

Date: 18/11/2020

**Practical no 9**

**AIM:** Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.

**Note:** To do this practical, follow the steps from 1 to 18 present in practical – 7.  
After that do not close or restart the NetBeans.

**Steps :-**

**1) Create an another Web Application project and Give name as Consumer.**

**New Web Application**

**Steps**

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

**Name and Location**

Project Name:

Project Location:

Project Folder:

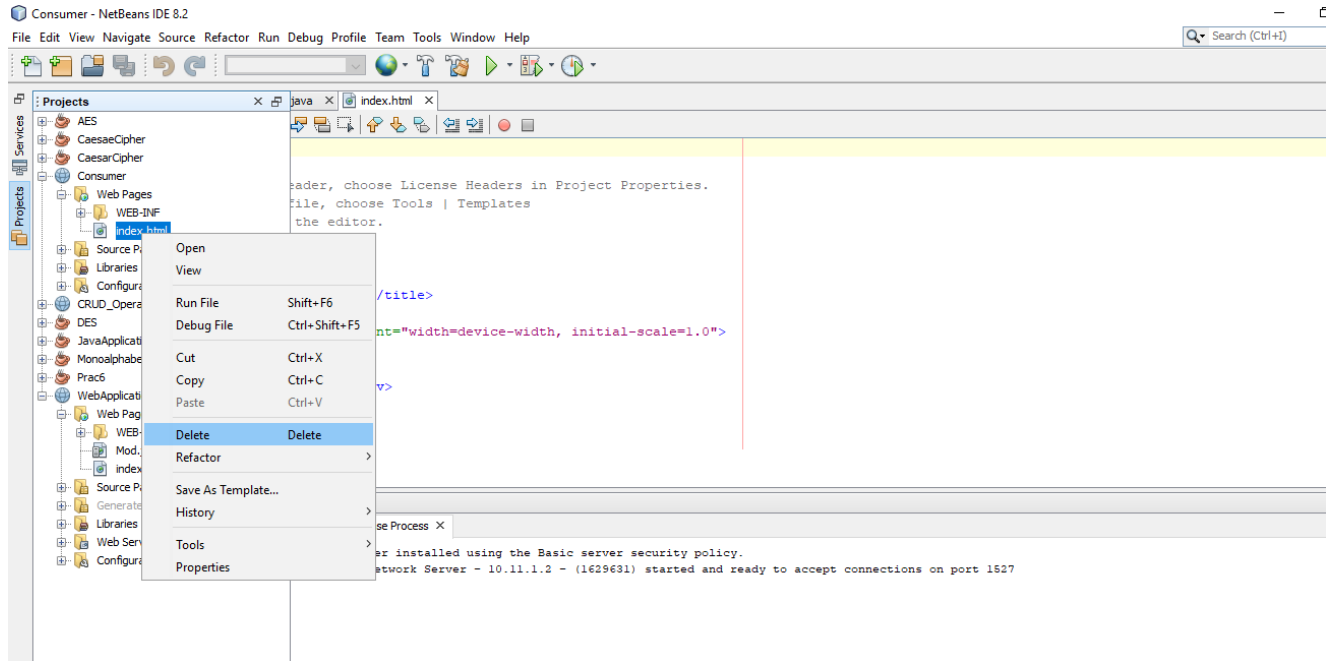
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

