SESSION-13
JSP-ACTION TAGS

STANDARD ACTIONS:

These are basically used to pass runtime information to the container. As a part of JSP we have the following *standard actions*; they are **<JSP:forward/>**, **<JSP:include/>**, **<JSP:param/>**, **<JSP:useBean/>**, **<JSP:setProperty/>** and **<JSP:getProperty>**.

1. <JSP:forward/>:

When we want to forward a request and response to the destination JSP page from the source JSP we must use <JSP:forward>.

Syntax:

Without body:

```
<JSP:forward page="relative or absolute path of JSP page"/>
```

With body:

```
<JSP:forward page=" relative or absolute path of JSP page">
      <JSP:param name="param name1" value="param value1"/>
      <JSP:param name="param name2" value="param value2"/>
</JSP:forward>
```

For example:

```
<JSP:forward page="y.jsp">
      <JSP:param name="v1" value="10"/>
      <JSP:param name="v2" value="20"/>
</JSP:forward>
```

When we use this tag we get the response of destination JSP page only but not source JSP page.

2. <JSP:include/>:

This tag is used for processing a client request by a source JSP page by including other JSP pages and static resources like html's. One source JSP can include 'n' number of server side resources and finally we get the response of source JSP only.

Syntax:

Without body:

```
<JSP:include page="relative or absolute path of JSP page"/>
```

With body:

```
<JSP:include page=" relative or absolute path of JSP page">
      <JSP:param name="param name1" value="param value1"/>
      <JSP:param name="param name2" value="param value2"/>
</JSP:include>
```

For example-1:

```
<JSP:include page="y.jsp">
      <JSP:param name="v1" value="10"/>
      <JSP:param name="v2" value="20"/>
</JSP:include>
```

For example-2:

```
<JSP:include page="z.jsp">
```

```
        <JSP:param name="v3" value="30"/>
        <JSP:param name="v4" value="40"/>
</JSP:include>
```

3. <JSP:param/>:

This tag is used for passing the local data of one JSP page to another JSP page in the form of (key, value) pair.

Syntax:

```
<JSP:param name="name of the parameter" value="value of the parameter"/>
```

Here, name represents name of the parameter or attribute and it must be unique value represents value of the parameter and should be always string. <JSP:param/> tag should be used in connection with either <JSP:forward/> or <JSP:include/>.

For example-1:

```
<JSP:forward page="y.jsp">
      <JSP:param name="v1" value="10"/>
      <JSP:param name="v2" value="20"/>
</JSP:forward>
```

For example-2:

```
<JSP:include page="y.jsp">
      <JSP:param name="v1" value="10"/>
      <JSP:param name="v2" value="20"/>
</JSP:include>
```

For example-2:

```
<JSP:include page="z.jsp">
      <JSP:param name="v3" value="30"/>
      <JSP:param name="v4" value="40"/>
</JSP:include>
```

JavaBeans in JSP

A JavaBeans class is a software reusable component. Every JavaBeans class must belong to a package. Since, it is reusable. Every JavaBeans class modifier must be public. Every JavaBeans class must contain set of data members (known as properties).

For each and every data member of JavaBeans class we must have set of set methods whose general representation is:

```
public void setXxx (datatype FormalVariableName)
{
      ……..;
      ……..;
      ……..;
}
```

For each and every data member of JavaBeans class we must have set of get methods whose general representation is:

```
public datatype getXxx ()
{
      ……..;
      ……..;
      ……..;
}
```

The set of set methods are used for setting the values to JavaBeans class object whereas set of get methods are used for getting the values from JavaBeans class object.

Properties or characteristics of JavaBeans:

Every JavaBeans class contains simple property, boolean property and indexed properties.

- A simple property is one in which a method takes and returns elementary or single value (set of set and get methods are known as *simple properties* of a JavaBeansclass.)
- A boolean property is one in which a method takes or return boolean value.
- An indexed property is one in which a method takes or return array of values.

Develop a JavaBeans class which will check whether the username and password correct ornot?

