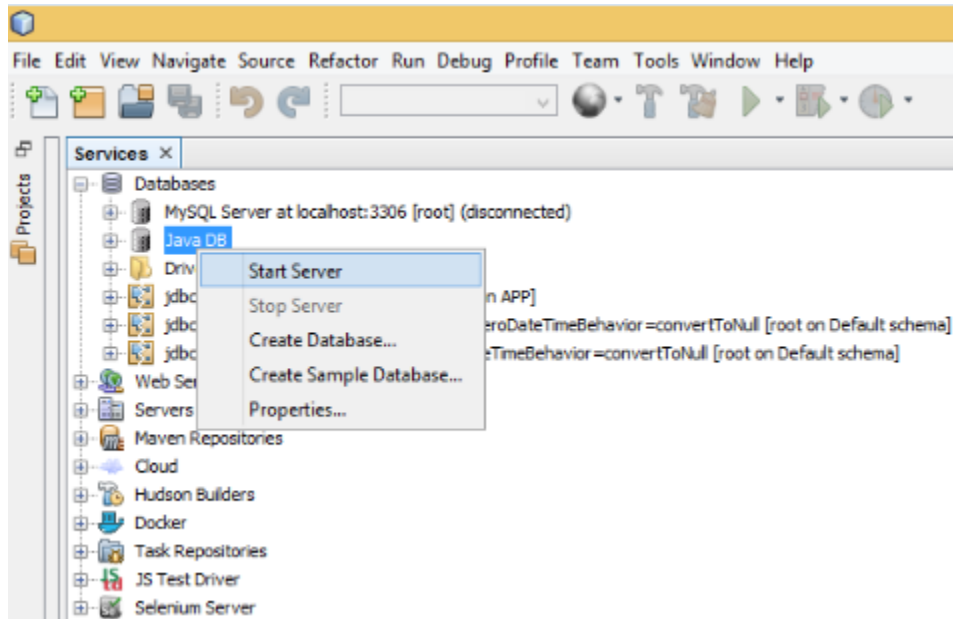


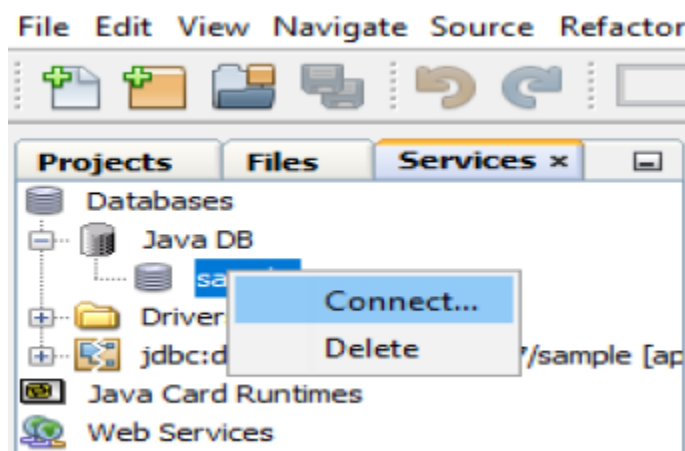
## Practical:5

**Aim:** write a program to implement the operation can receive request and will return a response in two ways a)one way operation b)request and response

Step 1: click on the services > Databases . **Right click on Java DB** and then **click on Start Server** to start the server .



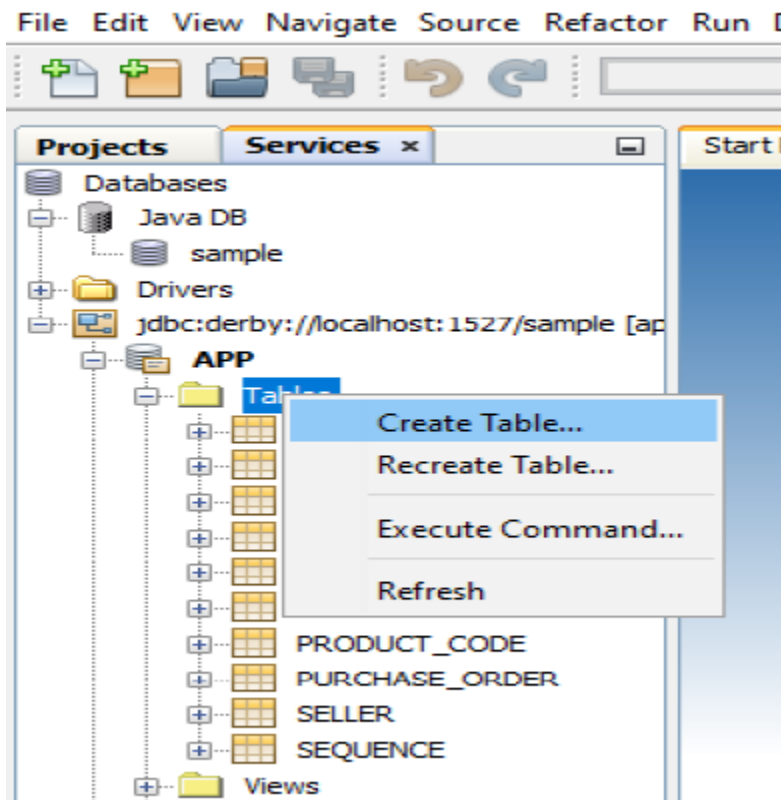
2) Now expand Java DB and **right click on sample** and then **click on connect** to connect the sample database with server.



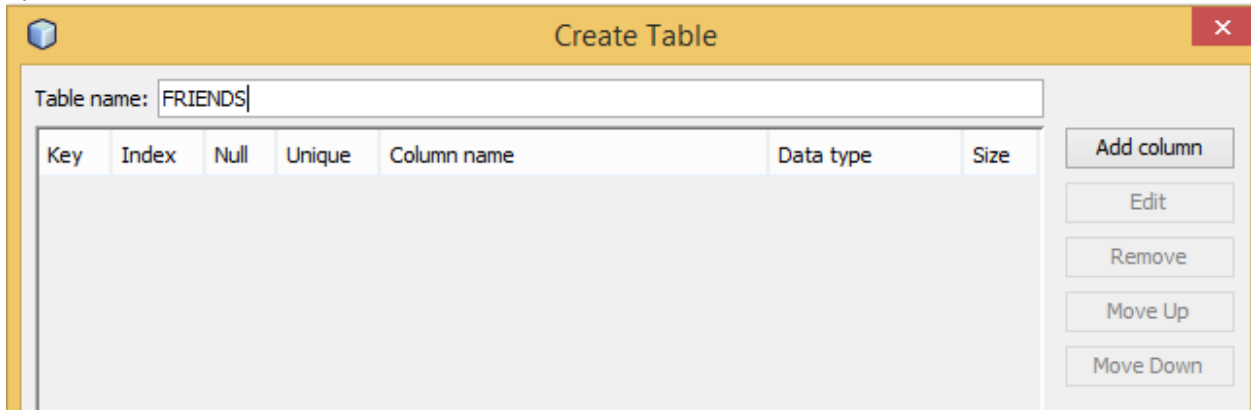
## Practical:5

3) Now we are going to create a table in default database **sample**.

