

Java Beans and JSP Actions

Content By: Bairon Vasquez

Questions?



Agenda

- What is Java bean?
- What is JSP action tag?
- Type of JSP action tags
- How to use JSP action tags?

What is server side java bean?

- A server side java bean is a class used to store the details of real world entities
 - Example: Employee > Employee Name and Employee Salary,
Student > Student Name, Student Address
- Bean is a plain java class which contains
 - Fields (or) Properties: Fields to store data. Example: Employee Name, Salary
 - Method: Methods are retrieving and modifying the attributes like setEmployeeName(), setStudentAddress(). The methods are referred to as accessors/mutator

Java Bean design conventions

- A Java bean component property can be read/write, read-only, or write-only.
- The bean property needs to be accessible using public methods.
 - For each readable property, the bean must have a method as illustrated below to retrieve the property value

Syntax:

```
Public Datatype get<PropertyName> {  
    return value; // Returns the property value  
}
```

Example:

```
Public int getEmployeeId {  
    return empld; // Returns the employee id value  
}
```

Java Bean design conventions

- For each writable property, the bean must have a method as illustrated below,

Syntax:

```
Public set<PropertyName>(Datatype newValue) {  
    property= newValue; // sets the new value into the  
    property  
}
```

Syntax:

```
Public setEmployeeId(int newEmpld) {  
    empld= newEmpld; // sets the new employee id into the  
    employee id property  
}
```

Sample Bean Class

```
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;  
    }  
}
```

The bean properties User Name and password.
Declared **private**.

Getter method for retrieving the user name.
Declared **Public**.

Setter method for modifying user name.
Declared **Public**.

Note the getter and setter names convention
is
1. **get + property name** with the first letter
of the property name capitalized
and
2. **set+ property name** with the first letter of
the property name capitalized .



Why to follow naming convention for methods?

8

- The web container maps the request attribute with the method names and triggers the appropriate methods for retrieving or setting the values.

Example: **Request Attribute Name:** empld

Method Triggered: setEmpld(), getEmpld()

Note: Web container automatically capitalizes the first character of the attribute name and concatenates “set” (or) “get” to it for either setting or retrieving property values in bean.

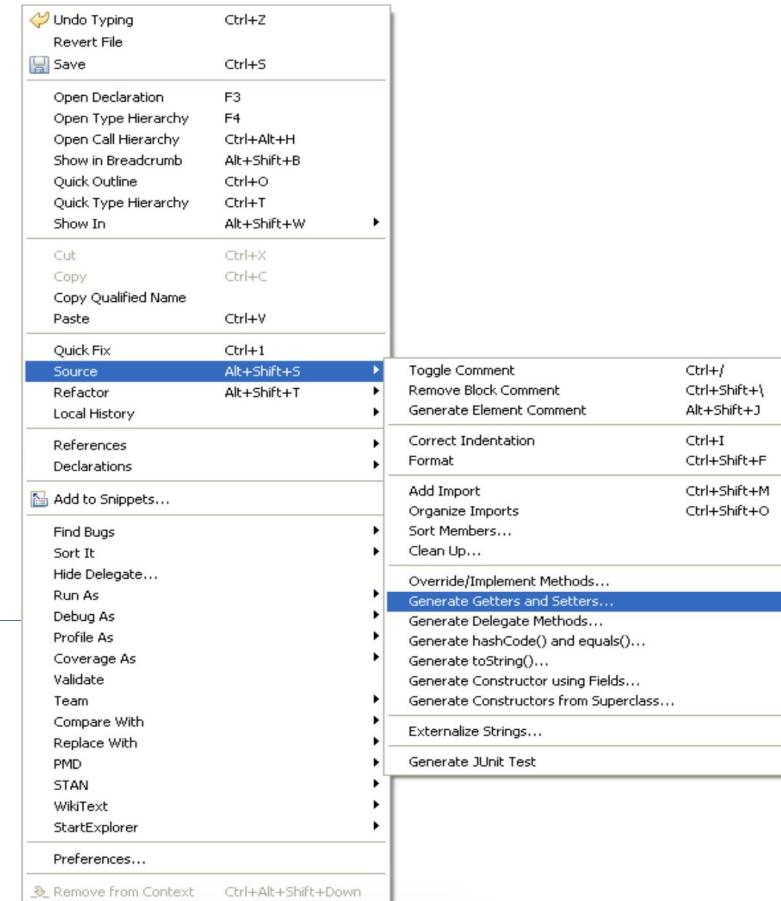
- How is this done?