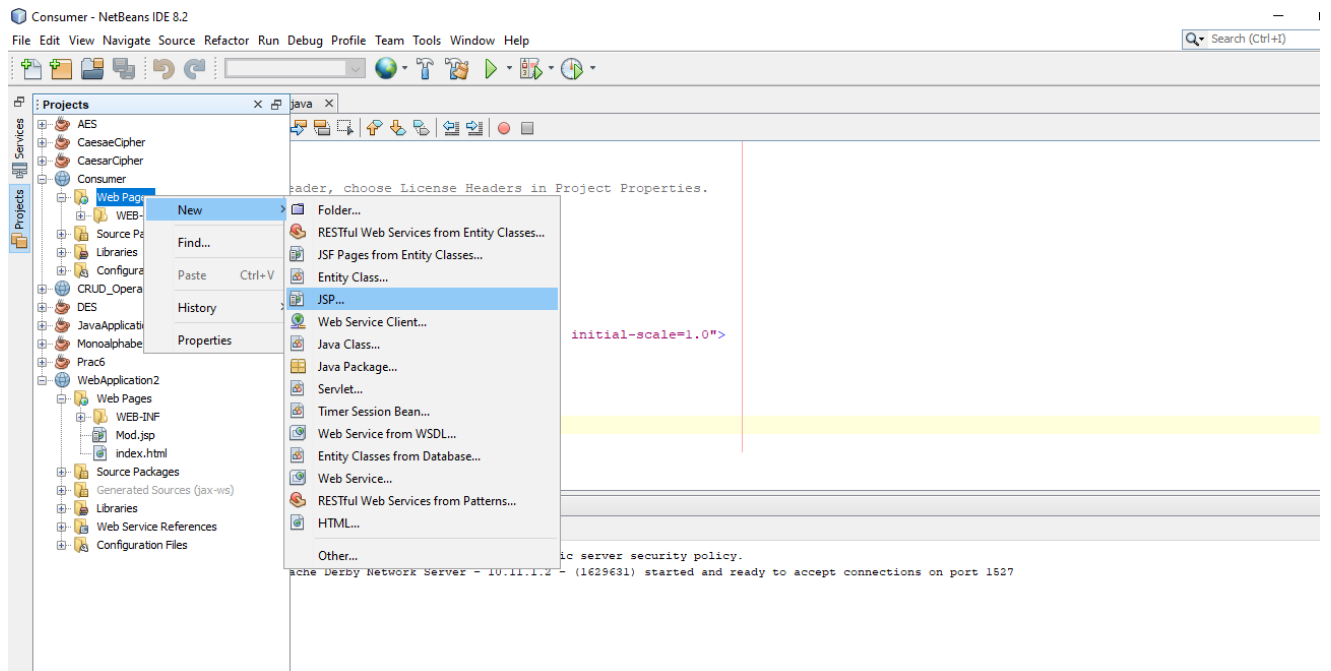
Different users and projects can share the same compilation libraries (see Help for details).

< Back   **Next >**   Finish   Cancel   Help

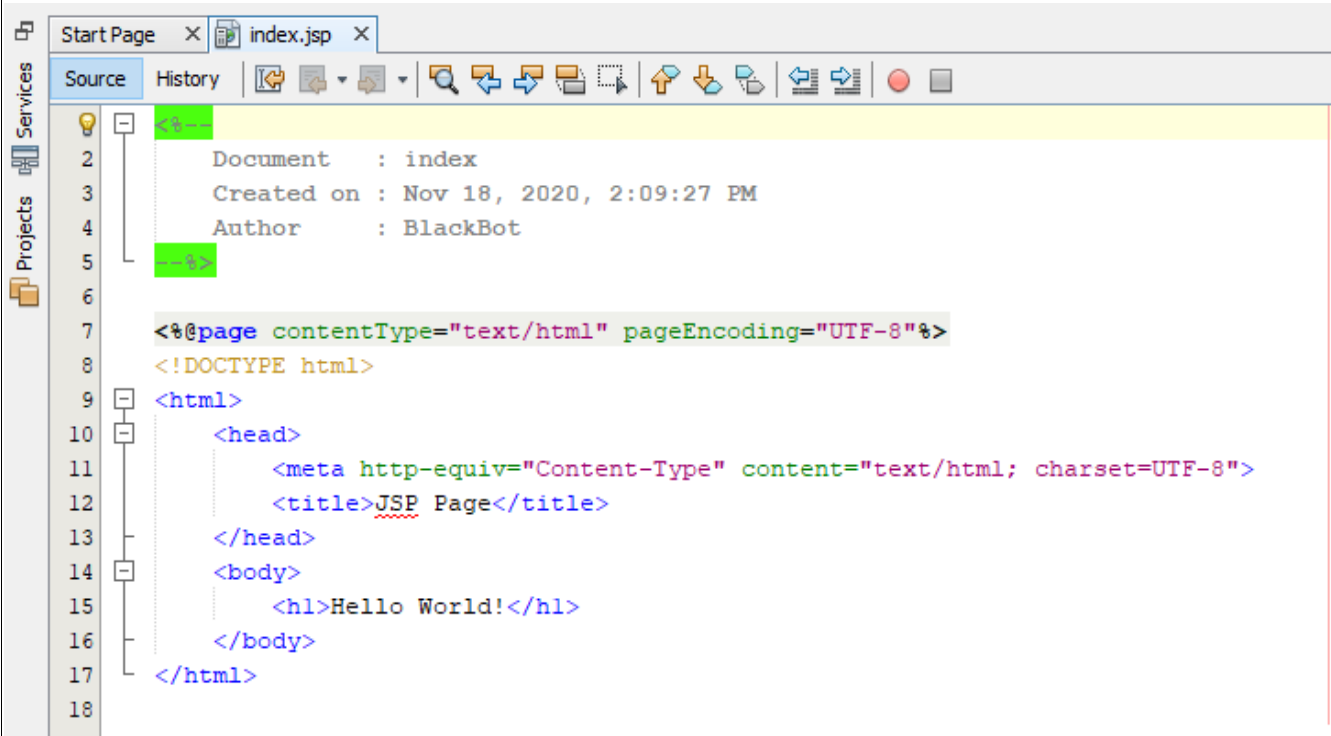
- 2) And create it. Next Finish.
- 3) Now right click on index.html and delete it.