**Right click on Table -> Create Table**

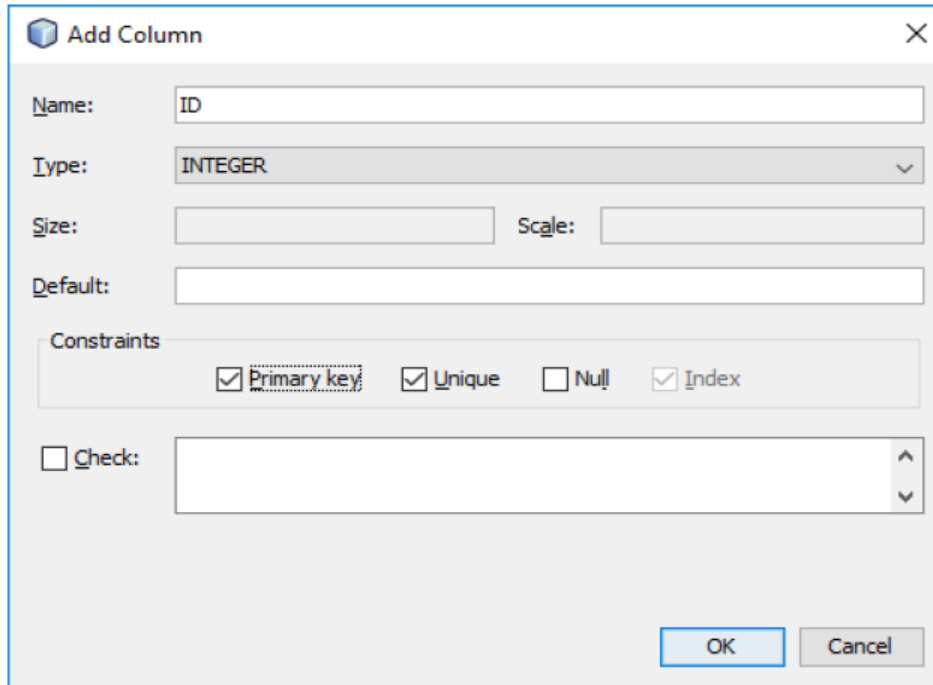


4) Give **table name** as **FRIENDS**.



## Practical:5

5) Now **click on Add column button to add columns** in table.  
Enter details as in below pic and **select Primary key**. After that **click on OK** button.

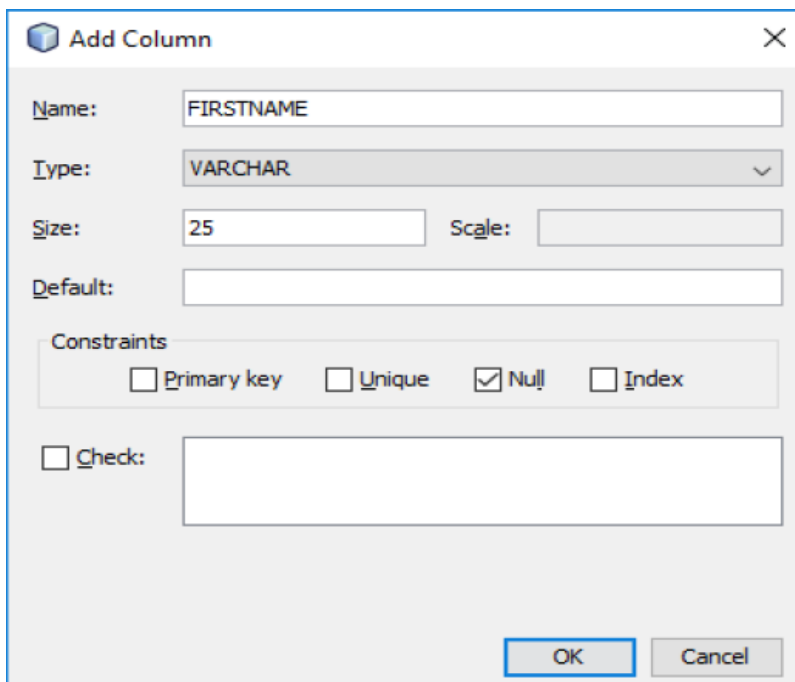


The 'Add Column' dialog box shows the following configuration for the first column:

- Name:** ID
- Type:** INTEGER
- Size:** (empty)
- Scale:** (empty)
- Default:** (empty)
- Constraints:** ☒ Primary key, ☒ Unique, ☐ Null, ☒ Index
- Check:** (empty)

Buttons: OK, Cancel

6) Now **add second column** with following detail. But **don't select primary** and click on OK button.



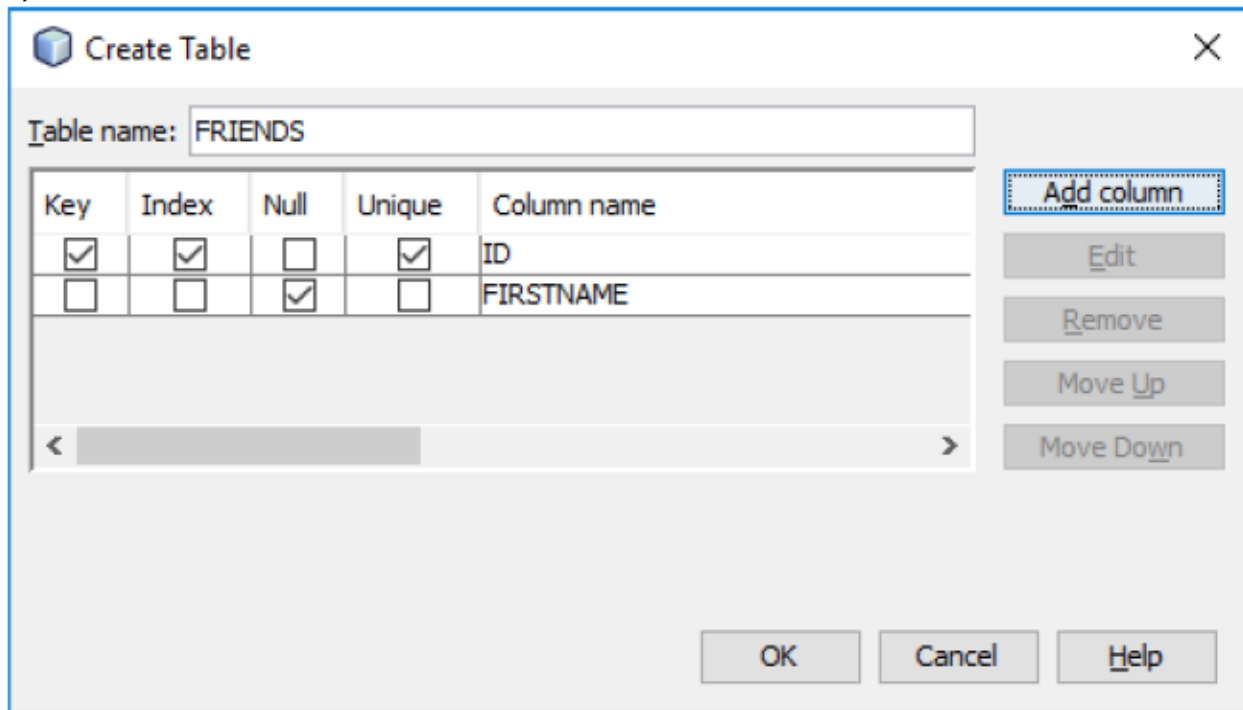
The 'Add Column' dialog box shows the following configuration for the second column:

- Name:** FIRSTNAME
- Type:** VARCHAR
- Size:** 25
- Scale:** (empty)
- Default:** (empty)
- Constraints:** ☐ Primary key, ☐ Unique, ☒ Null, ☐ Index
- Check:** (empty)

Buttons: OK, Cancel

## Practical:5

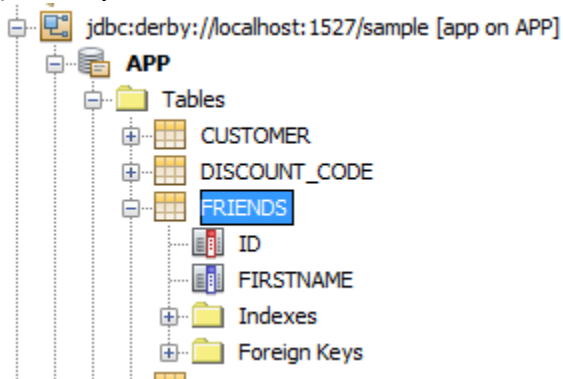
7) Now **click on OK** button.