This is done using **reflection and introspection**.

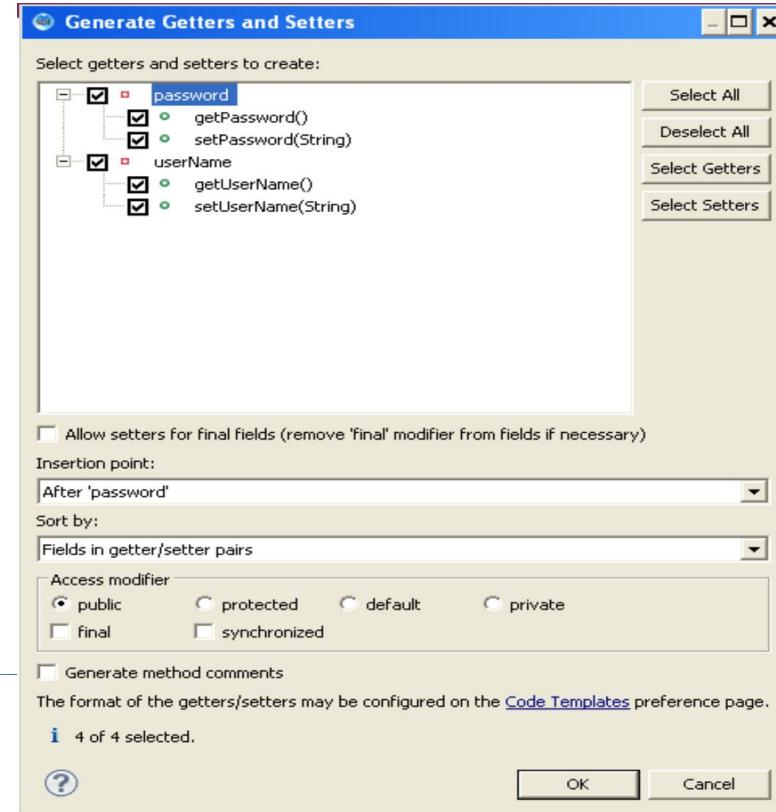
Reflection and introspection are java API's used for finding the properties and methods of a class and invoking it dynamically.

Easy steps for creating Beans using SDE

- Step 1: Create a class named User
- Step 2: Declare two properties for the class
 - userName – type String.
 - Password – type String.
- Right click source ->
source -> generate getter and setter



Easy steps for creating Beans using SDE



Select the properties and the accessor methods to be generated and click Ok.

Need for Beans in JSP

Beans are used in JSP for collectively storing some information.

Beans makes transfer of data between JSP's easier.

For example, if you are handling with a registration form all the registration details can b loaded into a **RegistrationBean** and can be transported across other components as a single object.

Need for Beans in JSP

❑ How to set value to a Bean?

Values can be set to the bean using the setter method.

```
userBean.setName(request.getParameter("name"));
```

Reads the parameter **name** from the request and sets it to the property **name** in **userBean**.

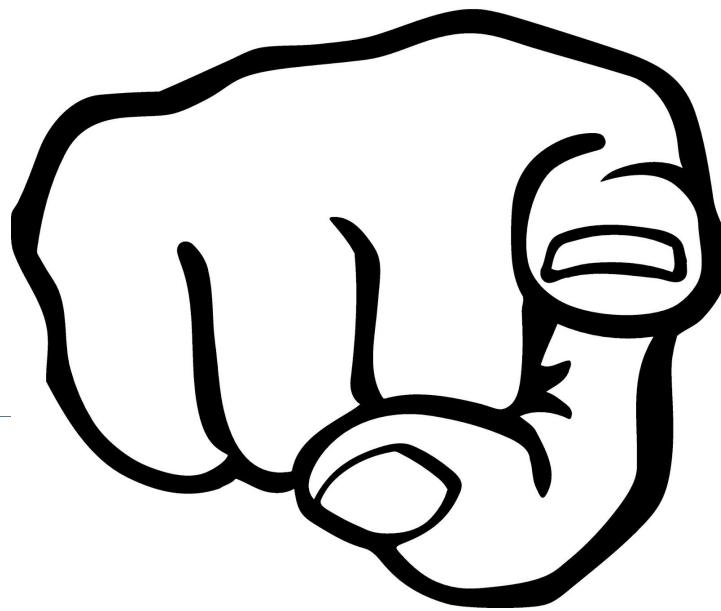
❑ How to read values from a bean?