Answer:

Test.java:

```
package abc;
public class Test
{
	String uname;
	String pwd;
	public void setUname (String uname)
	{
		this.uname=uname;
	}
	public void setPwd (String pwd){
		this.pwd=pwd;
	}
	public String getUname ()
	{
		return (uname);
	}
	public String getPwd ()
	{
		return (pwd);
	}
	public boolean validate ()
	{
		if (uname.equals ("kalpana") && pwd.equals ("test"))
		{
			return (true);
		}
		else
		{
			return (false);
		}
	}
}
```

NOTE: It is highly recommended to use a JSP page with scriptless.

Using a JavaBean class into JSP page the following tags are used to use a JavaBean class as a part of JSP:

4. **<JSP:useBean/>:** This tag is used for creating an object of JavaBeans class as a part of JSP page.
   Syntax:

```
<JSP:useBean     id="object name of a JavaBeans class"
                 class="fully qualified name of JavaBeans class"
                 scope="scope attribute"
                 type="name of base interface or class" />
```

- Here, *JSP* represents prefix or short name of *useBean* tag.
- *useBean* represents a tag for representing details about JavaBeans class.
- *id* and *name* are the mandatory attributes of *useBean* tag.
- *id* represents object name of JavaBeans class.
- *name* represents fully qualified name of JavaBeans class.
- *scope* represents the visibility or accessibility of a JavaBeans class.

The scope attribute represents any one of the following:

- page - represents a JavaBeans class object can be accessed in current JSP page only. It cannot be accessed in another JSP pages. By default the scope is *page*.
- request - represents a JavaBeans class object can be accessed in those JSP pages which are participating in processing a single request.
- session - represents a JavaBeans class object can be accessed in all the JSP pages which are participating in a session and it is not possible to access in those JSP pages which are not participating in session.
- application - represents a JavaBeans class object can be accessed in all the JSP pages which belongs to the same web application but it is not possible to access in those JSP pages which are belongs to other web applications.

The type attribute represents specification of base interface or class name of a JavaBeans class.

For example:

```
<JSP:useBean id="eo"
             class="ep.Emp"
             scope="session"
             type="ep.GenEmp" />
```

When the above statement is executed the container creates an object eo is created in the following way:

```
ep.GenEmp eo=new ep.Emp ();
```

If we are not specifying the value for type attribute then the object eo is created in the following way:

```
ep.Emp eo=new ep.Emp ();
```

NOTE:

In the above <JSP:useBean/> tag if we use a tag called <JSP:setProperty/> then that tag becomes body tag of <JSP:useBean/> tag.

5. <JSP:setProperty/>:

This tag is used for setting the values to the JavaBeans class object created with respect to <JSP:useBean/> tag.

Syntax-1:

```
<JSP:setProperty      name=”object name of a JavaBeans class”
                      Property=”property name of JavaBeans class”
                      Value=”value for property” />
```

For example:

```
<JSP:useBean id=”eo” class=”ep.Emp”>
      <JSP:setProperty   name=”eo”
                         Property=”empno”
                         Value=”123” />
</JSP:useBean>
```

When the above statement is executed by the container, the following statement will be taken place.

```
ep.Emp eo=new ep.Emp ();
eo.setEmpno (“123”);
```

The above syntax used to call a specific set method by passing a specific value statically.

Syntax-2:

```
<JSP:setProperty      name=”object name of a JavaBeans class”
                      property=”*” />
```

The above syntax is used for calling all generic set methods by passing there values dynamically.

For example:

```
<JSP:useBean id=”eo” class=”ep.Emp”>
      <JSP:setProperty name=”eo” property=”*” />
</JSP:useBean>
```

Dynamically we can pass the values through HTML program to a JSP page. All the form fields of HTML program must be similar to data members or properties of a JavaBeans class and in the same order we must define set of set methods.

6. **<JSP:getProperty/>:** This tag is used for retrieving the values from JavaBeans class object.