The 'Create Table' dialog box shows the table name 'FRIENDS'. It contains a table with columns: Key, Index, Null, Unique, and Column name. The 'ID' column has 'Key' and 'Unique' checked. The 'FIRSTNAME' column has 'Null' checked. To the right of the table are buttons: 'Add column' (highlighted with a dashed border), 'Edit', 'Remove', 'Move Up', and 'Move Down'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

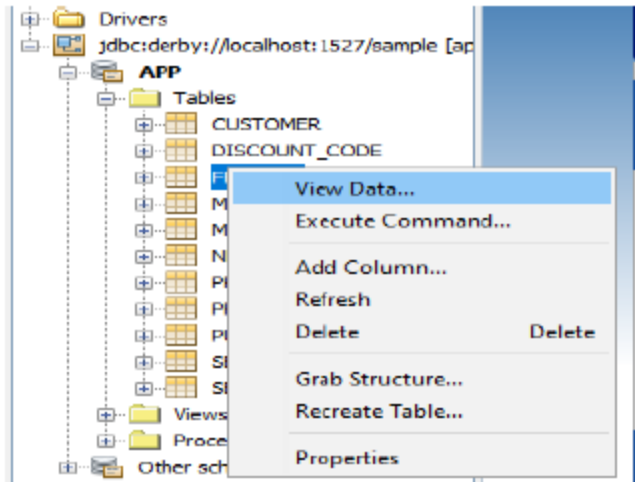
Key	Index	Null	Unique	Column name
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ID
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	FIRSTNAME

8) Now you can see a table with name **FRIENDS** in the table.

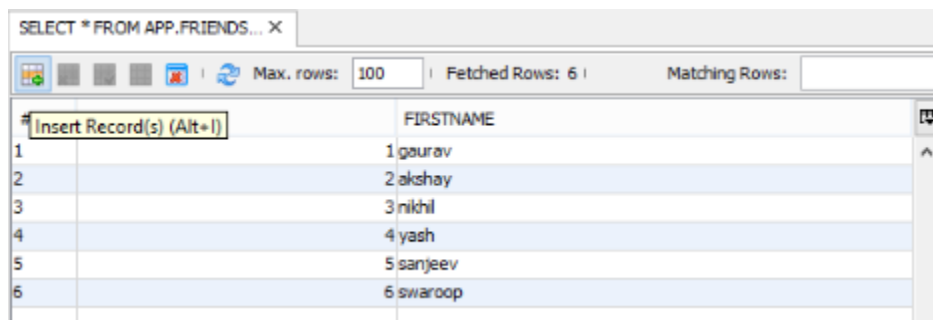


9) **Right click on FRIENDS** to view and add records into it.

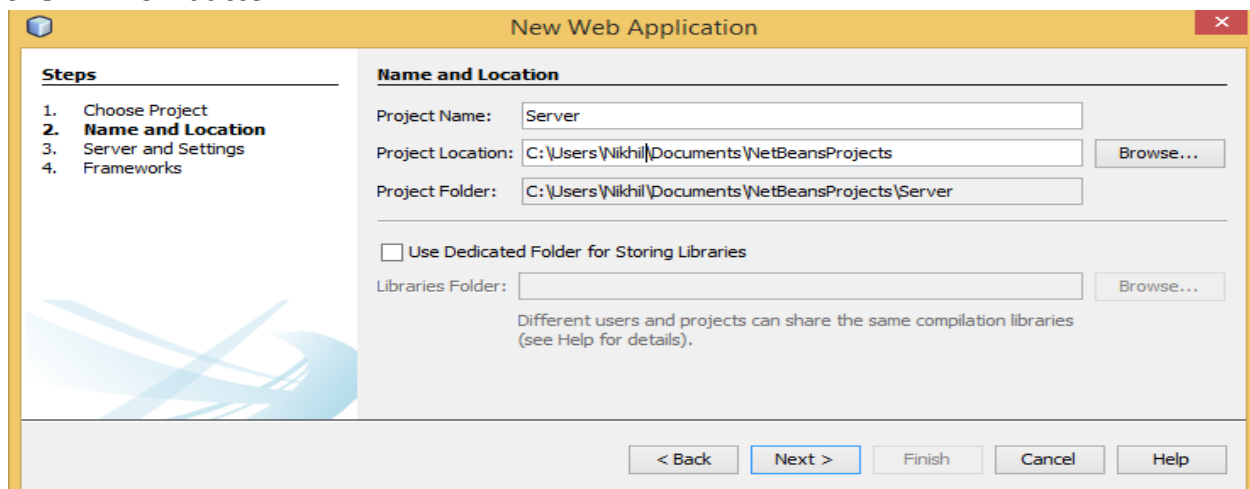
## Practical:5



10) Now **click on the leftmost icon in second panel to insert some record.** **Insert a record** and then **click on Add Row button** to insert more record. After that **click on OK button** to finish.

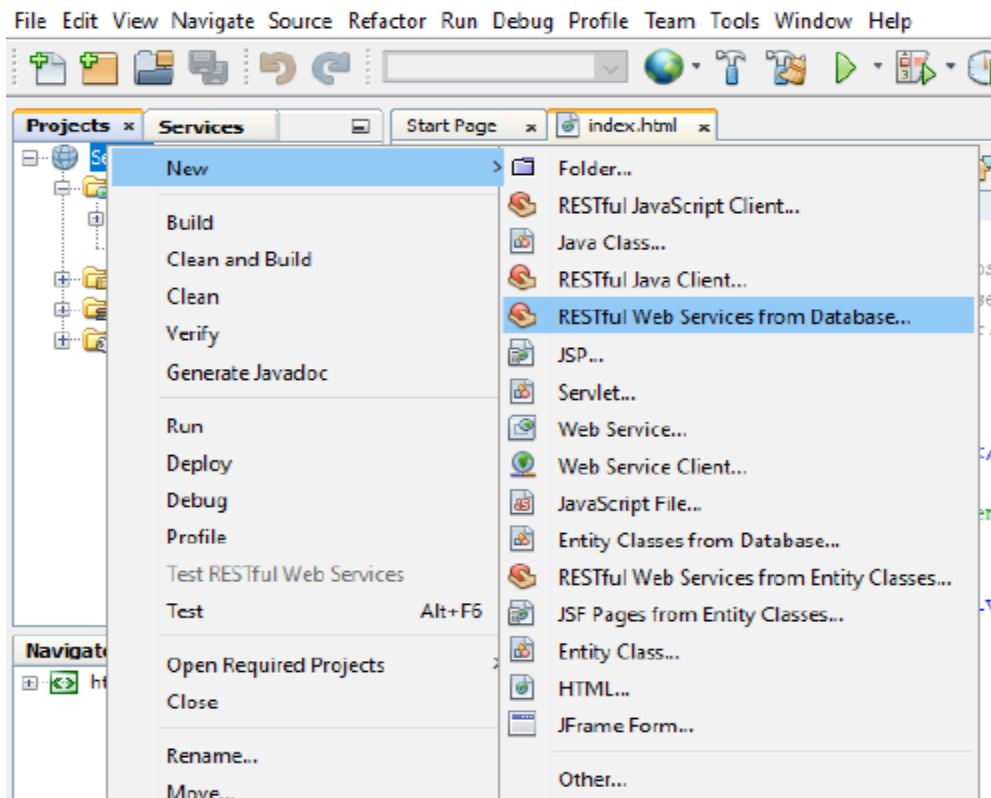


11) Now **create a web application** with name **Server**. After that **click on Next** and then **Finish** button.