Values can be retrieved from a bean using the getter method.

```
String username = userBean.getName();
```

Reads the property value **name** from the bean and assigns it to a variable.

Your turn



Your Turn - Using Java beans in JSP

In this demo we are going to familiarize how java beans are used with JSP. We are going to develop a login page and validate the credentials and redirect the response to success or error page.

Components:

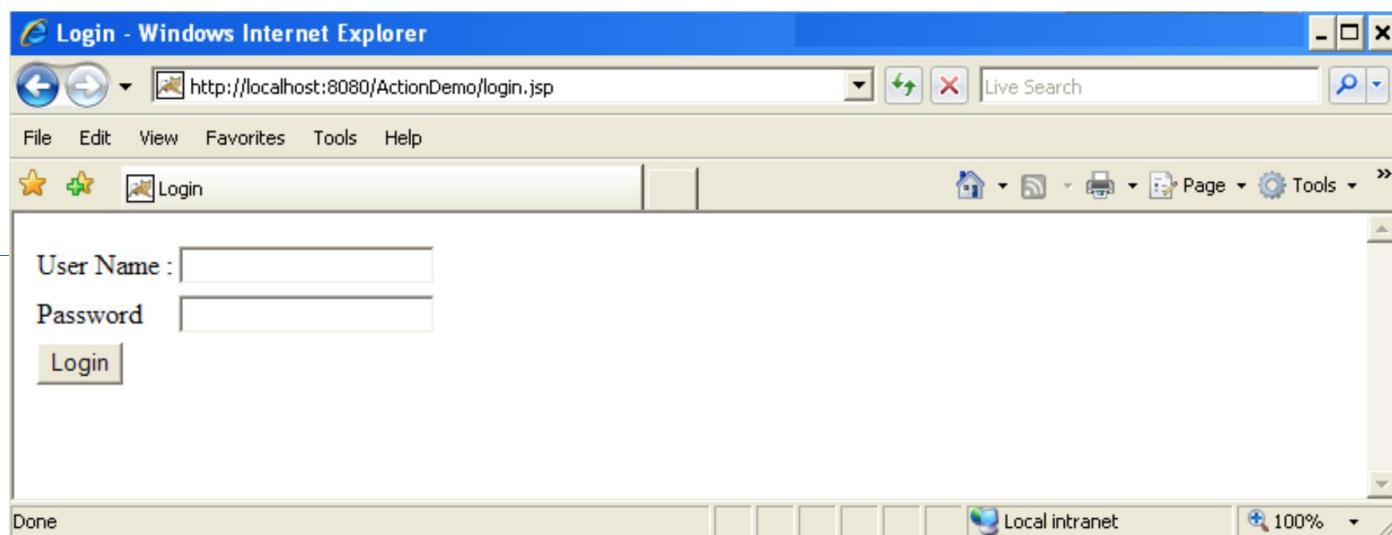
- 1. login.jsp** : login page
- 2. success.jsp** : success page
- 3. User.java** : The user bean class.

The login.jsp will validate the username/password, create a user bean if successful redirect it to success page. Success page should access the bean properties and display it on the screen.

Your Turn – login.jsp

We will create a login page as shown below,

1. User Name – Text Field
2. Password – Password Field
3. Login – Submit button



Your Turn – Develop login.jsp

```

<%>
if(request.getParameter("login") !=null){
    User user=new User();
    user.setUserName(request.getParameter("userName"));
    user.setPassword(request.getParameter("password"));
    if("tom".equals(user.getUserName()) &&
       "jerry".equalsIgnoreCase(user.getPassword()))
    {
        RequestDispatcher dispatcher=request.getRequestDispatcher("success.jsp");
        request.setAttribute("user",user);
        dispatcher.forward(request,response);
    }   else
    {
        out.print("User Name and Password does not match");
    }
}
%>
<body>
<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>

