

Practical No. 16

Objective: Demonstrate the use of Stateful session beans using RMI.

Theory:

Steps for creating a Stateful Session Bean using RMI

1. Create a **Java Class Library – “TutorialInterface”**
2. Create a **Java EE → EJB Module –“MyListBean”**
3. In “**MyListBean**” create a **session bean** --- create package “**com.sess.bean**”→new **sessionbean** “**TestBean.java**”→ **type=Stateful**→ **RemoteInterface**→”**TutorialInterface**”.
4. This will create a remote interface with name “**TestBeanRemote**” under the same package “**com.sess.bean**” in **TutorialInterface**.
5. In the **TestBean** add business methods.
6. **Build the class library project “TutorialInterface”.**
7. In the EJB Module- “**MyListBean**” create a **GlassFishDescriptor file**.
8. Inside the GlassFishDescriptor file select **EJB→TestBean→JNDI Name(Tutorial)**
9. **Deploy the EJB Module.**
10. Create a client application. **New project→add Java webapplication. Eg. EJBwebClient.**
11. In the libraryfolder of **EJBwebClient** →**add Project→TutorialInterface**.
12. Now in the **EJBwebclientint**→ **create a JSP webclient.jsp** 13. Inside this just write a scriptlet to define the **jspInit()**.
14. Inside the **jspInit() initialize the InitialContext object and call the lookup()** with this object.
15. Rest in the html part create a html form and inside that add a text field and two buttons, ADD and REM
16. The value entered in the textbox should be added or rem from the list by clicking on ADD or REM button.
17. The affected list must be shown below.
18. **Deploy this JSP(webclient.jsp). Then Run it.**

Source Code:

```
TutorialInterface (TestBeanRemote.java) package  
com.sess.bean;  
  
import java.util.List;  
import javax.ejb.Remote;  
  
@Remote  
public interface TestBeanRemote {  
    void addElement(int e);  
    void  
    removeElement(int e);  
    List getElement();  
}
```

```
MyListBean (TestBean.java) package  
com.sess.bean;
```

```

import java.util.ArrayList;
import java.util.List; import
javax.ejb.Stateful;

@Stateful
public class TestBean implements TestBeanRemote
    { List<Integer> values=new ArrayList(); @Override
        public void addElement(int e) {
            values.add(e);

        }
        @Override
        public void removeElement(int e) {
            Integer i = new Integer(e);
            values.remove(i);

        }
        @Override
        public List
getElement() {
            return
values;
        }
    }
}

```

EJBWebCleint (webclient.jsp) <%--

```

Document      : webclient
Created on : Feb 21, 2018, 2:19:49 AM
Author       : DEll
--%>

```

```

<%@page import="javax.naming.Context"%>

<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%> <%@page
import="javax.naming.InitialContext"%> <%@page
import="com.sess.bean.TestBeanRemote"%> <%@page
import="com.sess.bean.TestBean"%> <!DOCTYPE html> <%!

```

```

private static TestBeanRemote r;

public void jspInit(){
try{
    InitialContext ic=new InitialContext();
    r =(TestBeanRemote) ic.lookup("Tut");
}

```

```

        }
    catch(Exception e){
        System.out.println("Uninitialised object");
    }
}

%>
<%

    if(request.getParameter("ADD")!=null) { int e =
Integer.parseInt(request.getParameter("t1")); out.println(e);
r.addElement(e);
}

    if(request.getParameter("REM")!=null)
{
    int e = Integer.parseInt(request.getParameter("t1"));
r.removeElement(e);
}

%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

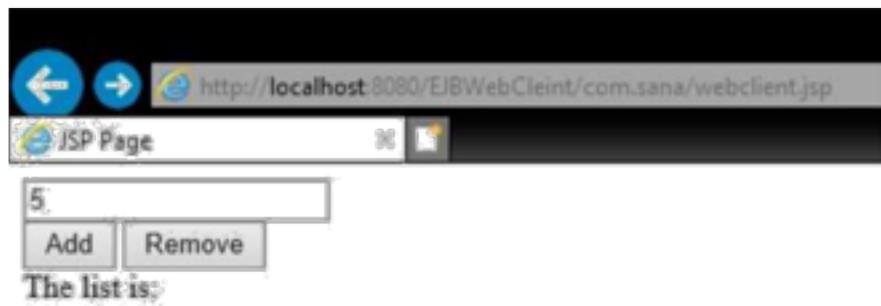
<title>JSP Page</title>
</head>
<body>
<form method="POST" action="webclient.jsp"><input
type="text" name="t1" value="" /><br><input type="submit"
value="Add" name="ADD"/>
<input type="submit" value="Remove" name="REM"/><br>
</form>
The list is:
<% if(r!=null){

    List<Integer> num=r.getElement();
for(int i : num){

    out.println("<br>" + i);
}
%>
</body>
</html>

```

Output:



Result: The given program has been compiled and executed successfully.