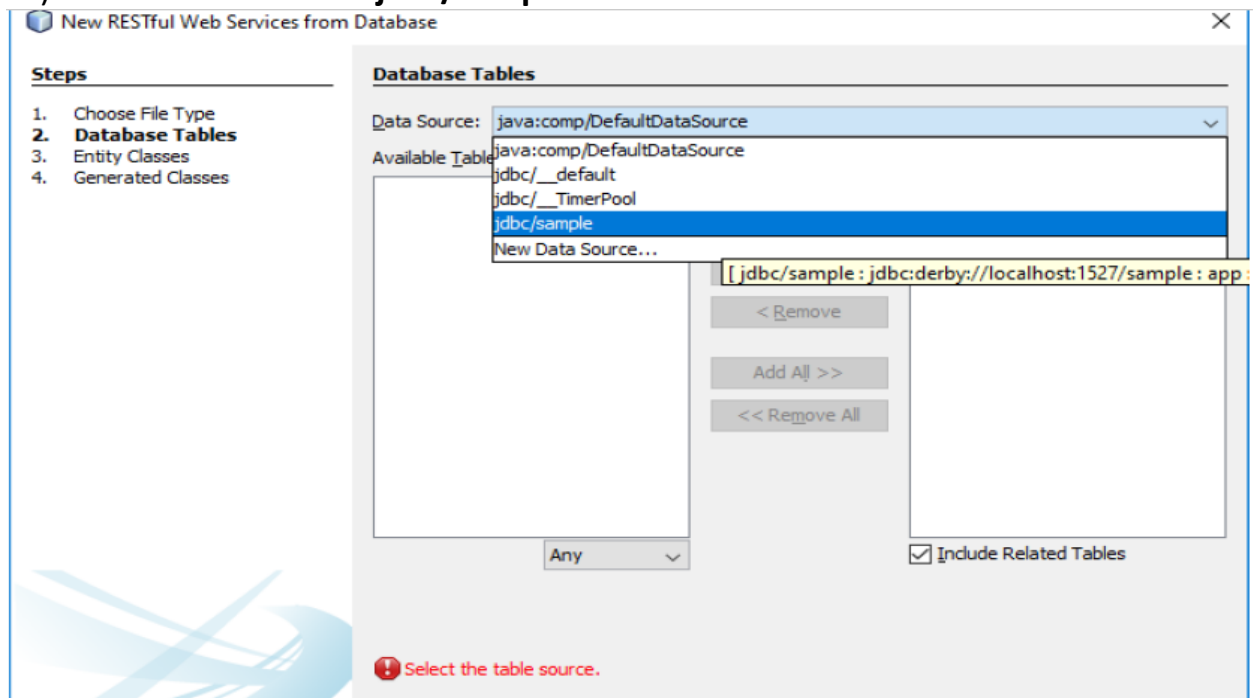


## Practical:5

12) Now create a RESTful Web Service from Database by right click on project name.



13) Choose Data Source jdbc/sample.



## Practical:5

14) Now select **FRIENDS** and click on **Add** button. After that click on **Next** button.

**Steps**

1. Choose File Type
2. **Database Tables**
3. Entity Classes
4. Generated Classes

**Database Tables**

Data Source: jdbc/sample

Available Tables:

- CUSTOMER
- DISCOUNT\_CODE
- FRIENDS**
- MANUFACTURER
- MICRO\_MARKET
- NEWENTITY
- PRODUCT
- PRODUCT\_CODE
- PURCHASE\_ORDER
- SELLER
- SEQUENCE

Selected Tables:

☒ Include Related Tables

Select at least one table.

< Back   Next >   Finish   Cancel   Help

15) Enter Package name as **com.kk** and click on **Next** button and then **Finish**.

**Steps**

1. Choose File Type
2. Database Tables
3. **Entity Classes**
4. Generated Classes

**Entity Classes**

Specify the names and the location of the entity classes.

Class Names:

Database Table	Class Name	Generation Type
FRIENDS	Friends	New

Project: Server

Location: Source Packages

Package: com.kk

☒ Generate Named Query Annotations for Persistent Fields

☒ Generate JAXB Annotations

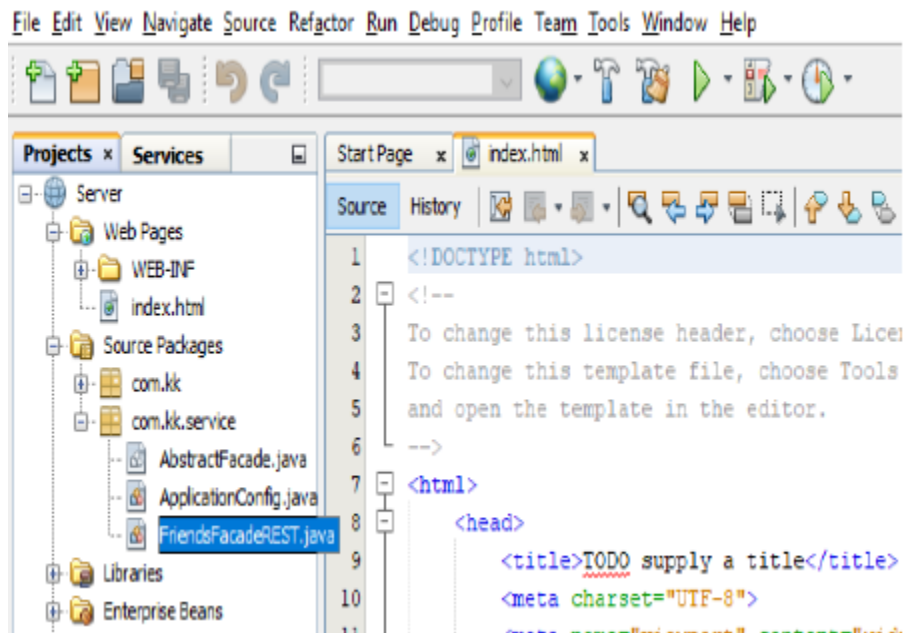
☐ Generate MappedSuperclasses instead of Entities