```

Sets the values to the bean using the setter methods

Reads the values to the bean using the getter methods

Sets the bean as a request attribute



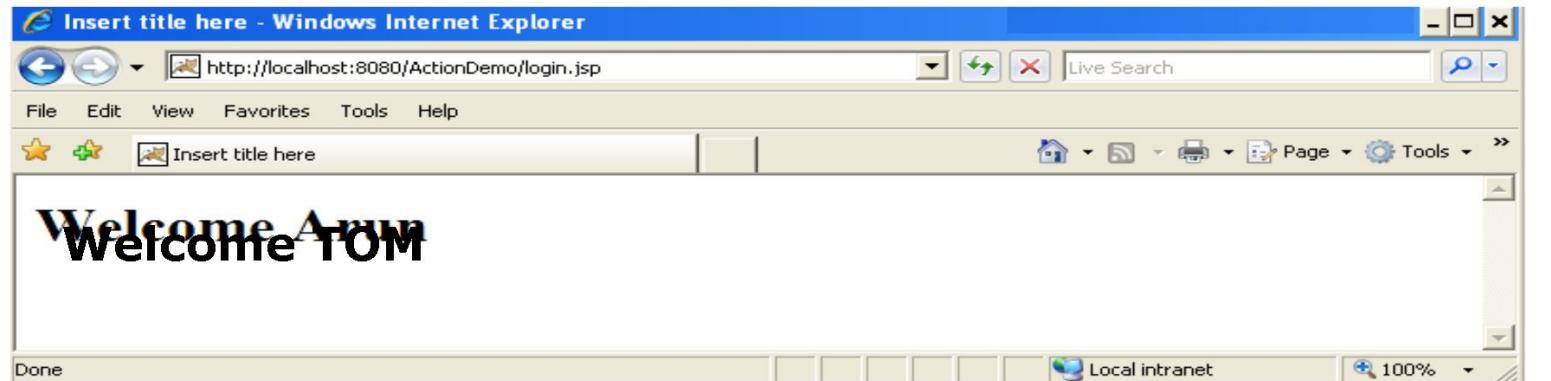
Your Turn – Develop user Bean

Develop the user bean as mentioned below.

```
package com.catp.beans;

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;
    }
}
```

Your Turn – Invalid Login



```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="com.catp.beans.User"%>
<!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>Insert title here</title>
</head>
<%
    User user = (User) request.getAttribute("user");
%>
<body>
<h1>Welcome <%=user.getUserName()%></h1>
</body>
</html>
```

Reads the bean object from request and type casts it.

Reads the username from the bean using getter method.

Your Turn – Deploy and Run

Step 1: Deploy and run the application

Step 2: Call login.jsp from the browser

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

Step 3: Enter user name and password as “tom” & “jerry” and click login. The success page should be displayed.

Step 4: Enter an invalid username and password the control should be redirected to an error page.

What is JSP action tag?

JSP action tags are a set of predefined tags provided by the JSP container to perform some common tasks thus reducing the java code in JSP.

Some of the common tasks are:

- Instantiate java bean object.
- Setting values to beans.
- Reading values from beans.
- Forward the request to another resource.
- Including another resorce.

JSP action tag syntax

```
<jsp:action_name attribute="value" />
```

Where,

action_name : specifies the name of the action to be performed

attribute : specifies the attributes relevant to the action.

Example:

```
<jsp:include page="myPage.jsp" />
```

Where “**include**” is the action and “**page**” is the attribute of the action include, this performs the include action of including a JSP page.

