

jsp:Param Example

A Scenario: Consider a scenario on successful login the user needs to be forwarded to the success page with the user name set as an parameter to the request object so that name can be accessed in the home page and displayed.

Jsp:Param can be used along with jsp:Forward as shown

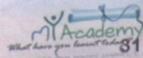
```
<jsp:forward page =“success.jsp” />
    <jsp:param name =“userName” value =“<name>”/>
</jsp:forward>
```

Where

success.jsp : success page

userName : Request attribute Name

name : The variable containing the user name



jsp:useBean

The `<jsp:useBean>` tag attempts to locates a bean or if the Bean does not exist, instantiates it from the class specified.

Syntax: `<jsp:useBean id="name" class="package.class"
scope="request/session/page/application"/>`

Where ,

Id – The name used for referring for the bean object

Class – The Bean Class

Scope - The scope in which the bean object is available



Other Attributes for useBean action tag

Attribute	Description
id	Gives a name to the variable that will reference the bean. A previous bean object is used instead of instantiating a new one if one can be found with the same id and scope.
class	Instantiates a Bean from a class, using the new keyword and the class constructor. The class must not be abstract and must have a public, no-argument constructor. The package and class name are case sensitive.
scope	scope="page request session application" Defines a scope in which the bean exists .The default value is page.
type	If the Bean already exists in the scope, gives the Bean a data type other than the class from which it was instantiated. If you use type without class or beanName, no Bean is instantiated. The package and class name are case sensitive.
beanName	Gives the name of the bean, as you would supply it to the instantiate method of Beans.

How `jsp:useBean` works?

Example : `<jsp:useBean id="user" class="com.catp.beans.UserBean" scope="request" />`

How it works? Sequence of steps,

1. Attempts to locate a Bean with the name "UserBean" in the request scope.
2. If it finds the Bean, stores a reference in the variable **user**.
3. If it does not find the Bean, instantiates a bean using the class **UserBean**, and stores the reference to the variable **user**.

Bean Object Scopes

- **page** : is available only within the JSP page and is destroyed when the page has finished generating its output for the request.
- **request** : valid for the current request and is destroyed when the response is sent
- **session** : valid for a user session and is destroyed when the session is destroyed
- **application** : valid throughout the application and is destroyed when the web application is destroyed/uninstalled.

jsp:setProperty

The **setProperty** action sets the properties of a Bean. The Bean must have been previously defined before this action.

Syntax:

```
<jsp:useBean id="myName" class="package.class" />
<jsp:setProperty name="myName" property="someProperty"
                 value ="someValue" />
```

Where,

name : the name of the bean object and should be the same as the id value specified in usebean

property : the bean property (field name) for which the value is to be set. There should be a instance variable with the property name specified and accessors/mutator methods.

the property

jsp:setProperty mapping HTML forms

The following options can be used to automatically set the values from an HTML form to a java bean using the `setProperty` action.

Option 1

```
<jsp:setProperty name="UserBean" property="someProperty" param="userName" />
```

Sets the value of the HTML form element with the name `userName` into the bean `UserBean`'s property `someProperty`.

Option 2

```
<jsp:setProperty name="myName" property="*" />
```

Sets all the values of the form elements into the bean properties.

In this case the *name of the property* and the *name of the form element* should be the same.

HTML form

```
<form method="post">
<table>
  <tr>
    <td>User Name :</td>
    <td><input type="text" name="userName" /></td>
  </tr>
  <tr>
    <td>Password</td>
    <td><input type="password" name="password" /></td>
  </tr>
  <tr>
    <td colspan="2"><input type="submit" name="login" value="Login" />
  </td>
  </tr>
</table>
</form>
```

UserBean

```
public class User {
  private String userName;
  private String password;
  public String getUserName() {
    return userName;
  }
  public void setUserName(String userName) {
    this.userName = userName;
  }
  public String getPassword() {
    return password;
  }
  public void setPassword(String password) {
    this.password = password;
  }
}
```

UserBean

```
public class User {  
    private String userName;  
    private String password;  
    public String getUserName() {  
        return userName;  
    }  
    public void setUserName(String userName) {  
        this.userName = userName;  
    }  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword(String password) {  
        this.password = password;  
    }  
}
```

