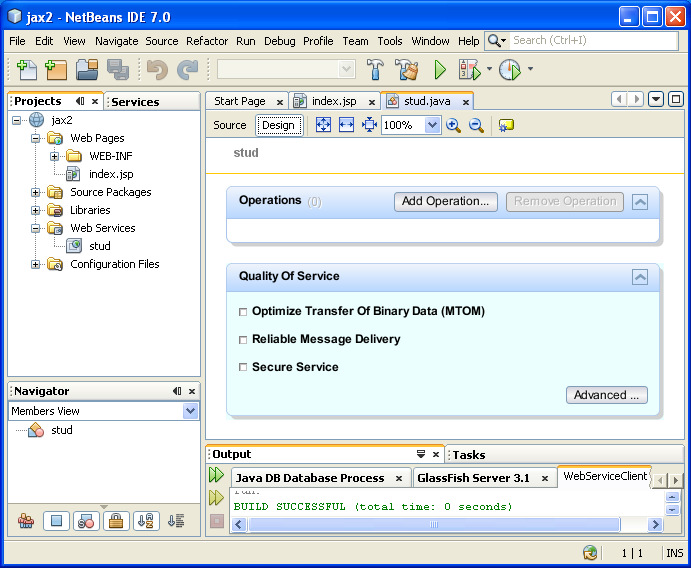


This will create a stud Web Service class

Adding operation

• In the design view Click on the Add Operation

• It will generate an Add operation Dialog Box

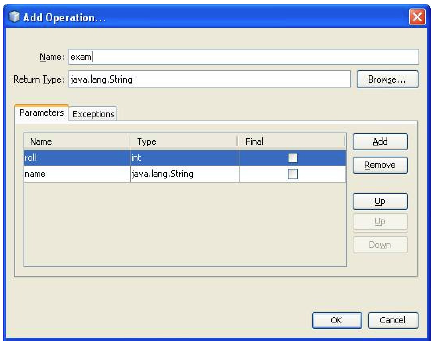


In the pop up dialog box give the operation name and parameters

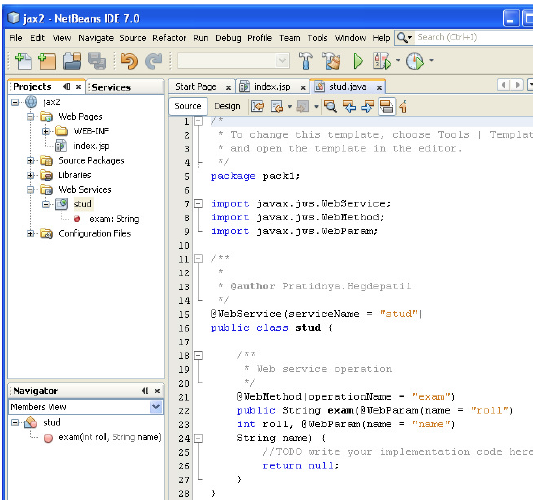
o Operation name exam

o Return type String

o Parameters name roll and name



It creates a full web service class



Adding the code into stud.java :

package pack1;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.annotation.Resource;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.sql.DataSource;

@WebService(serviceName = "stud")

public class stud {

@Resource(name = "data1")

private DataSource data1;

/\*\*

\* Web service operation

\*/

@WebMethod(operationName = "exam")

public String exam(@WebParam(name = "roll")

int roll, @WebParam(name = "name")

String name) {

String status="record not inserted";

try {

Connection con=data1.getConnection();

PreparedStatement ps=con.prepareStatement("INSERT INTO stud VALUES(?,?)");

ps.setInt(1,roll);

ps.setString(2,name);

int i=ps.executeUpdate();

if(i!=0) {

status="record inserted";

}

}

catch(Exception e){

System.out.println("error in strong data"+e);

}

return status;

}

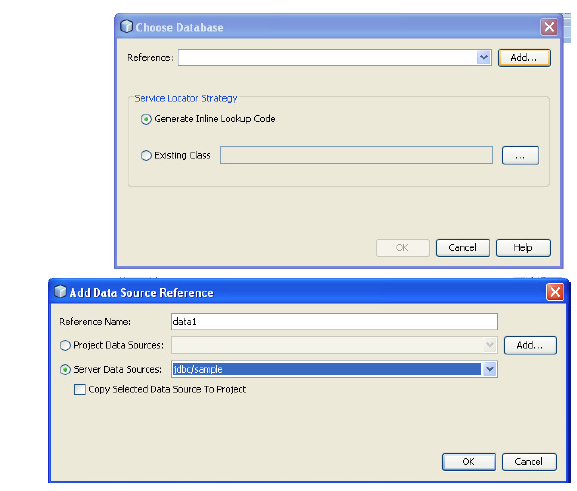
}

Adding Database capabilities:

• Database and table created above is used in the web service

• Right Click in the code of Web Service

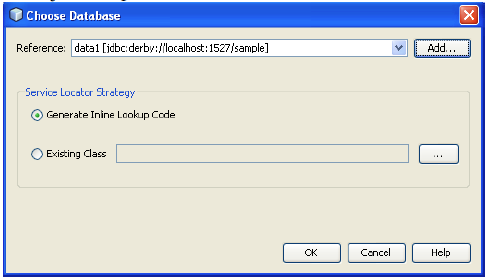
• Select the Insert Code -> Use Database



Add the data source reference

• Type Reference name data1

• Select jdbc/sample Server Data Source



This creates Data Source reference variable data1 at the top of method

Running the project:

* Build the above created project
* Deploy the project on the server
* This deploys the project on the server
* Run our Web Service by Right Clicking on Web Service stud
* Select Test Web Service



To check the inserted values in the table:

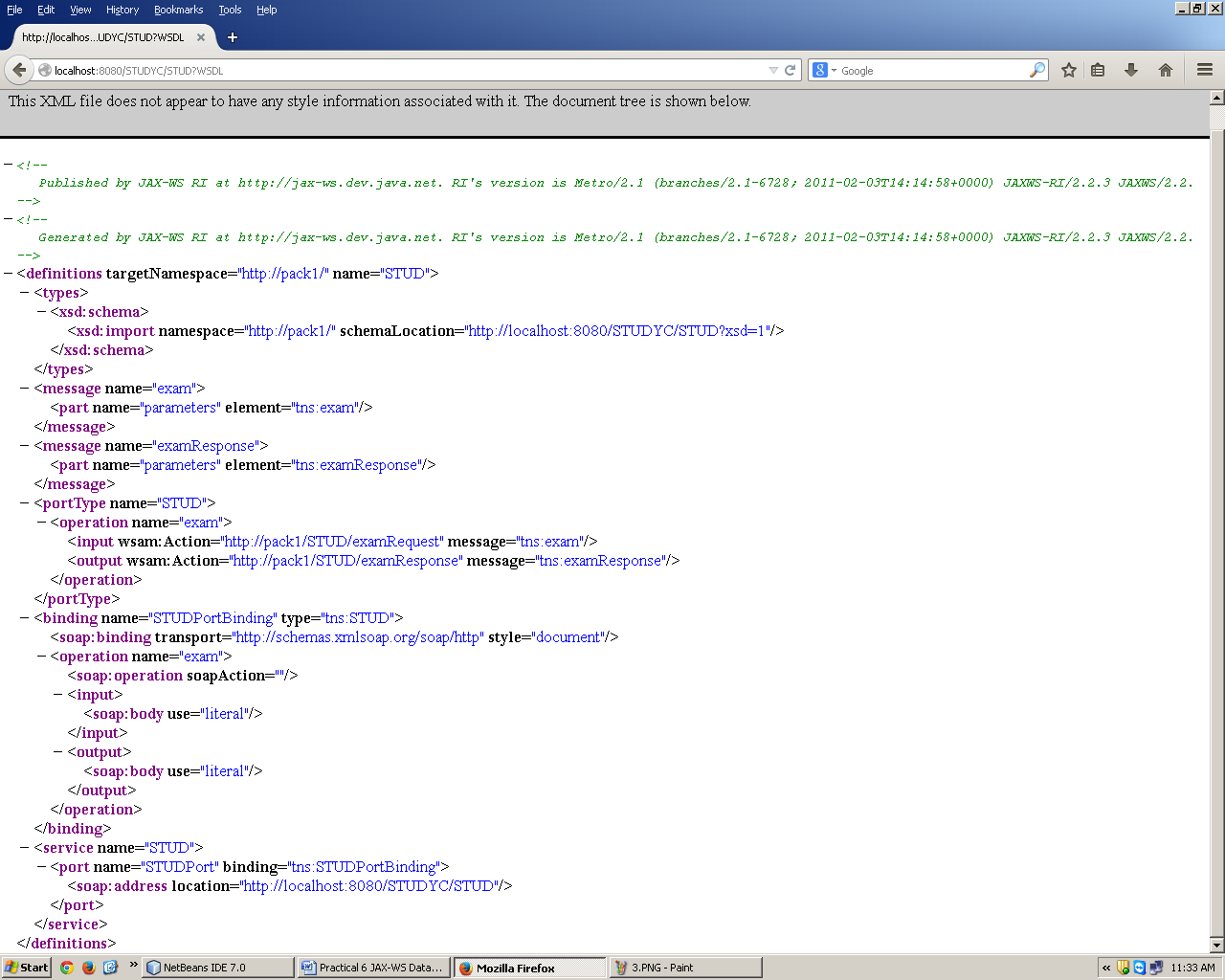
• Right Click on the stud table in derby database

• Select View Data it shows the inserted data

This ends the database web service practical.