Action tags in JSP

The followings are the action tags available in JSP

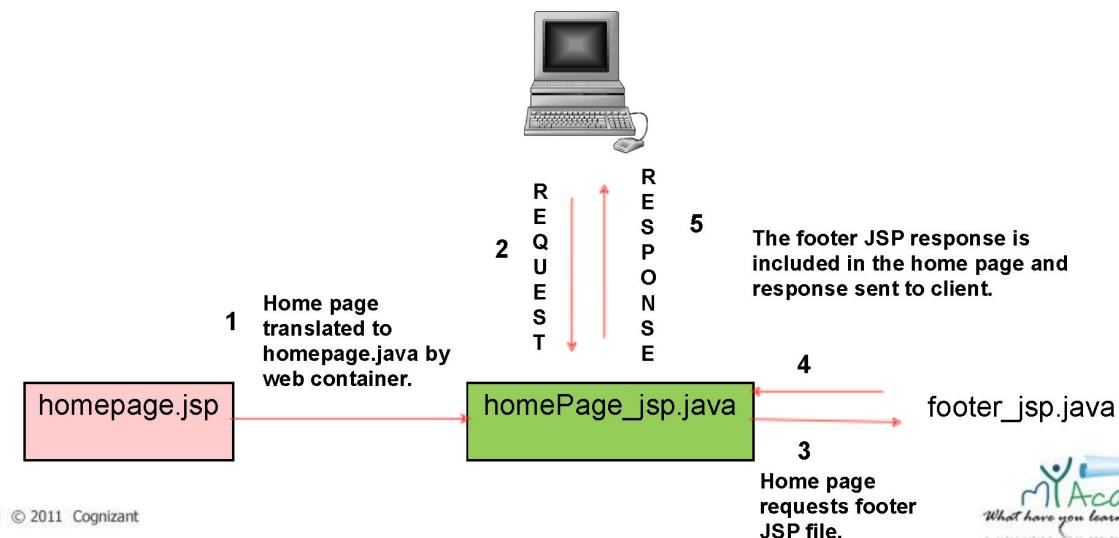
- **jsp:include**
- **jsp:forward**
- **jsp:usebean**
- **jsp:setProperty**
- **jsp:getProperty**
- **jsp:fallback**
- **jsp:element**
- **jsp:body**
- **jsp:text**
- **jsp:attribute**
- **jsp:param**
- **jsp:plugin**

The highlighted ones will be covered in detail in the coming slides.

JSP:include

- ❑ Used for dynamically including the pages
- ❑ Includes the output of the included page during run time
- ❑ Contents of the page are not included – Only the response is included.

Assume footer.jsp is included homepage.jsp



JSP:include tag

Syntax :

```
<jsp:include page="PageName" />
```

Example : Assume that the following line is included in *index.jsp*

```
<jsp:include page="myPage.jsp" />
```

Here the *myPage.jsp* response is included in the *index.jsp*

Directive Vs Action include

Consider a scenario in which we are asked to create a login application. They have the following requirements:

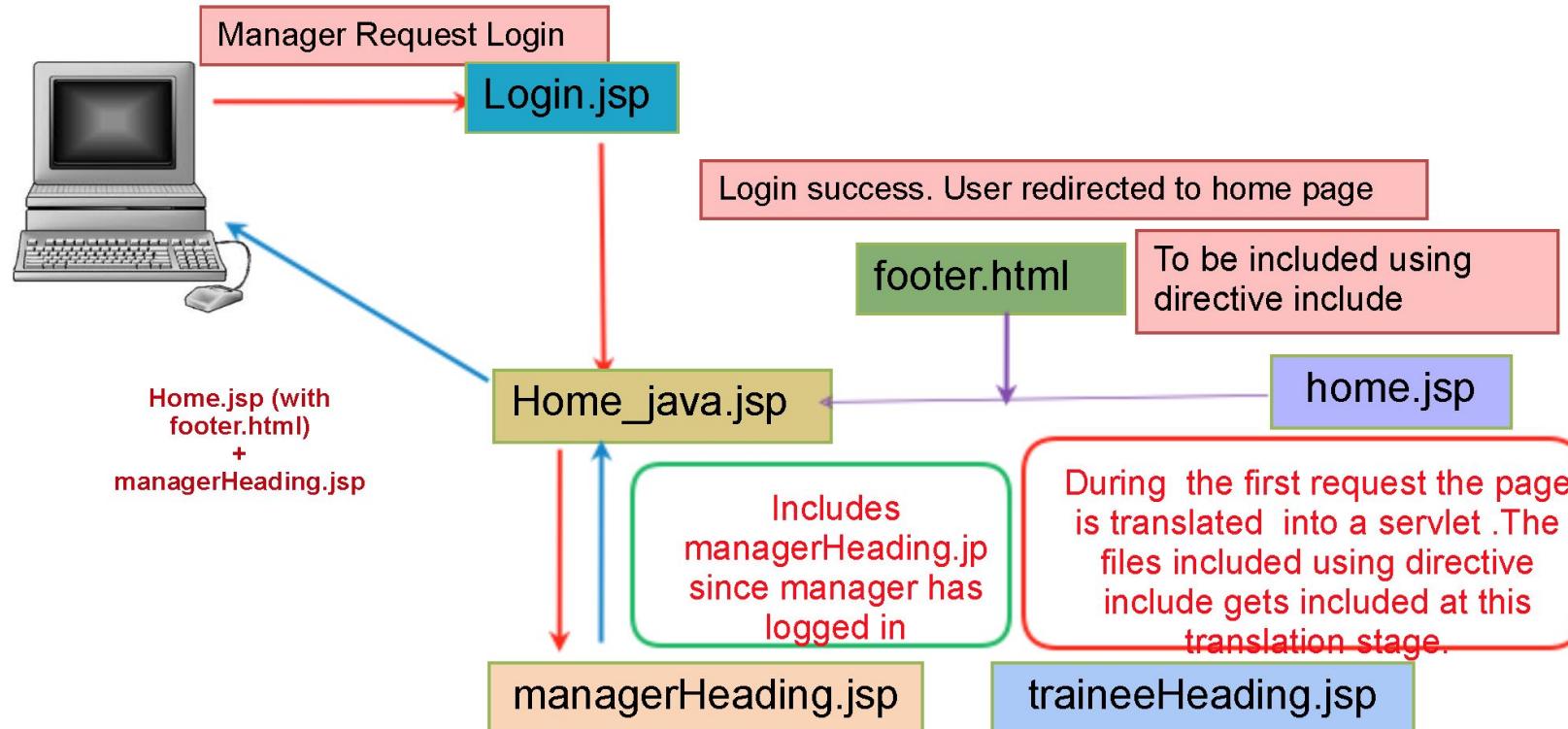
- Home page is common for all categories of employees
- The heading is different for different designation of employees and should be included in all the pages based on the employee designation
- The footer page is common for all and should be included in all the pages.