Example for jsp.setProperty usage

Option 1:

```
<jsp:useBean id="user" class="com.catp.beans.User"/>
<jsp:setProperty name="user" property="userName" param="userName"/>
<jsp:setProperty name="user" property="password" param="password"/>
```

Option 2:

```
<jsp:useBean id="user" class="User"/>
<jsp:setProperty name="user" property="*"/>
```

Example for jsp.setProperty usage

Option 1:

```
<jsp:useBean id="user" class="com.catp.beans.User"/>
<jsp:setProperty name="user" property="userName" param="userName"/>
<jsp:setProperty name="user" property="password" param="password"/>
```

Option 2:

```
<jsp:useBean id="user" class="User"/>
<jsp:setProperty name="user" property="*"/>
```



jsp:getProperty

The **getProperty** action is used to retrieve the value of a given property and converts it to a string, and finally inserts it into the output.

Syntax:

```
<jsp:useBean id="myName" type="package.class" />
<jsp:getProperty name="myName" property="someProperty" />
```

Where,

name : Bean name same as the id specified in the usebean action

property : The bean property name whose value is to be retrieved

jsp:getProperty

The **getProperty** action is used to retrieve the value of a given property and converts it to a string, and finally inserts it into the output.

Syntax:

```
<jsp:useBean id="myName" type="package.class" />  
<jsp:getProperty name="myName" property="someProperty" />
```

Where,

name : Bean name same as the id specified in the usebean action

property : The bean property name whose value is to be retrieved

jsp:getProperty Example

Example:

```
<jsp:useBean id="user" type="com.catp.beans.UserBean" />
<jsp:getProperty name="user" property="userName" />
```

This Reads the value of the property named **userName** from the **UserBean** and prints it

Recap JSP Action tags

JSP Action tags	Description
<code>jsp:include</code>	For including a page dynamically in a parent page.
<code>jsp:forward</code>	For forwarding the request to another resource.
<code>jsp:usebean</code>	To reference a bean and use its properties.
<code>jsp:setProperty</code>	Set the value of a property referred using use bean.
<code>jsp:getProperty</code>	Gets the property value of a bean referred in use bean tag.
<code>jsp:Param</code>	Used to share parameter and its value with the page being included or forwarded.

Recap JSP Action tags

JSP Action tags	Description
<code>jsp:include</code>	For including a page dynamically in a parent page.
<code>jsp:forward</code>	For forwarding the request to another resource.
<code>jsp:usebean</code>	To reference a bean and use its properties.
<code>jsp:setProperty</code>	Set the value of a property referred using use bean.
<code>jsp:getProperty</code>	Gets the property value of a bean referred in use bean tag.
<code>jsp:Param</code>	Used to share parameter and its value with the page being included or forwarded.



Time To Reflect



Associates to quickly summarize the following before ending the session

- What is a java bean?
- What is the difference between action and directive include?
- What is the action tag used for setting a beans property value?
- What is the action tag used for forwarding the request?
- What is the action tag used for including a page?
- What is the action tag used for mapping HTML form elements with the bean.?



Lend a Hand : Action Tags

Using this demo the associates gets familiarized with the following action tags in JSP

- `jsp:include`
- `jsp:param`
- `jsp:useBean`
- `jsp:setProperty`
- `jsp:getProperty`
- `jsp:forward`

Lend a Hand - Requirement

Consider a scenario in which ABC Soft corp have approached you to create a registration form for their Employees for maintaining profile of each employee in the company.

The requirement is as slated below

- There are three designation of employees
 - a)Manager
 - b)Developer
 - c)Trainee
- Employees should enter details in the registration form
- On successful entry the employee is forwarded to their respective home pages.

Lend Hand- Requirement (Cont)

- The heading of the home page will be based on the designation of the employee.
- The heading should be dynamically loaded based on designation of employee.
- A welcome message should be present in the home page.

Components

1. **Registration.jsp** : Registration page used by employees to register.
2. **Success.jsp** : Common Home page for all employees.
3. **traineesHeading.jsp** : Heading page for trainee's success page.
4. **developersHeading.jsp** : Heading page for developer's success page.
5. **managersHeading.jsp** : Heading page for manager's success page.
6. **Employee.java** : The java bean class for storing employee details.