- 4) Now right click on Web Pages and select JSP to add a JSP page.



- 5) Give index name to it and then click on Finish.  
 6) Now index.jsp page will open like below.



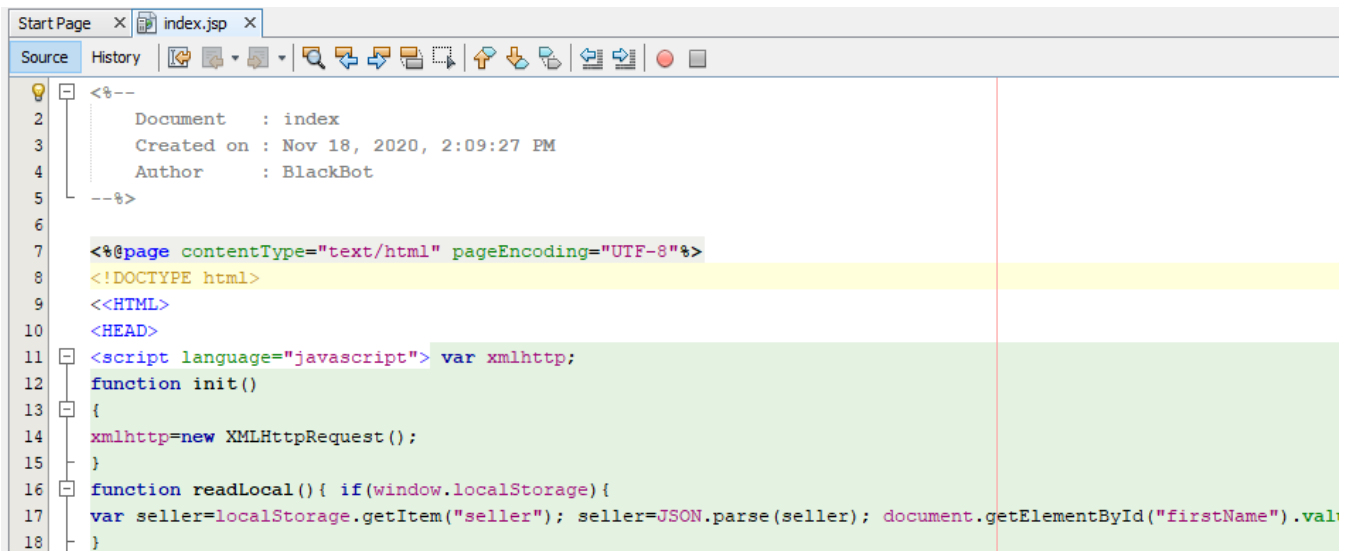
- 7) Now select HTML content of index.jsp file and replace it with following bold letter codes.

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<<HTML>
<HEAD>
<script language="javascript"> var xmlhttp;
function init()
{
xmlhttp=new XMLHttpRequest();
}
function readLocal(){ if(window.localStorage){
var seller=localStorage.getItem("seller");
seller=JSON.parse(seller);
document.getElementById("firstName").value=seller.firstName;
document.getElementById("sellerid").value=seller.id;
}
}
  
```