Given the above scenario,

- What is the include mechanism you will use to include the heading page?
- What is the include mechanism you will use to include the footer?

Difference between Directive Vs Action include

26



Jsp:forward

The **forward** action tag is used to transfer control to a static or dynamic resource.

- ❑ The forward action terminates the action of the current page and forwards the request to another resource such as static page, another JSP page, or a Java servlet.
- ❑ The static or dynamic resource to which control has to be transferred is represented as a URL.
- ❑ The user can have the target file as an HTML file, another JSP file, or a servlet.

Jsp:forward Syntax

- Syntax:

```
<jsp:forward page="URL" />
```

Forwards the request to the specified URL

- Example:

```
<jsp:forward page="success.jsp" />
```

Forwards the request to the success.jsp

- When used?

- Provides the same functionality of the forward() method in RequestDispatcher interface
- Used for forwarding request from one JSP to another

jsp:param

- The jsp:param action is used to add the specific parameter to current request
- The jsp:param tag can be used inside a jsp:include (or) jsp:forward
- Syntax:

```
<jsp:forward page =“URL” />  
    <jsp:param name =“paramName” value="parameterValue"/>  
</jsp:forward>  
  
<jsp:include page =“URL” />  
    <jsp:param name =“paramName“ value="parameterValue"/>  
</jsp:include>
```

jsp:param Example

- Scenario – Consider a scenario on successful login the user needs to be forwarded to the success page with the username set as a 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>
```

success.jsp – success page

username – Request attribute name

name – The variable containing the user's name

jsp:useBean

- The <jsp:useBean> tag attempts to locate 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"/>
```

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 <u>beanName</u> , no Bean is instantiated. The package and class name are case sensitive.
<u>beanName</u>	Gives the name of the bean, as you would supply it to the instantiate method of Beans.

jsp:useBean

- Example:

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

- How it works? Sequence of steps:
 - Attempts to locate a bean with the name "UserBean" in the request scope
 - If it finds the bean, stores a reference in the variable user
 - 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" />
```

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 an instance variable with the property name specified and accessor/mutator methods

value – the value to be set for the property

jsp:setProperty mapping HTML forms

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;
    }
}
```