Lend a Hand : Registration page Design

Create registration page as shown below

Registration Page Windows Internet Explorer
http://localhost:8080/ActionDemo/registration.jsp

Enter Details

First Name []
Last Name []
Password []
User Name []

Gender Male Female

Hobbies Dance Music
 Sports Painting

Designation ...select...

Register

Should have three values
Manager, Trainees,
Developer.

The screenshot shows a Java web application interface. At the top, there is a decorative header with puzzle pieces and the text "Java in Hand - Registration.jsp". Below this is the JSP code for the registration page.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Registration Page</title>
</head>
<jsp:useBean id="userBean" class="com.catp.beans.Employee" scope="request" />
<jsp:setProperty name="userBean" property="*" />
<% if (request.getParameter("register") != null) { %>
<jsp:forward page="success.jsp">
    <jsp:param value="Registration Sucessful" name="message" /></jsp:forward>
<% } %>
<body>
<h2 style="font-size: 26px;">Enter Details</h2>
<form method="post">
<table>
    <tr><td>First Name</td><td><input type="text" name="fName" /></td>
    </tr>
    <tr><td>Last Name</td> <td><input type="text" name="lName" /></td>
    </tr>
    <tr><td>Password</td><td><input type="password" name="password" /></td>
    </tr>
    <tr><td>User Name</td><td><input type="text" name="uName" /></td>
    </tr>
    <tr><td>Gender</td>
        <td><input type="radio" name="gender" value="Male" />Male <br />
        <input type="radio" name="gender" value="Female" />Female</td>
    </tr>
</table>
</form>

```

Annotations:

- A callout points from the `<jsp:useBean ...>` and `<jsp:setProperty ...>` lines to the text: "Instantiates bean using action and sets the value `setProperty` action."
- A callout points from the `<jsp:forward ...>` and `<jsp:param ...>` lines to the text: "IMPORTANT: Ensure the form elements are set to the bean property names. case sensitive."
- A callout points from the `<input type="radio" ...>` lines to the text: "Use `JSP:forward` to forward request to success.jsp page and `jsp:param`"

```

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Registration Page</title>
<head>
<jsp:useBean id="userBean" class="com.jspbeans.Employee" scope="request" />
<% if (request.getParameter("designation") == null) { %>
<jsp:forward page="success.jsp">
<jsp:param value="Registration Successful" name="message" /></jsp:forward>
<% } %>

```

<body>

<h2 style="font-size: 26px;">Enter Details</h2>

<form method="post">

First Name	<input name="fName" type="text"/> </td>
Last Name	<input name="lName" type="text"/> </td>
Password	<input name="password" type="password"/> </td>
User Name	<input name="uName" type="text"/> </td>
Gender	<input name="gender" type="radio" value="Male"/> >Male <input name="gender" type="radio" value="Female"/> >Female</td>
Hobbies	<input name="hobbies" type="checkbox" value="dance"/> > Dance <input name="hobbies" type="checkbox" value="music"/> > Music <input name="hobbies" type="checkbox" value="sports"/> > Sports <input name="hobbies" type="checkbox" value="painting"/> > Painting</td>
Designation	<select name="designation"> <option selected="selected"-->--select--</option> <option value="1">Manager</option><option value="2">Trainee</option> </select>
<td colspan="2"><input type="submit" value="Register" name="register" /></td></td></tr></table></form>	

Instantiates bean using **useBean** action and sets the value using **setProperty** action.

IMPORTANT: Ensure the names of the form elements are same as the bean property names. They are case sensitive.

Use **JSP:forward** to forward request to success.jsp and **JSP:param** to pass message.

```

<%-- If (request.getParameter("message") != null) { --%>
    <bean:Employee> scope="request" />
<jsp:forward page="success.jsp">
    <jsp:param value="Registration Successful" name="message" /></jsp:forward>
}