   Syntax:

   ```
   <JSP:getProperty      name="object name of JavaBeans class"
                         property="property name of JavaBeans class" />
   ```

   For example:

   ```
   <JSP:getProperty name="eo" property="empno" />
   [or]
   <%= eo.getEmpno () %>
   ```

For example (bean1):

web.xml:

```
<web-app>
</web-app>
```

Bean.html:

```
<html>
  <body>
       <h3>Bean tag test</h3>
       <form name="b1" action="bean.jsp" method="post">
         Enter ur name : <input type="text" name="b1_name"><p>
         Select the language : 
         <select name="b1_lang">
              <option value=""></option>
              <option value="c"> C </option>
              <option value="c++"> C++ </option>
              <option value="java"> Java </option>
              <option value=".net"> .NET </option>
         </select><p>
         <input type="submit" value="Send"> 
         <input type="reset" value="Clear">
       </form>
  </body>
</html>
```

Bean.jsp:

```
<html>
  <body>
       <jsp:useBean id="obj" class="tp.TechBean">
         <jsp:setProperty name="obj" property="*"/>
       </jsp:useBean>
       <h3>Result of bean action tags</h3>
       Hello <jsp:getProperty name="obj" property="b1_name"/><p>
       <jsp:getProperty name="obj" property="b1_lang"/><p>
       <jsp:getProperty name="obj" property="langComments"/><p>
       <h3>Result of expression tags</h3>
       Name : <%= obj.getName () %><br>
       Language : <%= obj.getLang () %><br>
       Comment : <%= obj.getLangComments %>
  </body>
</html>
```

TechBean.java:

```
package tp;
public class TechBean
{
       String name;
       String lang;
       public TechBean () //recommended to write
       {
       }
       public void setName (String name)
       {
```

```
		this.name=name;
	}
	public void setLang (String lang)
	{
		this.lang=lang;
	}
	public String getName ()
	{
		return name;
	}
	public String getLang ()
	{
		return lang;
	}
	public String getLangComments ()
	{
		if (lang.equals ("c"))
		{
			return ("Mr. Kalyan Reddy is the best faculty in Hyderabad");
		}
		else if (lang.equals ("c++"))
		{
			return ("Kalyan IT is the best institute for it");
		}
		else if (lang.equals ("java"))
		{
			return ("Mr KVR is the best faculty in Hyderabad");
		}
		else if (lang.equals (".net"))
		{
			return ("Mr. Nageswara is the best faculty in Hyderabad");
		}
		else
		{
			return ("No idea..!");
		}
	}
}
```

CheckBean.html:

```
<html>
  <body>
       <form name="checkbean" action="CheckBean.jsp" method="post">
       Enter user name : <input type="text" name="checkbean_name" value=""><br>
       Enter password : <input type="password" name="checkbean_pwd" value=""><br>
       <input type="submit" value="Send"> 
       <input type="reset" value="Clear">
       </form>
  </body>
</html>
```

CheckBean.jsp:

```
<%@ page import="mypack.CheckBean" %>
<jsp:useBean id="check" class="CheckBean" scope="session">
       <jsp:setProperty name="check" property="*"/>
</jsp:useBean>
<%= check.validate () %>
```

CheckBean.java:

```
package mypack;
public class CheckBean
{
       String uname;
       String pwd;
       public CheckBean ()
       {
       }
       public void setUname (String uname)
       {
              this.uname=uname;
       }
       public void setPwd (String pwd)
       {
              this.pwd=pwd;
       }
       public String getUname ()
       {
              return uname;
       }
       public String getPwd ()
       {
              return pwd;
       }
       public boolean validate ()
       {
              if (uname.equals ("asha") && pwd.equals ("krishna"))
                                   {
              }      Return true;
              else

                   Return false

      }
}
```

JSP Action Tags

The action tags are used to control the flow between pages and to use Java Bean.

jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resurce.
jsp:useBean	creates or locates bean object.
jsp:setProperty	sets the value of property in bean object.
jsp:getProperty	prints the value of property of the bean.
jsp:param	sets the parameter value. It is used in forward and include mostly.

jsp:forward action tag:

The jsp:forward action tag is used to forward the request to another resource, it may be jsp, html or another resource.

```
<html>
<body>
<h2>this is index page</h2>
<jsp:forward page="print.jsp" />
</body>
</html>
```

print.jsp

```
<html>
<body>
<% out.println("forward"); %>
</body>
</html>
```

Example of jsp:forward action tag with parameter

index.jsp

```
<html>
	<body>
		<h2>this is index page</h2>
		<jsp:forward page="print.jsp" >
			<jsp:param name="name" value="Prasad" />
		</jsp:forward>
	 </body>
</html>
```

Printdate.jsp

```
<html>
	<body>
		<% out.println("forward with param"); %>
		<%= request.getParameter("name") %>
	</body>
</html>
```

The jsp:include action tag is used to include the content of another resource it may be jsp, html or servlet.

index.jsp

```
<html>
	<body>
		<h2>this is index page</h2>
		<jsp:include page="print.jsp" />
			<h2>end section of index page</h2>
		</ jsp:include>
	</body>
<html>
```

print.jsp

```
<html>
    <body>
        <% out.println("include action tag"); %>
    </body>
</html>
```

Java Bean

A Java Bean is a java class that should follow following conventions:

- It should have a no-arg constructor.
- It should be Serializable.
- It should provide methods to set and get the values of the properties, known as getter and setter methods.

```
//Employee.java
package mypack;
public class Employee implements java.io.Serializable
{
    private int id;
    private String name;
    public Employee()
    {
    }
    public void setId(int id)
    {
        this.id=id;
    }

    public int getId()
    {
        return id;
    }
```

```
        public void setName(String name)
        {
                this.name=name;
        }
        public String getName()
        {
                return name;
         }

}
```

To access the java bean class, we should use getter and setter methods.

```
package mypack;
public class Test
{
        public static void main(String args[])
        {
                Employee e=new Employee();//object is created
                e.setName("vij");//setting value to the object
                System.out.println(e.getName());
        }
}
```

Syntax of jsp:useBean action tag

```
<jsp:useBean id= "instanceName" scope= "page | request | session | application"
class= "packageName.className" type= "packageName.className"
beanName="packageName.className | <%= expression >" >
</jsp:useBean>
```

Calculator.java

```
package mypack;
public class Calculator
    {
        public int cube(int n)
        {
            return n*n*n;
        }
    }
```

index.jsp file

```
<jsp:useBean id="obj" class="mypack.Calculator"/>
<%
    int m=obj.cube(5);
    out.print("cube of 5 is "+m);
%>
```

The jsp:setProperty action tag sets a property value or values in a bean using the setter method.

Syntax of jsp:setProperty action tag:

```
<jsp:setProperty name="instanceOfBean" property= "*" |
property="propertyName" param="parameterName" |
property="propertyName" value="{ string | <%= expression %>}"
/>
```

```
<jsp:setProperty name="bean" property="username" />
```

Example of jsp:setProperty action tag if you have to set a specific value in the property

```
<jsp:setProperty name="bean" property="username" value="Prasad" />
<jsp:getProperty name="instanceOfBean" property="propertyName" />
```

index.html

```
<html>
<body>
	<form action="process.jsp" method="post">
	Name:<input type="text" name="name"><br>
	Password:<input type="password" name="password"><br>
	Email:<input type="text" name="email"><br>
	<input type="submit" value="register">
	</form>
</body>
</html>
```

process.jsp

```
	<jsp:useBean id="user" class="mypack.User"></jsp:useBean>
	<jsp:setProperty property="*" name="user"/>
	Record:<br>
	<jsp:getProperty property="name" name="user"/><br>
	<jsp:getProperty property="password" name="user"/><br>
	<jsp:getProperty property="email" name="user" /><br>
```

User.java

```
	package mypack;
	public class User
	{
		private String name,password,email;
		void setName(String name)
		{
			this.name=name;
		}
```

```
    void setPassword(String password)
    {
        this.password=password;
    }
    void setEmail(String email)
    {
        this.email=email;
    }
    String getName()
    {
        return name;
    }
    String getPassword()
    {
        return password;
    }
    String getEmail()
    {
        return email;
    }
}
```

class Test

```
{
    public static void main(String args[])
    {
        User u=new User();
        u.setName("hhh");
        System.out.println(u.getName());
    }
}
```