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" />
```

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

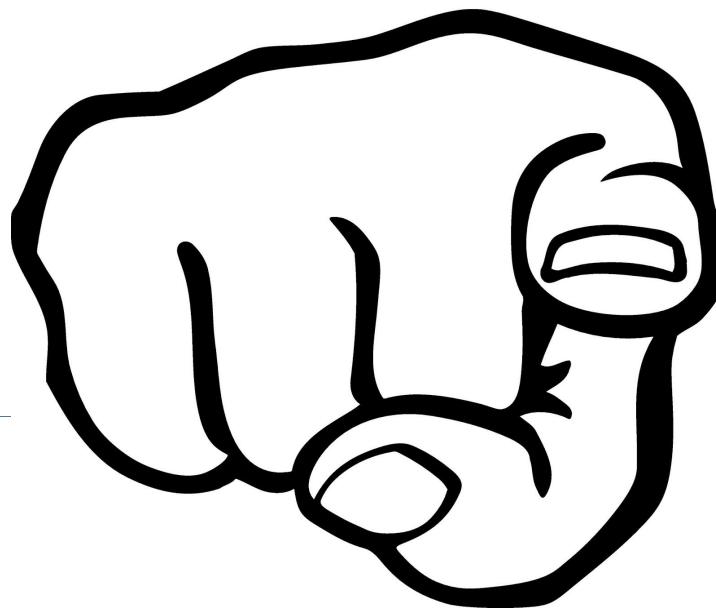
Recap – JSP Action Tags

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.

Questions?



Your turn



Your Turn – Using Java beans in JSP

- 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`

Your Turn (Details)

- Consider a scenario in which ABC SoftCorp have approached you to create a registration form for their Employees for maintaining profile of each employee in the company
- The requirement is as stated below:
 - There are three designation of employees
 - ▶ Manager
 - ▶ Developer
 - ▶ Trainee
 - Employees should enter details in the registration form
 - On successful entry the employee is forwarded to their respective home pages

Your Turn (Details 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

- registration.jsp – Registration page used by employees to register
- success.jsp – Common home page for all employees
- traineesHeading.jsp – Heading page for trainee's success page
- developersHeading.jsp – Heading page for developer's success page
- managersHeading.jsp – Heading page for manager's success page
- Employee.jsp – This is a home page for displaying the employee details

Your Turn (Code Assistance)

□ Sample Code (registration.jsp):

```

<%@ 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: 25px;">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">Developer</option>
        </select></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="Register" name="register" /></td>
    </tr></table></form>
</body>
</html>

```

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

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

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

Your Turn (Code Assistance)

□ Sample Code (Employee.java):

```
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;  
    }  
}
```

Your Turn (Code Assistance)

□ Sample Code (success.jsp):

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="com.catp.beans.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
<style type="text/css">
html,body {
    height: 100%;
}
td{ border:1px solid blue;}
table {
    border: 2px solid red;
}
</style>
</head>
<jsp:useBean id="userBean" class="com.catp.beans.Employee" scope="request"></jsp:useBean>
<body>
<div>
<% String designation = request.getParameter("designation");
if (designation.equals("1")) {
%> <jsp:include page="managerHeading.jsp">
<jsp:param value=<%=userBean.getName() %>" name="name" /></jsp:include>
<%
} else if (designation.equals("2")) (%> <jsp:include
page="traineesHeading.jsp">
<jsp:param value=<%=userBean.getName() %>" name="name" /></jsp:include>
<%
} else if (designation.equals("3"))
(
%> <jsp:include page="developersHeading.jsp">
<jsp:param value=<%=userBean.getName() %>" name="name" /></jsp:include>
<%
)
</div>
<hi style="color: green;"><%=request.getParameter("message") %></hi>
<hi>Registration Details</hi>
<table>
<tr><td>First Name</td> <td><jsp:getProperty property="fName" name="userBean" /></td></tr>
<tr><td>Last Name</td><td><jsp:getProperty property="lName" name="userBean" /></td> </tr>
<tr><td>User Name</td> <td><jsp:getProperty property="uName" name="userBean" /></td></tr>
<tr><td>Gender</td> <td><jsp:getProperty property="gender" name="userBean" /></td>
</tr>
<% String hobbies[] = userBean.getHobbies();%>
<tr><td>Hobbies</td><td>
<% for (String hobby : hobbies)
        ( out.print(hobby + ",")); %></td></tr>
</table>
</body>
</html>

```

Use **JSP:include** to include the respective pages and use **jsp:param** to set name value as a parameter for heading page to display.

Reads the value using **getProperty** action and prints it.

Your Turn (Code Assistance)

□ Sample Code (traineeHeading.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.

Your Turn (Code Assistance)

□ Sample Code (managerHeading.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>
```

Your Turn (Code Assistance)

□ Sample Code (developerHeading.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>
```

Let's Take A Break...

51



**Time for
a Break**

Summary

- Java Beans
 - Java classes that model an object in web applications
- JSP Action Tags
 - `jsp:include`, `jsp:forward`, `jsp:useBean`, etc.

END

END