<body>
<h2 style="font-size: 26px;">Enter Details</h2>
<form method="post">
<table>
    <tr><td>First Name</td><td><input type="text" name="fName" /></td>
    </tr>
    <tr><td>Last Name</td> <td><input type="text" name="lName" /></td>
    </tr>
    <tr><td>Password</td><td><input type="password" name="password" /></td>
    </tr>
    <tr><td>User Name</td><td><input type="text" name="uName" /></td>
    </tr>
    <tr><td>Gender</td>
        <td><input type="radio" name="gender" value="Male" />Male <br />
            <input type="radio" name="gender" value="Female" />Female</td>
    </tr>
    <tr><td>Hobbies</td>
        <td><input type="checkbox" name="hobbies" value="dance" /> Dance
            <input type="checkbox" name="hobbies" value="music" /> Music <br />
            <input type="checkbox" name="hobbies" value="sports" /> Sports
            <input type="checkbox" name="hobbies" value="painting" /> Painting</td>
    </tr>
    <tr>
        <td>Designation</td><td><select name="designation">
            <option selected="selected">--select--</option>
            <option value="1">Manager</option><option value="2">Trainee</option>
            <option value="3">Analyst</option>
        </select></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="Register" name="register" /></td>
    </tr>
</table></form>
</body>
</html>

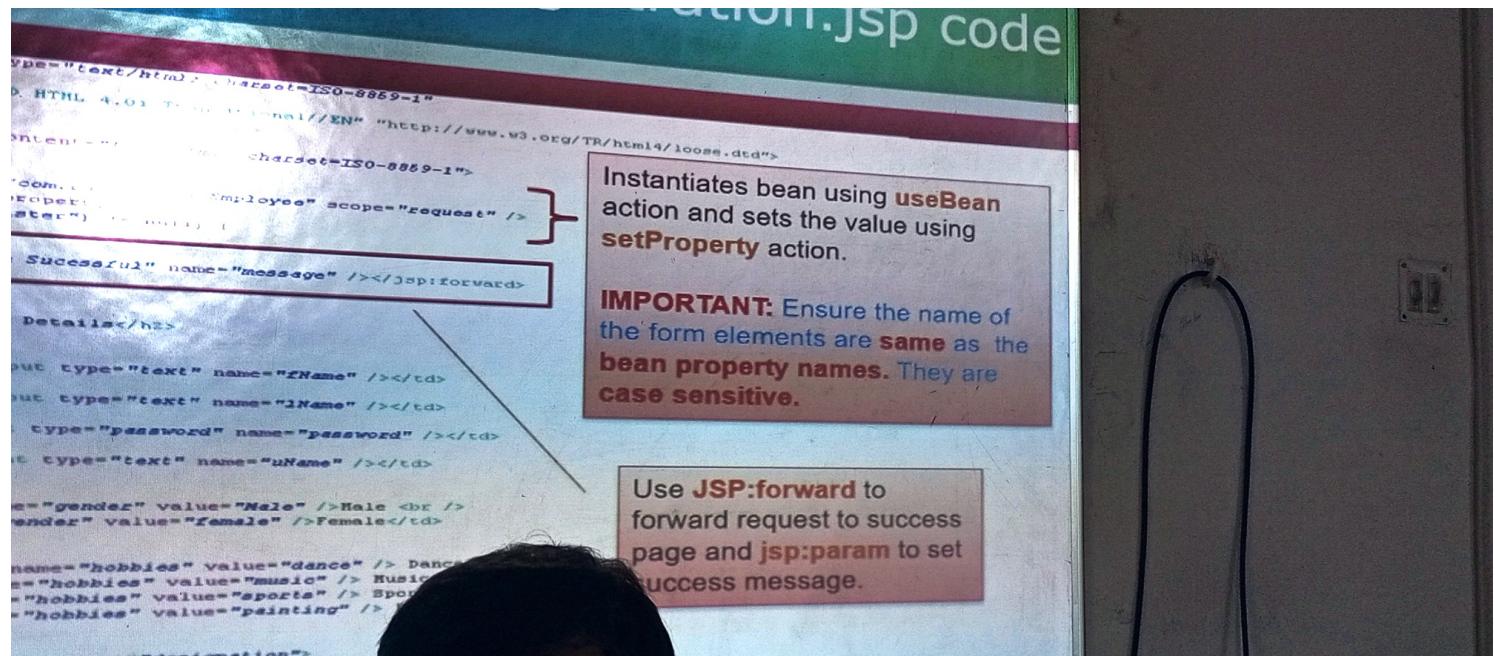
```

| © 2011 Cognizant

action and sets the value of the **setProperty** action.

IMPORTANT: Ensure that the form elements are **set** to the bean property names. Case sensitive.

Use **JSP:forward** to forward request to success.jsp and set the **param** message.





Lend a Han