☒ Create Persistence Unit

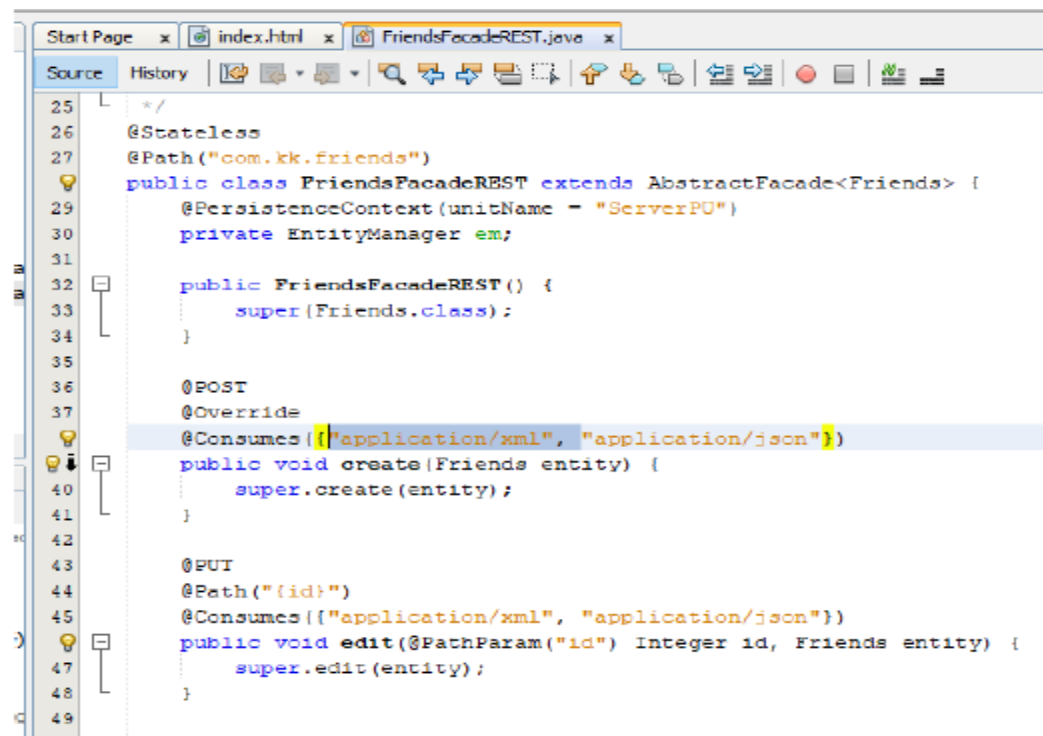
< Back   Next >   Finish   Cancel   Help

## Practical:5

16) Now open selected file by double click on it.



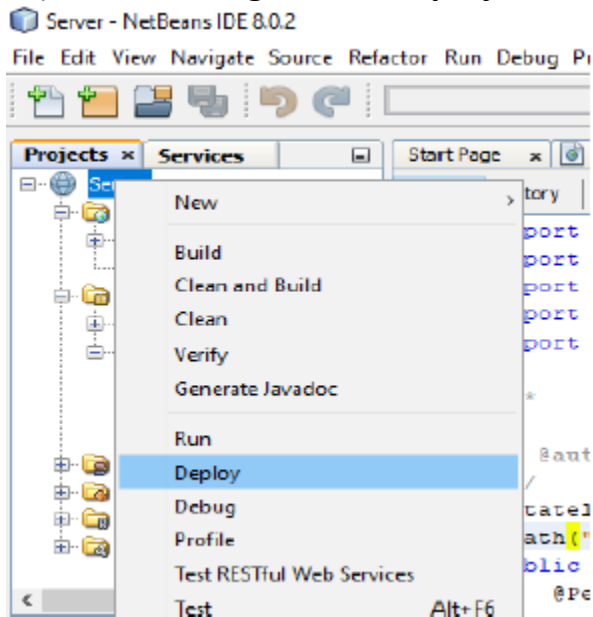
17) Now remove the selected part from every method in this file. So that it will communicate only in JSON format. You can also use methods to convert it. But this is easiest method.



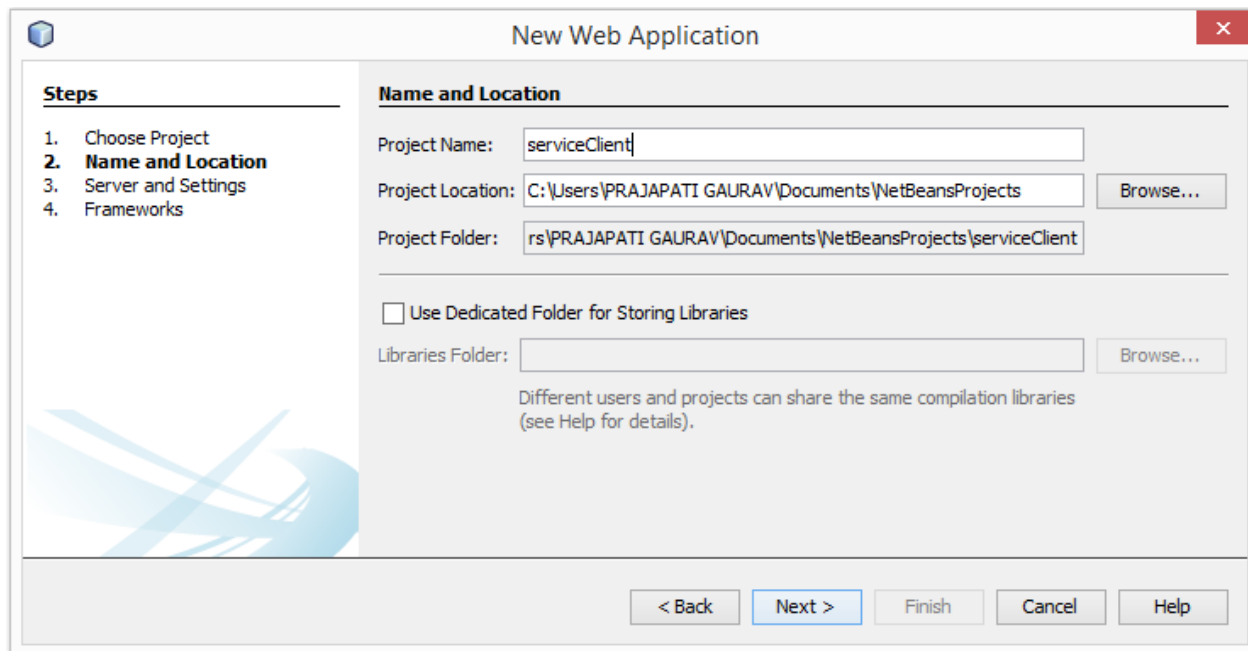


## Practical:5

18) After that **right click** on project name and **Deploy** it.