**Printout:**

• WSDL.



• SOAP Request & Response.



• Stud.java.

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package pack1;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.annotation.Resource;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.sql.DataSource;

@WebService(serviceName = "STUD")

public class STUD {

@Resource(name = "data1")

private DataSource data1;

/\*\*

\* Web service operation

\*/

@WebMethod(operationName = "exam")

public String exam(@WebParam(name = "roll")

int roll, @WebParam(name = "name")

String name) {

String status="record not inserted";

try {

Connection con=data1.getConnection();

PreparedStatement ps=con.prepareStatement("INSERT INTO studyc VALUES(?,?)");

ps.setInt(1,roll);

ps.setString(2,name);

int i=ps.executeUpdate();

if(i!=0) {

status="record inserted";

}

}

catch(Exception e){

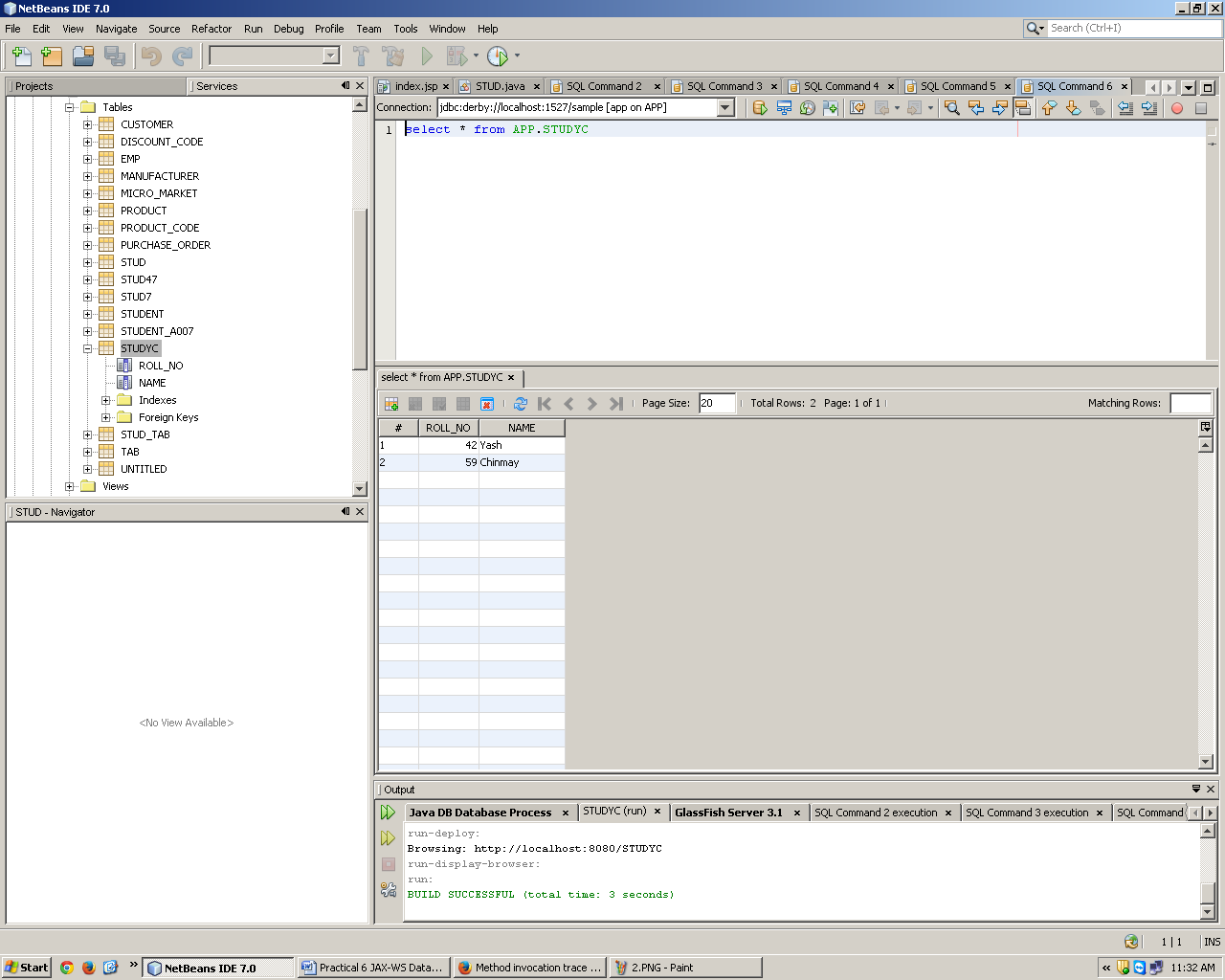
System.out.println("error in strong data"+e);

}

return status;

}

}



**Conclusion:** Thus we have identified the need for accessing a web service within an application and also created a client web service to access the online web service.