```
}
}
function saveLocal()
{
var sellerid=document.getElementById("sellerid");
var url="http://localhost:8080/CRUD_Operation/webresources/com.kk.seller/"+
sellerid.value;
xmlhttp.open('GET',url,true); xmlhttp.send(null);
xmlhttp.onreadystatechange =function(){
if(xmlhttp.readyState===4){alert("6"+sellerid);
if(xmlhttp.status===200){alert("7"+sellerid);
var seller =eval("("+xmlhttp.responseText+")");
if(window.localStorage){
localStorage.setItem("seller",JSON.stringify(seller));
alert("information stored successfully"+seller.firstName);
}
else{ alert("notstored");
}}
else
alert("error");
}
}
}
</script>
</head>
<body onLoad = "init()">
<table>
<tr>
<td>Enter id:</td>
<td><input type="text" id="sellerid"/>
<input type="button" value="load employee in local browser"
onClick="saveLocal()"/>
</td>
</tr>
<tr>
<td>read from local</td>
<td><input type="button" value="Send values" onClick="readLocal()"/></td>
</tr>
<tr>
<td>first Name:</td>
```

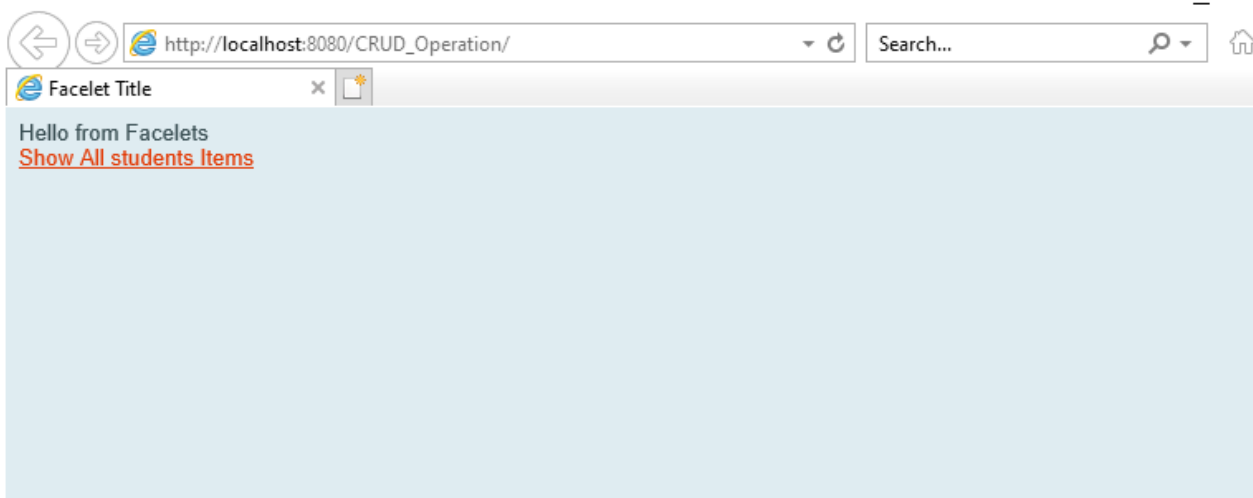
```
<td> <input type="text" id="firstName"/></td>
</tr>
<tr>
    performed by krunal 713
</tr>
</table>
</body>
</html>
```



