#### 1) Create stateless bean component

To create the stateless bean component, you need to create a remote interface and a bean class.

*File: AdderImplRemote.java*

1. **package** com.javatpoint;
2. **import** javax.ejb.Remote;
4. @Remote
5. **public** **interface** AdderImplRemote {
6. **int** add(**int** a,**int** b);
7. }

*File: AdderImpl.java*

1. **package** com.javatpoint;
2. **import** javax.ejb.Stateless;
4. @Stateless(mappedName="st1")
5. **public** **class** AdderImpl **implements** AdderImplRemote {
6. **public** **int** add(**int** a,**int** b){
7. **return** a+b;
8. }
9. }

#### 2) Create stateless bean client

The stateless bean client may be local, remote or webservice client. Here, we are going to create remote client. It is console based application. Here, we are not using dependency injection. The dependency injection can be used with web based client only.

*File: AdderImpl.java*

1. **package** com.javatpoint;
2. **import** javax.naming.Context;
3. **import** javax.naming.InitialContext;
5. **public** **class** Test {
6. **public** **static** **void** main(String[] args)**throws** Exception {
7. Context context=**new** InitialContext();
8. AdderImplRemote remote=(AdderImplRemote)context.lookup("st1");
9. System.out.println(remote.add(32,32));
10. }
11. }

#### Output

Output: 64

#### Create stateful bean component

Let's create a remote interface and a bean class for developing stateful bean component.

*File: BankRemote.java*

1. **package** com.javatpoint;
2. **import** javax.ejb.Remote;
3. @Remote
4. **public** **interface** BankRemote {
5. **boolean** withdraw(**int** amount);
6. **void** deposit(**int** amount);
7. **int** getBalance();
8. }

*File: Bank.java*

1. **package** com.javatpoint;
2. **import** javax.ejb.Stateful;
3. @Stateful(mappedName = "stateful123")
4. **public** **class** Bank **implements** BankRemote {
5. **private** **int** amount=0;
6. **public** **boolean** withdraw(**int** amount){
7. **if**(amount<=**this**.amount){
8. **this**.amount-=amount;
9. **return** **true**;
10. }**else**{
11. **return** **false**;
12. }
13. }
14. **public** **void** deposit(**int** amount){
15. **this**.amount+=amount;
16. }
17. **public** **int** getBalance(){
18. **return** amount;
19. }
20. }

#### 2) Create stateful bean client

The stateful bean client may be local, remote or webservice client. Here, we are going to create web based client and not using dependency injection.

*File: index.jsp*

1. **<a** href="OpenAccount"**>**Open Account**</a>**

*File: operation.jsp*

1. **<form** action="operationprocess.jsp"**>**
2. Enter Amount:**<input** type="text" name="amount"**/><br>**
4. Choose Operation:
5. Deposit**<input** type="radio" name="operation" value="deposit"**/>**
6. Withdraw**<input** type="radio" name="operation" value="withdraw"**/>**
7. Check balance**<input** type="radio" name="operation" value="checkbalance"**/>**
8. **<br>**
9. **<input** type="submit" value="submit"**>**
10. **</form>**

*File: operationprocess.jsp*

1. **<**%@ page import="com.javatpoint.\*" %**>**
2. **<**%
3. BankRemote remote=(BankRemote)session.getAttribute("remote");
4. String operation=request.getParameter("operation");
5. String amount=request.getParameter("amount");
7. if(operation!=null){
9. if(operation.equals("deposit")){
10. remote.deposit(Integer.parseInt(amount));
11. out.print("Amount successfully deposited!");
12. }else
13. if(operation.equals("withdraw")){
14. boolean status=remote.withdraw(Integer.parseInt(amount));
15. if(status){
16. out.print("Amount successfully withdrawn!");
17. }else{
18. out.println("Enter less amount");
19. }
20. }else{
21. out.println("Current Amount: "+remote.getBalance());
22. }
23. }
24. %**>**
25. **<hr/>**
26. **<jsp:include** page="operation.jsp"**></jsp:include>**

*File: OpenAccount.java*

1. **package** com.javatpoint;
2. **import** java.io.IOException;
3. **import** javax.ejb.EJB;
4. **import** javax.naming.InitialContext;
5. **import** javax.servlet.ServletException;
6. **import** javax.servlet.annotation.WebServlet;
7. **import** javax.servlet.http.HttpServlet;
8. **import** javax.servlet.http.HttpServletRequest;
9. **import** javax.servlet.http.HttpServletResponse;
10. @WebServlet("/OpenAccount")
11. **public** **class** OpenAccount **extends** HttpServlet {
12. //@EJB(mappedName="stateful123")
13. //BankRemote b;
14. **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)
15. **throws** ServletException, IOException {
16. **try**{
17. InitialContext context=**new** InitialContext();
18. BankRemote b=(BankRemote)context.lookup("stateful123");
20. request.getSession().setAttribute("remote",b);
21. request.getRequestDispatcher("/operation.jsp").forward(request, response);
23. }**catch**(Exception e){System.out.println(e);}
24. }
25. }