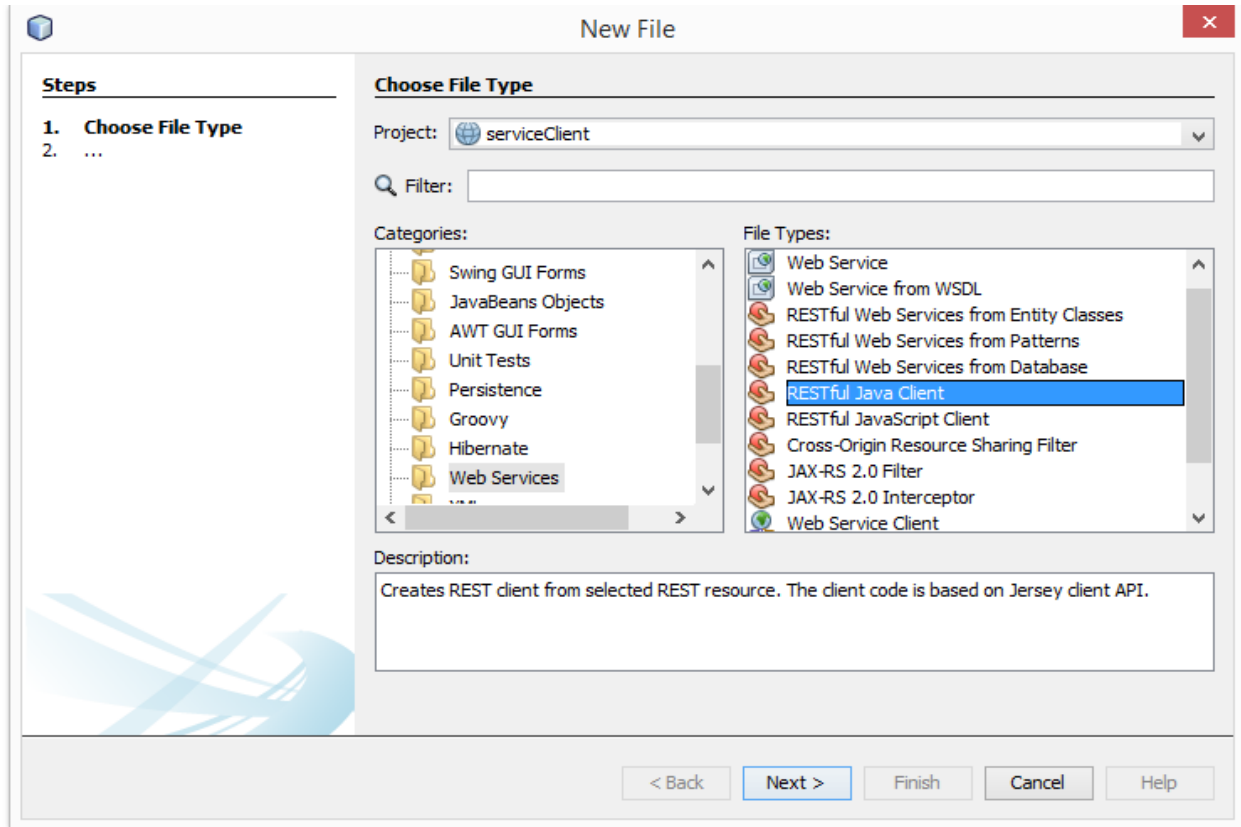


19) Create a Web Application with name ServiceClient.

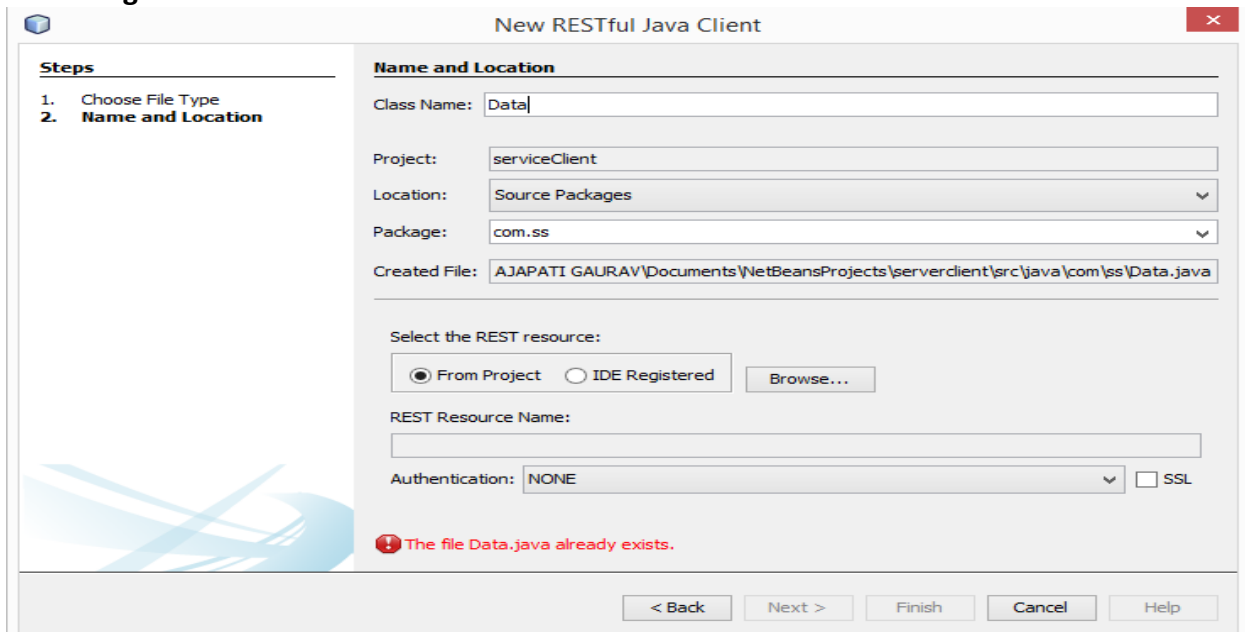


## Practical:5

20) Now create a RESTful Java Client. Right click on ServiceClient -> New -> Other. Drag down and select Web Services and in side panel select RESTful Java Client. And click on next.

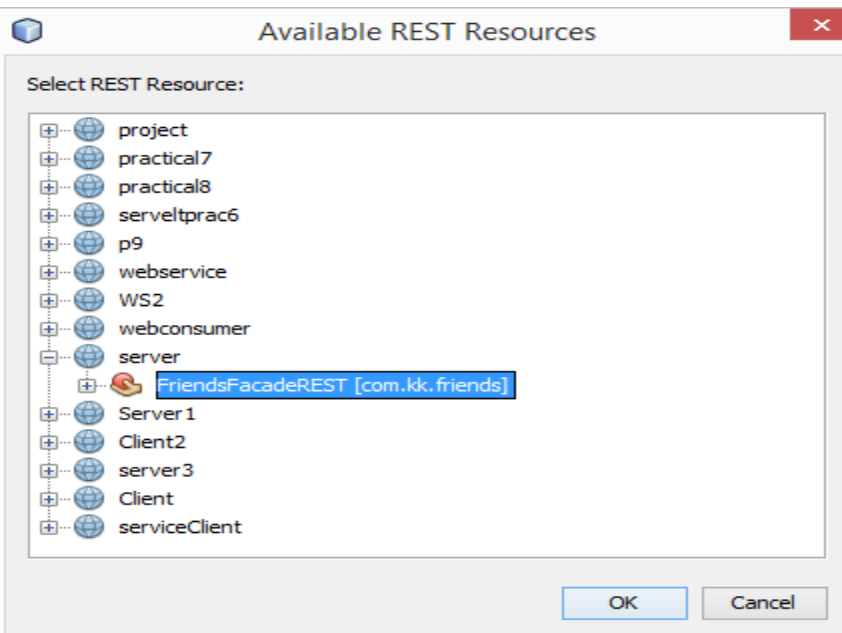


21) Enter following data. Class Name -> Data  
Package -> com.ss

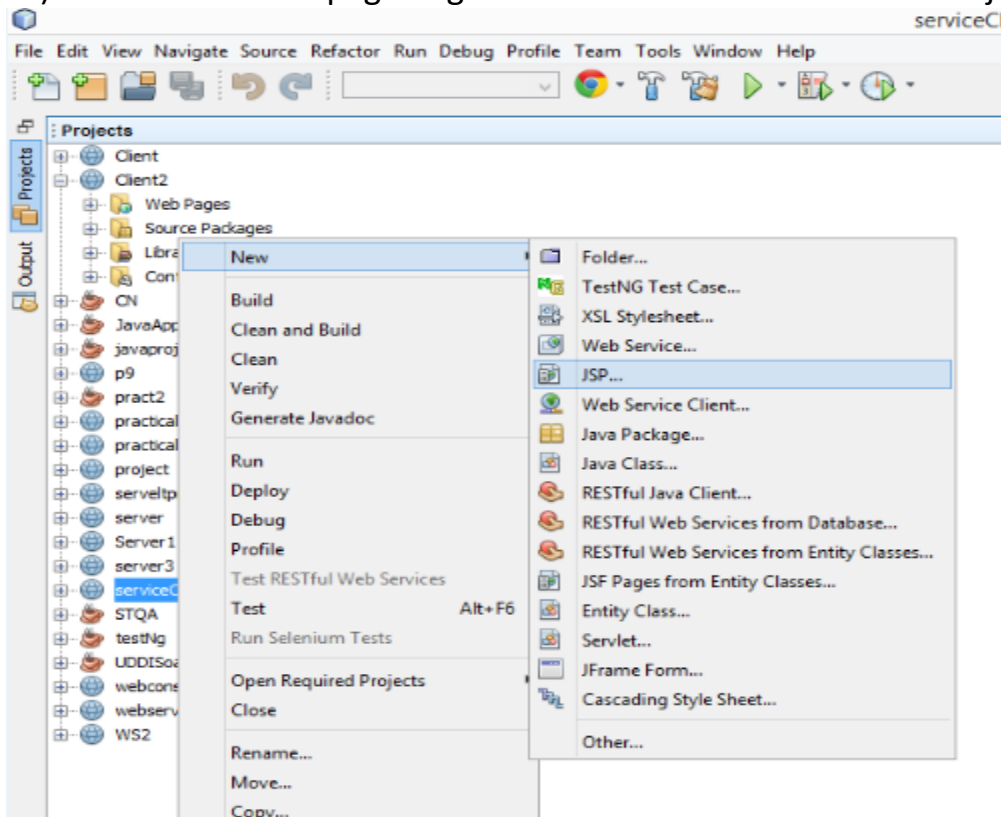


## Practical:5

22) Now click on Browse button and select the option into the below pic. After select click on OK button. Then click on finish .