```
1 <!--
2     Document    : index
3     Created on  : Nov 18, 2020, 2:09:27 PM
4     Author     : BlackBot
5 -->
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <<HTML>
10 <HEAD>
11 <script language="javascript"> var xmlhttp;
12 function init()
13 {
14     xmlhttp=new XMLHttpRequest();
15 }
16 function readLocal(){ if(window.localStorage){
17     var seller=localStorage.getItem("seller"); seller=JSON.parse(seller); document.getElementById("firstName").val
18 }
```

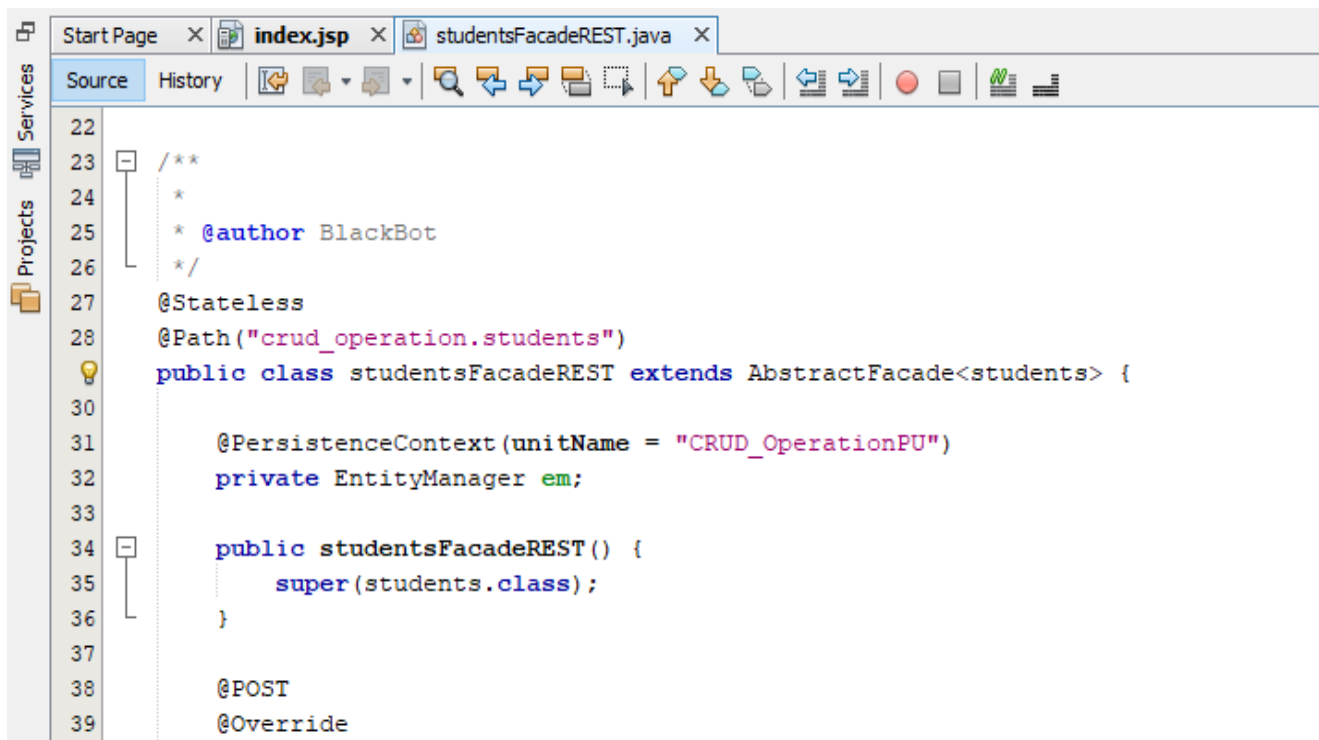
9) [http://localhost:8080/CRUD\\_Operation/webresources/com.kk.seller/](http://localhost:8080/CRUD_Operation/webresources/com.kk.seller/)

Red part is the URL of which is obtained in browser by running of CRUD\_Operation web application.



**Blue part** in above link is static. But red part is dynamic which is based on practical 7. So if you have changed name anywhere then the above URL will change accordingly.

10) Now open studentsFacadeREST.java file by follow below pic available in CRUD\_Operation web application.

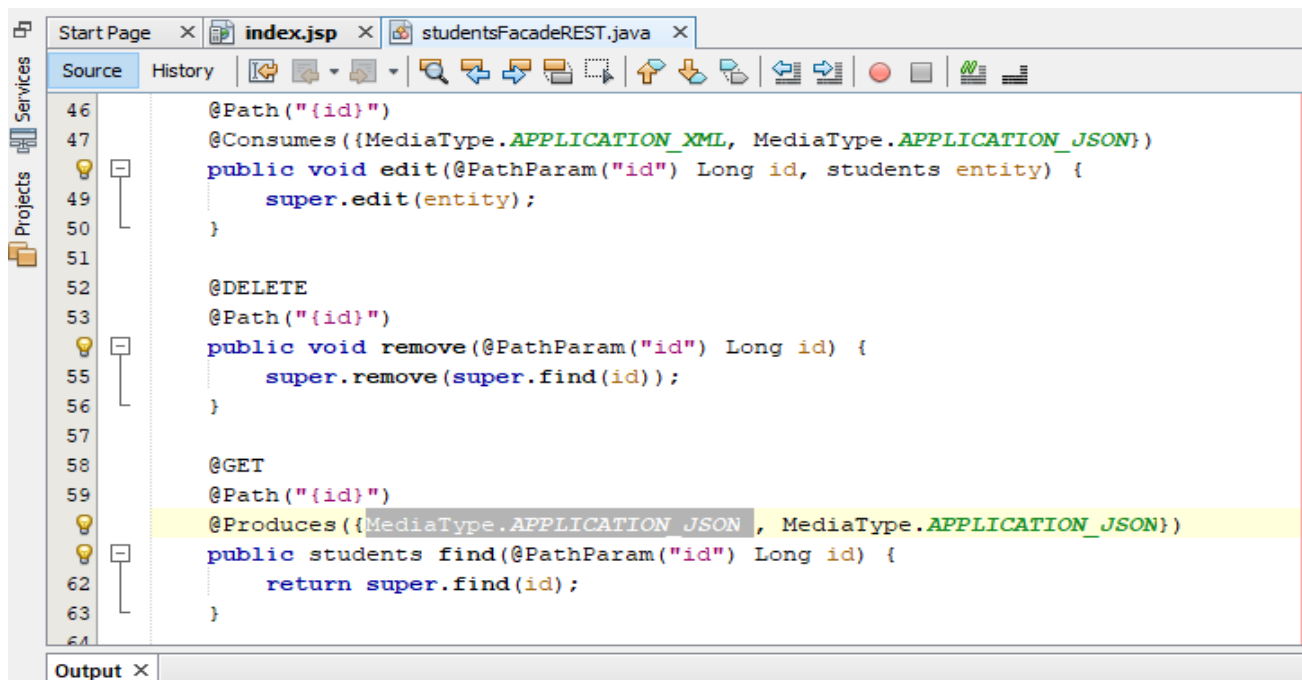


