**Registration Form in JSP**

1. [Registration Form in JSP](http://www.javatpoint.com/registration-form-in-jsp)
2. [Example of Registration Form in JSP](http://www.javatpoint.com/registration-form-in-jsp#jspregisex)

For creating registration form, you must have a table in the database. You can write the database logic in JSP file, but separating it from the JSP page is better approach. Here, we are going to use DAO, Factory Method, DTO and Singletion design patterns. There are many files:

* **index.jsp** for getting the values from the user
* **User.java**, a bean class that have properties and setter and getter methods.
* **process.jsp**, a jsp file that processes the request and calls the methods
* **Provider.java**, an interface that contains many constants like DRIVER\_CLASS, CONNECTION\_URL, USERNAME and PASSWORD
* **ConnectionProvider.java**, a class that returns an object of Connection. It uses the Singleton and factory method design pattern.
* **RegisterDao.java**, a DAO class that is responsible to get access to the database

**Example of Registration Form in JSP**

|  |
| --- |
| In this example, we are using the Oracle10g database to connect with the database. Let's first create the table in the Oracle database: |

1. CREATE TABLE  "USER432"
2. (    "NAME" VARCHAR2(4000),
3. "EMAIL" VARCHAR2(4000),
4. "PASS" VARCHAR2(4000)
5. )
6. /

We have created the table named user432 here.

**index.jsp**

We are having only three fields here, to make the concept clear and simplify the flow of the application. You can have other fields also like country, hobby etc. according to your requirement.

1. <form action="process.jsp">
2. <input type="text" name="uname" value="Name..." onclick="this.value=''"/><br/>
3. <input type="text" name="uemail"  value="Email ID..." onclick="this.value=''"/><br/>
4. <input type="password" name="upass"  value="Password..." onclick="this.value=''"/><br/>
5. <input type="submit" value="register"/>
6. </form>

**process.jsp**

This jsp file contains all the incoming values to an object of bean class which is passed as an argument in the register method of the RegisterDao class.

1. <%@page import="bean.RegisterDao"%>
2. <jsp:useBean id="obj" class="bean.User"/>
4. <jsp:setProperty property="\*" name="obj"/>
6. <%
7. int status=RegisterDao.register(obj);
8. if(status>0)
9. out.print("You are successfully registered");
11. %>

**User.java**

|  |
| --- |
| It is the bean class that have 3 properties uname, uemail and upass with its setter and getter methods. |

1. package bean;
3. public class User {
4. private String uname,upass,uemail;
6. public String getUname() {
7. return uname;
8. }
10. public void setUname(String uname) {
11. this.uname = uname;
12. }
14. public String getUpass() {
15. return upass;
16. }
18. public void setUpass(String upass) {
19. this.upass = upass;
20. }
22. public String getUemail() {
23. return uemail;
24. }
26. public void setUemail(String uemail) {
27. this.uemail = uemail;
28. }
30. }

**Provider.java**

This interface contains four constants that can vary from database to database.

1. package bean;
3. public interface Provider {
4. String DRIVER="oracle.jdbc.driver.OracleDriver";
5. String CONNECTION\_URL="jdbc:oracle:thin:@localhost:1521:xe";
6. String USERNAME="system";
7. String PASSWORD="oracle";
9. }

**ConnectionProvider.java**

This class is responsible to return the object of Connection. Here, driver class is loaded only once and connection object gets memory only once.

1. package bean;
2. import java.sql.\*;
3. import static bean.Provider.\*;
5. public class ConnectionProvider {
6. private static Connection con=null;
7. static{
8. try{
9. Class.forName(DRIVER);
10. con=DriverManager.getConnection(CONNECTION\_URL,USERNAME,PASSWORD);
11. }catch(Exception e){}
12. }
14. public static Connection getCon(){
15. return con;
16. }
18. }

**RegisterDao.java**

This class inserts the values of the bean component into the database.

1. package bean;
3. import java.sql.\*;
5. public class RegisterDao {
7. public static int register(User u){
8. int status=0;
9. try{
10. Connection con=ConnectionProvider.getCon();
11. PreparedStatement ps=con.prepareStatement("insert into user432 values(?,?,?)");
12. ps.setString(1,u.getUname());
13. ps.setString(2,u.getUemail());
14. ps.setString(3,u.getUpass());
16. status=ps.executeUpdate();
17. }catch(Exception e){}
19. return status;
20. }
22. }