23) Now create a JSP page. Right click on ServiceClient -> New -> jsp



## Practical:5

24) Enter File Name Del\_data and click on Finish button.

**New JSP**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

The file Del\_data.jsp already exists.

25) Now create one more JSP file. But File Name will be getData.

**New JSP**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

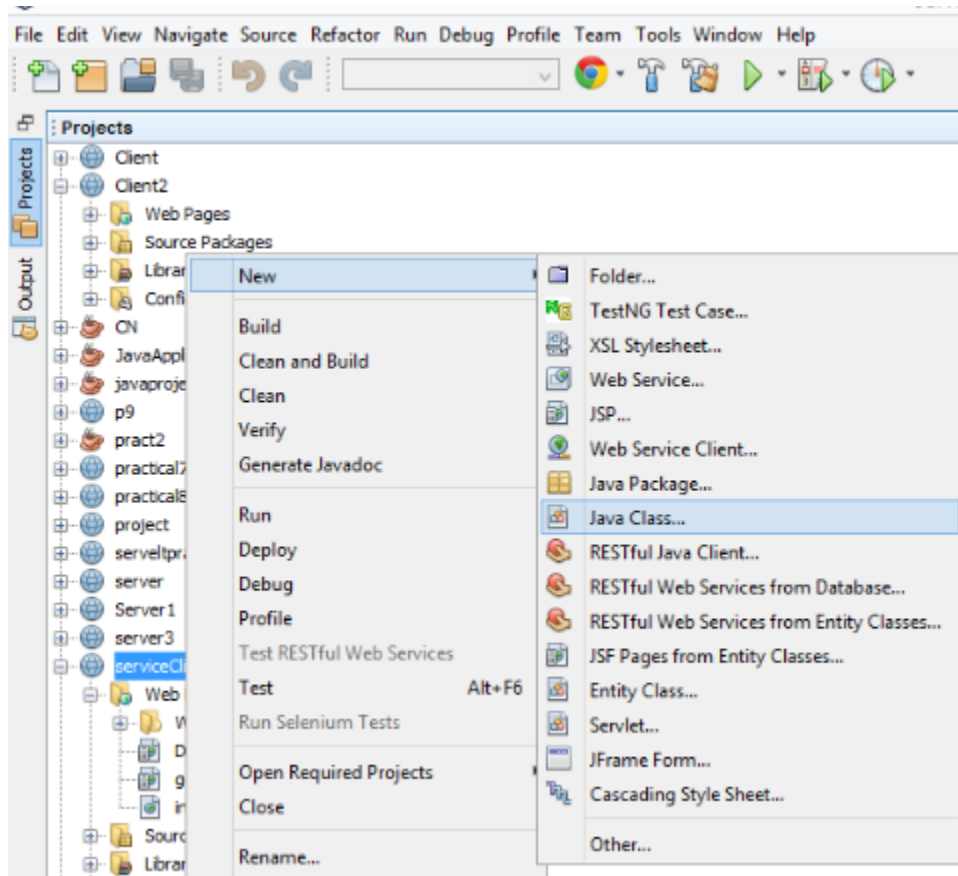
☐ JSP Document (XML Syntax)

Description:

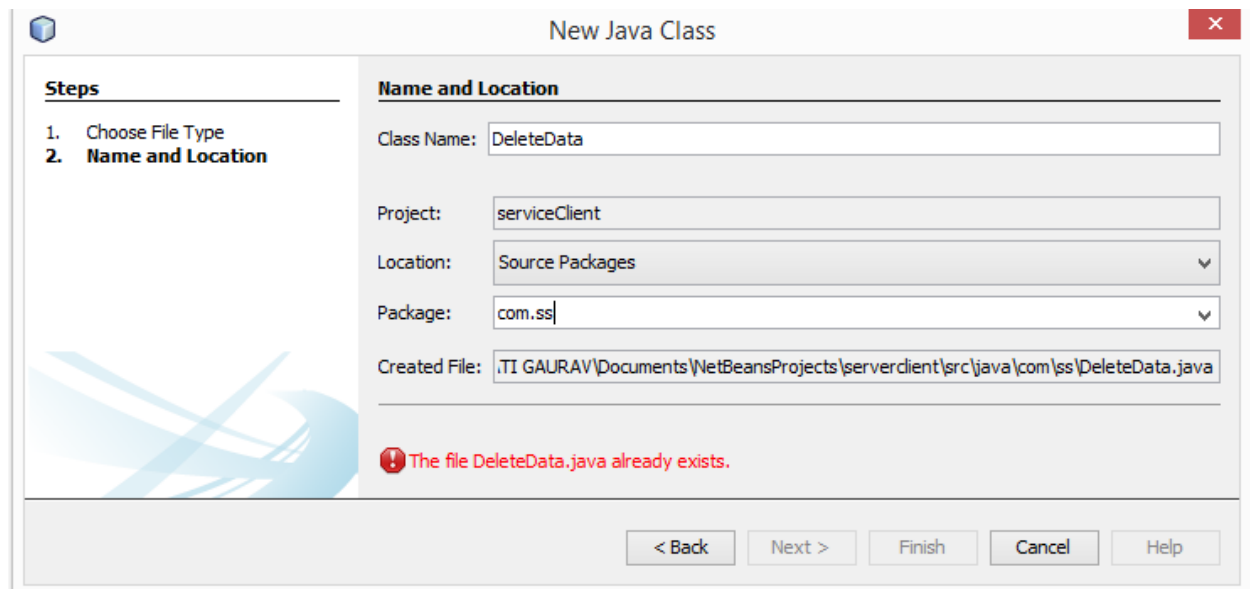
The file getData.jsp already exists.

## Practical:5

26) Now create a Java class. Right click on ServiceClient -> New -> java Class

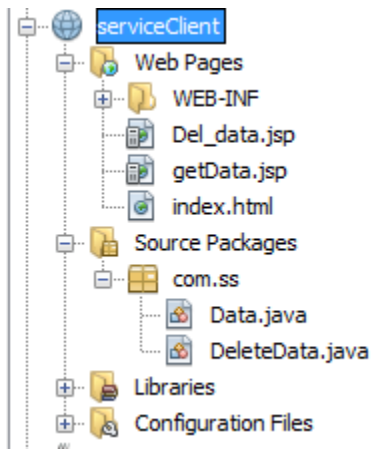


27) Enter Class Name DeleteData and Package com.ss. After that click on Finish.



## Practical:5

28) Your project file structure will look like below.



29) Now open the index.html of ServiceClient project by double click on it and add the following code.