```
22
23 /**
24  *
25  * @author BlackBot
26  */
27 @Stateless
28 @Path("crud_operation.students")
29 public class studentsFacadeREST extends AbstractFacade<students> {
30
31     @PersistenceContext(unitName = "CRUD_OperationPU")
32     private EntityManager em;
33
34     public studentsFacadeREST() {
35         super(students.class);
36     }
37
38     @POST
39     @Override
```

11) Now delete the selected part. Because it will return data in XML format to the consumer.

But we have written javascript in index.jsp for JSON data format only.

After that deploy the **CRUD\_Operation** web application; so that it will update the changes.



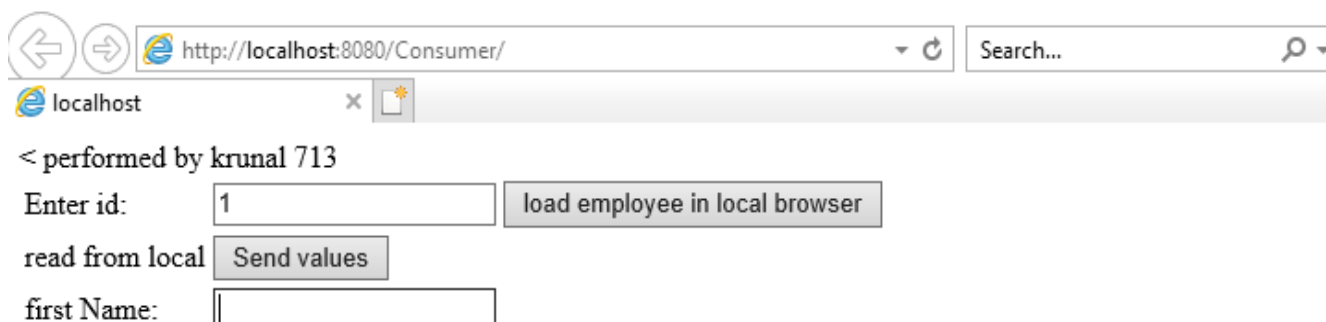
```

46  @Path("{id}")
47  @Consumes({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
48  public void edit(@PathParam("id") Long id, students entity) {
49      super.edit(entity);
50  }
51
52  @DELETE
53  @Path("{id}")
54  public void remove(@PathParam("id") Long id) {
55      super.remove(super.find(id));
56  }
57
58  @GET
59  @Path("{id}")
60  @Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
61  public students find(@PathParam("id") Long id) {
62      return super.find(id);
63  }
64

```

12) Now run the Consumer application. A window will open in browser like below.

13) Now if you will enter an id into Enter id textbox which you have created in database (id from data in last step of practical 7) and then click on load employee in local browser button. It will get the detail of particular entered id and will store it into local storage of browser. Now click on Send Values to get the first name of entered id.



< performed by krunal 713

Enter id:

read from local

first Name: