

# Standard Tag(Action Element)

JSP specification provides **Standard**(Action) tags for use within your JSP pages. These tags are used to remove or eliminate scriptlet code from your JSP page because scriptlet code are technically not recommended nowadays. It's considered to be bad practice to put java code directly inside your JSP page.

Standard tags begin with the `jsp:` prefix. There are many JSP Standard Action tag which are used to perform some specific task.

The following are some JSP Standard Action Tag available:

Action Tag	Description
<code>jsp:forward</code>	forward the request to a new page
<code>jsp:getProperty</code>	retrieve a property from a JavaBean instance.
<code>jsp:include</code>	include the runtime response of a JSP page.
<code>jsp:plugin</code>	Generates client browser-specific construct that makes an OBJECT or EMBED tag for the Java Applets
<code>jsp:element</code>	Defines XML elements dynamically
<code>jsp:attribute</code>	defines dynamically defined XML element's attribute
<code>jsp:param</code>	Adds parameters to the request
<code>jsp:fallback</code>	Supplies alternate text if java applet is unavailable on the client

<code>jsp:body</code>	Used within standard or custom tags to supply the tag body.
<code>jsp:text</code>	Use to write template text in JSP pages and documents.
<code>jsp:setProperty</code>	store data in JavaBeans instance.
<code>jsp:useBean</code>	instantiates a JavaBean

## jsp:useBean Tag

If you want to interact with a JavaBeans component using the Action tag in a JSP page, you must first declare a bean. The `<`

`jsp:useBean>` is a way of declaring and initializing the actual bean object. By bean we mean JavaBean component object. Syntax of `<jsp:useBean>` tag

```
< jsp:useBean id = "beanName" class = "className"
                scope = "page | request | session | applicatio
n" >
```

Here the id attribute specifies the name of the bean. Scope attribute specify where the bean is stored. Theclass attribute specify the fully qualified classname.

name of bean i.e object      fully qualified classname

```
<jsp:useBean id="person" class="PersonBean"
              scope="request" >
```

scope of bean

Given a useBean declaration of following

```
< jsp:useBean id="myBean" class="PersonBean" scope="request" />
```

is equivalent to the following java code

```
PersonBean myBean =(PersonBean)request.getAttribute("myBean");
if(myBean=null)
{
    myBean = new PersonBean();
    request.setAttribute("myBean",myBean);
}
```

If jsp:useBean tag is used with a body, the content of the body is only executed if the bean is created. If the bean already exists in the named scope, the body is skipped.

---

### Example

In this example we will see how `< jsp:useBean>` standard tag is used to declare and initialize a bean object. We will use PersonBean class as JavaBean Component.

#### PersonBean.java

```
import java.io.Serializable;

public class PersonBean implements Serializable
{
    private String name;
```

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

### hello.jsp

```
<html>
    <head>
        <title>Welcome Page</title>
    </head>
    < jsp:useBean id="person" class="PersonBean" scope="request" />
    <body>
        Hello, < jsp:getProperty name="person" property="name" />
    </body>
</html>
```

```
< jsp:setProperty name="person" property="name" / >
```