```
public class Employee {  
    private String fName;  
    private String iName;  
    private String uName;  
    private String password;  
    private String hobbies[];  
    private String designation;  
    private String gender;  
    public String getName() {  
        return fName;  
    }  
    public void setfName(String fName) {  
        this.fName = fName;  
    }  
    public String getiName() {  
        return iName;  
    }  
    public void setiName(String iName) {  
        this.iName = iName;  
    }  
    public String getuName() {  
        return uName;  
    }  
    public void setuName(String uName) {  
        this.uName = uName;  
    }  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword(String password) {  
        this.password = password;  
    }  
    public String[] GetHobbies() {  
        return hobbies;  
    }  
    public void setHobbies(String[] hobbies) {  
        this.hobbies = hobbies;  
    }  
    public String getCountry() {  
        return designation;  
    }  
    public void setCountry(String country) {  
        this.designation = country;  
    }  
    public void setGender(String gender) {  
        this.gender = gender;  
    }  
    public String getGender() {  
        return gender;  
    }  
}
```

| © 2011 Cognizant



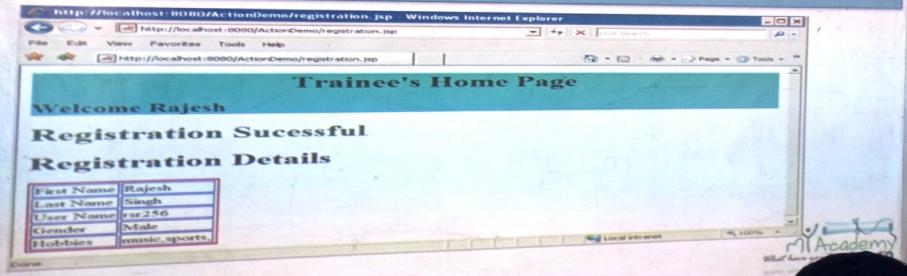
Lend a Hand

```
public class Employee {
    private String fName;
    private String lName;
    private String uName;
    private String password;
    private String hobbies[];
    private String designation;
    private String gender;
    public String getfName() {
        return fName;
    }
    public void setfName(String fName) {
        this.fName = fName;
    }
    public String getlName() {
        return lName;
    }
    public void setlName(String lName) {
        this.lName = lName;
    }
    public String getuName() {
        return uName;
    }
    public void setuName(String uName) {
        this.uName = uName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String[] getHobbies() {
        return hobbies;
    }
    public void setHobbies(String[] hobbies) {
        this.hobbies = hobbies;
    }
    public String getCountry() {
        return designation;
    }
    public void setCountry(String country) {
        this.designation = country;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
    public String getGender() {
        return gender;
    }
}
```

| © 2011 Cognizant

Lend a Hand- Success page Design

- Create a success page which includes the header page.
- The header page should be included based on the designation selected. Say “Manager Home Page”, “Trainee Home Page” (or) “Developer Home Page”
- The body of the success page should display all the details which the user entered in registration page in a tabular format





Lend a Hand – Develop traineesHeading.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<div style="background-color: aqua;color: Black;">
<h1 style="text-align: center;">Trainee's Home Page</h1>
<h2>Welcome <%=request.getParameter("name")%></h2></div>
```

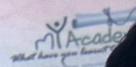
Access the parameter set
and print the message.

Lend a Hand – Develop managersHeading.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<div style="background-color: gray;color:Brown;">
<h1 style="text-align: center;">Manager's Home Page</h1>
<h2>Welcome <%=request.getParameter("name")%></h2>
</div>
```

Lend a Hand – Develop developersHeading.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<div style="background-color: olive; color: blue;">
<h1 style="text-align: center;">Developer's Home Page</h1>
<h2>Welcome <%=request.getParameter("name")%></h2>
</div>
```



Lend a Hand – Deploy and Run

Step 1 : Deploy and run the application

Step 2 : Invoke the registration.jsp from the browser

<http://localhost:8080/ActionDemo/registration.jsp>

Step 3 : Enter the registration details, designation as **manager** and click register.

Step 4 : The success page should be displayed with the **manager** header.

Continue step 3 and for 4 for **Developer** and **Trainee** designation. The screen should display the appropriate header pages.