Code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form>
      <h2>One-way Operation</h2><br>
      <input type="text" name="ID" placeholder="Enter ID"><br><br>
      <input type="submit" formaction="Del_data.jsp" value="Delete Data"><br>
      <h1>-----</h1>
      <h2>Request-Response operation</h2><br><br>
      <input type="submit" formaction="getData.jsp" value="Get Data">
    </form>

  </body>
</html>
```

## Practical:5

**30) Now open DeleteData.java file by double click on it and add following code.**

Code:

```
package com.ss;

public class DeleteData {
    public static void deldata(String id){
        String a = id;
        Data ob = new Data();
        ob.remove(a);
        System.out.println("Data is deleted.");
    }
}
```

**31) Now open the Del\_data.jsp file and replace the contents of body with the following code.**

Code:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%@ page import="com.ss.DeleteData"%>
        <%
            String id = request.getParameter("ID");
            DeleteData.deldata(id);
        %>

    </body>
</html>
```

## Practical:5

**32) Now open the getData.jsp file and replace the contents of html tag with the following code and save it.**

Code:

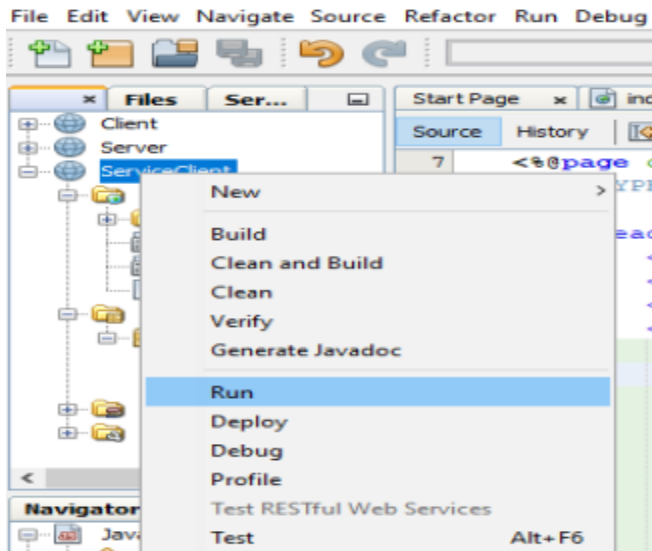
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      table {
        font-family: arial, sans-serif;
        border-collapse: collapse;
      }
      td, th {
        border: 1px solid #000000;
        text-align: center;
        padding: 8px;
      }
    </style>
    <script>
      var request = new XMLHttpRequest();
      request.open('GET',
        'http://localhost:8080/server/webresources/com.kk.Friends/', true);
      request.onload = function () {
        // begin accessing JSON data here
        var data = JSON.parse(this.response);
        for (var i = 0; i < data.length; i++) {
          var table = document.getElementById("myTable");
          var row = table.insertRow();
          var cell1 = row.insertCell(0);
          var cell2 = row.insertCell(1);
          cell1.innerHTML = data[i].id;
          cell2.innerHTML = data[i].firstname;
        }
      };
      request.send();
    </script>
  </head>
  <body>
    <table id="myTable">
      <tr>
```



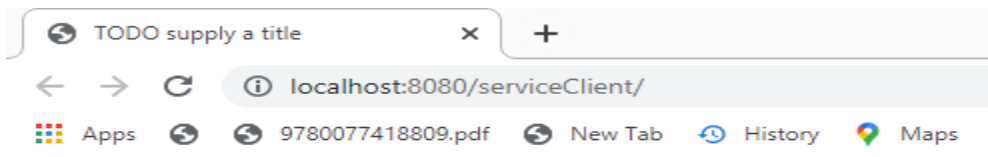
## Practical:5

```
<th> ID</th>
<th>NAME</th>
</tr>
</table>
</body>
</html>
```

### 33) Now Run the ServiceClient project



34) On run the project following window will open in browser.



### One-way Operation

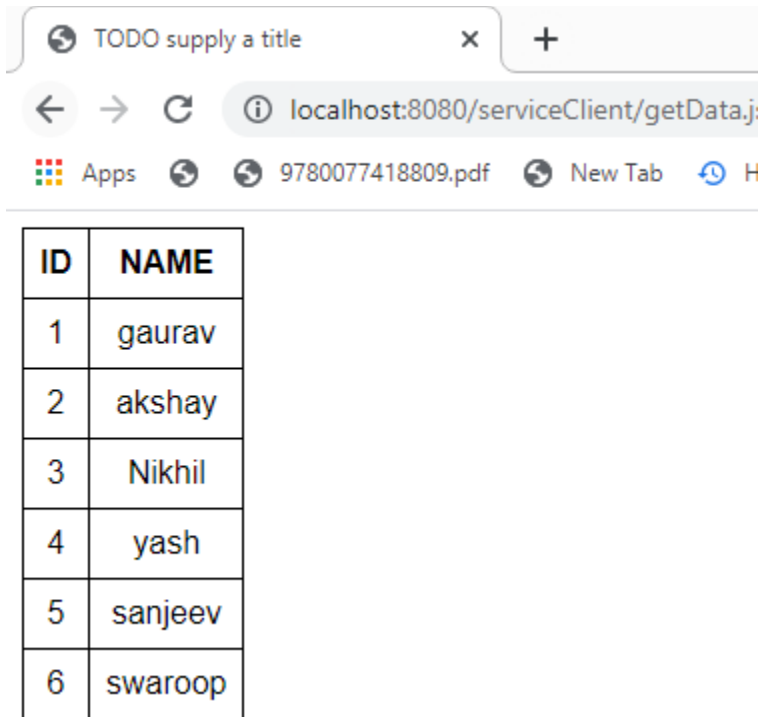
---

### Request-Response operation

## Practical:5

If you will click on Delete Data button, record of entered ID will be deleted. If you will click on Get Data button Request for the data will be send and it will return the data as response. Hence it is two way operation.

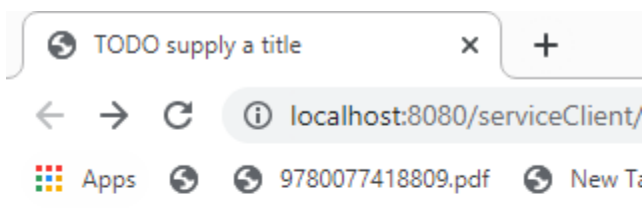
35) By click on Get Data button.



The screenshot shows a web browser window with a single tab titled 'TODO supply a title'. The address bar displays 'localhost:8080/serviceClient/getData.js'. Below the browser window, a table is displayed with two columns: 'ID' and 'NAME'. The table contains six rows of data.

ID	NAME
1	gaurav
2	akshay
3	Nikhil
4	yash
5	sanjeev
6	swaroop

36) Entering ID 6 and clicking on Delete Data button.



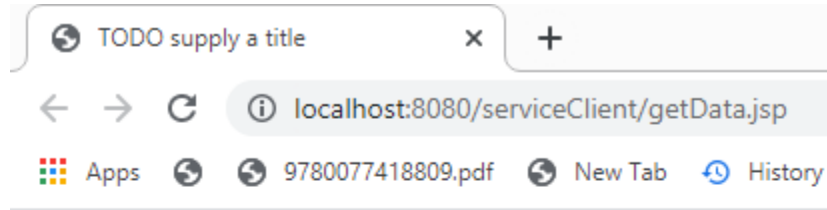
The screenshot shows a web browser window with a single tab titled 'TODO supply a title'. The address bar displays 'localhost:8080/serviceClient/'. Below the browser window, a form is displayed with a text input field containing the number '6' and a button labeled 'Delete Data'.

## One-way Operation

Delete Data

## Practical:5

37) after deleting id 6



ID	NAME
1	gaurav
2	akshay
3	Nikhil
4	yash
5	sanjeev