



VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY  
**UNIVERSITY OF INFORMATION TECHNOLOGY**

---



## Chapter 7

# JSP

# Standard Action Element

Lecturer: MSc. Kiet Van Nguyen

Faculty of Information Science and Engineering

University of Information Technology, VNU-HCM

# Standard Action Elements

---

- ☐ Là các thẻ đặc biệt được nhúng vào trong các trang JSP.
  - ☐ Các thẻ này cũng được chuyển sang mã Java trước khi thi hành.
  - ☐ Có 07 standard actions elements trong JSP:
    - `jsp:include`, `jsp:forward`, `jsp:param`
      - ☐ Các thẻ này thường xuyên được sử dụng trong trang JSP
    - `jsp:useBean`, `jsp:setProperty`, `jsp:getProperty`
      - ☐ Được sử dụng khi thao tác trên JavaBean
    - `jsp:plugin`
      - ☐ Dùng để tạo thẻ `<OBJECT>` hay `<EMBED>` để nhúng applet vào trang JSP.
-

# jsp:include Action

---

- ❑ include trang động – Dùng để kết hợp các tài nguyên tĩnh và động vào 1 trang JSP tại thời điểm request time.
- ❑ Có thể truyền thông tin vào trang đính kèm.
- ❑ Cú pháp:

**<jsp:include page = "relativeURL" flush = "true" />**

**</jsp:include>**

---

# jsp:include Action

---

- ❑ Ưu điểm so với include directive
    - Dễ bảo trì
    - Dễ debug
  - ❑ Nhược điểm
    - Chậm hơn so với include directive
  - ❑ Sử dụng khi:
    - Nếu nội dung file include có thể thay đổi lúc thi hành
    - Nếu file include thường xuyên thay đổi nội dung
    - Nếu kích thước file include quá lớn.
-

# jsp:include example

---

```
<BODY>
<TABLE BORDER=5 ALIGN="CENTER">
<TR><TH CLASS="TITLE">
What's New at JspNews.com</TABLE> What s New at
JspNews</TABLE>
<P>
Here is a summary of our three
most recent news stories: most recent news stories:
<OL>
<LI><jsp:include page="/WEB-INF/includes/Item1.jsp" />
<LI><jsp:include page="/WEB-INF/includes/Item2.jsp" />
<LI><jsp:include page="/WEB-INF/includes/Item3.jsp" />
</OL>
</BODY></HTML>
```

## Item1.jsp

---

<B>Bill Gates acts humble.</B> In a startling and unexpected development Microsoft big wig and unexpected development, Microsoft big wig Bill Gates put on an open act of humility yesterday.

<A  
    HREF="http://www.microsoft.com/Never.html"  
>

More details...</A>

---



# jsp:param element

---

## ☐ Code

```
<jsp:include page="/StandardHeading.jsp">  
    <jsp:param name="bgColor" value="YELLOW" />  
</jsp:include>
```

## ☐ URL

- <http://localhost/jsp/MainPage.jsp?fgColor=RED>

## ☐ Main page

- fgColor: RED
- bgColor: null – bgColor: null

## ☐ Included page

- fgColor: RED
- bgColor: YELLOW

---



# jsp:include element

---

- ❑ Thực hiện khi trang JSP được thi hành
  - ❑ Cho phép chèn nội dung tĩnh và động vào trang JSP
    - Nội dung tĩnh: chèn trực tiếp vào trang JSP
    - Nội dung động: yêu cầu trang được include thi hành và kết quả sau khi thi hành được gắn vào trang JSP hiện tại.
  - ❑ Cú pháp:
    - `<jsp:include page="Relative address">`
-

# Exercise

---

**So sánh Include Directive vs jsp:include element**

---

# jsp:forward Action

---

- ❑ Dùng để chuyển lời yêu cầu sang thi hành 1 trang khác.

- ❑ Cú pháp

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

Hoặc

```
<jsp:forward page="another.jsp">
```

```
<jsp:param name="callingPage" value="current.jsp">
```

```
</jsp:forward>
```

- ❑ "another.jsp" can obtain parameter values as

```
<%= request.getParameter("callingPage") %>
```

- ❑ Returns "current.jsp"
-

# jsp:plugin

---

☐ Nhúng Applet vào trang web

☐ Thẻ Applet

```
<APPLET CODE="MyApplet.class"  
        WIDTH=475 HEIGHT=350>  
</APPLET>
```

☐ jsp:plugin

```
<jsp:plugin type="applet"  
            code="MyApplet.class"  
            width="475" height="350">  
</jsp:plugin>
```

---

# jsp:plugin Example

---

<BODY>

<CENTER>

<TABLE BORDER=5> <TABLE BORDER=5>

<TR><TH CLASS="TITLE">

Using jsp:plugin</TABLE>

<P>

**<jsp:plugin type="applet"**

**code="PluginApplet.class"**

**width="370" height="420">**

**</jsp:plugin>**

</CENTER> </BODY> </HTML>

---

## jsp:plugin Example

---

```
import javax.swing.*;
public class PluginApplet extends JApplet
{
    public void init() {
        WindowUtilities.setNativeLookAndFeel(
        );
        setContentPane(new TextPanel());
    }
}
```

---

# jsp:plugin và jsp:params

---

## □ Param Tags

```
<APPLET CODE="MyApplet.class"  
WIDTH=475 HEIGHT=350>  
<PARAM NAME="PARAM1" VALUE="VALUE1">  
PARAM NAME "PARAM2" VALUE "VALUE2"  
</APPLET>
```

## □ jsp:param

```
<jsp:plugin type="applet"  
code="MyApplet.class" width="475" height="350">
```

```
  <jsp:params>
```

```
    <jsp:param name="PARAM1" value="VALUE1" />
```

```
    <jsp:param name="PARAM2" value="VALUE2" />
```

```
  </jsp:params>
```

```
</jsp:plugin>
```

# Sử Dụng JavaBean

---

- ❑ Java Bean là Java Class tuân thủ:
    - Có 1 constructor không có tham số (mặc định có sẵn nếu không implement). Constructor này được gọi khi element của JSP tạo bean.
    - Các thuộc tính (field) của bean: private → thông qua phương thức **getXxx** hay **setXxx** (**accessor method**) đối với các thuộc tính cần lưu trữ (**persistent**)
    - Lưu ý:
      - ❑ JSP actions sẽ truy cập phương thức get/set/is để truy cập bean
  - ❑ JSP cung cấp 03 Java bean tag (standard actions)
    - jsp:useBean
    - jsp:setProperty
    - Jsp:getProperty
-



# CÁCH SỬ DỤNG BEAN TRONG JSP

---

## ❑ Khai báo

`<jsp:useBean id="<định danh>" class="<tên class>" [scope = "loại scope"]/>`

- id: tương tự khai báo biến, đại diện cho đối tượng truy xuất đến Bean
- class: tên class được sử dụng
- scope: khả năng chia sẻ hay tầm hoạt động của Bean. Mặc định là request

## ❑ Cơ chế thực hiện: JSP Engine/ Container thực hiện

- Tìm kiếm class tương ứng được khai báo trong class
- Khởi tạo instance thông qua constructor
- Gán định danh id cho instance để có thể truy xuất trên trang JSP
- Xác định phạm vi hoạt động của instance này.

## ❑ Cách khai báo biến dùng scriptlet tương đương Java Bean:

`<% <tên class> <định danh> = new <tên class>(); %>`

## ❑ Ví dụ:

- `<jsp:useBean id="book1" class="store.book"/>` tương đương
- `<% store.book book1 = new store.book(); %>`

# CÁCH SỬ DỤNG BEAN TRONG JSP (tt)

---

- ❑ **Ép kiểu dữ liệu:** tên biến trả về có cùng kiểu dữ liệu class sử dụng  
`<jsp:useBean id="<định danh>" class="<tên class>" type = "<kiểu dữ liệu>" [scope = "loại scope"]/>`
  - type: kiểu dữ liệu của Java
- ❑ Cách khai báo biến dùng scriptlet tương đương Java Bean:  
`<% <kiểu dữ liệu> <định danh> = new <tên class>(); %>`
- ❑ **Ví dụ:**
  - `<jsp:useBean id="abc" class="cd" type="Runnable" />`
  - tương đương `<% Runnable abc = new cd(); %>`
- ❑ Cú pháp sử dụng khác của Java bean  
`<jsp:useBean ...> statement </jsp:useBean>`
  - **Ví dụ**  
`<jsp:useBean id="count" class="ABean.AccessBean" scope="application">  
  <jsp:setProperty name="count" property="firstPage" value="ATest.jsp" />  
</jsp:useBean>`
- ❑ **Lưu ý:**
  - Sử dụng số ký tự đặt biệt trong chuỗi thì cần đặt thêm dấu “\” phía trước ký tự như ‘(\’); “(\\’), \ (\\), % (\%), ...
  - Ngoài ra, id cũng được dùng như tên biến trong scriptlets

# TRUY CẬP THUỘC TÍNH BEAN

---

`<jsp:getProperty name="<định danh>" property="<tên thuộc tính>" />`

- name: tên id đã được xác định trong jsp:useBean
- property: tên thuộc tính tương ứng khai báo trong Java Bean

□ Cách khai báo biến dùng scriptlet tương đương Java Bean  
`<%= <định danh>.getXXX() %>`

□ Ví dụ:

- `<jsp:getProperty name="book1" property="title"/>`
  - tương đương `<%= book1.getTitle()%>`
-

# THIẾT LẬP THUỘC TÍNH BEAN

---

`<jsp:setProperty name="<định danh>" property="<tên thuộc tính>" value="<giá trị>" />`

- ☐ name: tên id đã được xác định trong jsp:useBean
  - ☐ property: tên thuộc tính tương ứng khai báo trong Java Bean
  - ☐ value: giá trị (JSP expression, hằng số) được gán cho thuộc tính
  - ☐ Các khai báo tương đương trong scriptlet:  
`<% <định danh>.setXXX(<giá trị>) %>`
  - ☐ Ví dụ:
    - `<jsp:setProperty name="book1" property="title" value="JSP Book" />`
    - tương đương `<% book1.setTitle("JSP Book")%>`
-

# VỊ TRÍ ĐẶT BEAN CLASS

---

- Đặt tại:
  - <tên ứng dụng>/WEB-INF/classes

- Ví dụ:

- **Java Bean**

**package JBean;**

```
public class SimpleBean {  
    private String message ="First Bean";  
    public String getMessage() {return message;}  
    public void setMessage(String message){  
        this.message=message;  
    }  
}
```

- **Jsp**

<h1>JSP with Java Bean</h1>

<jsp:useBean id="msg" class="JBean.SimpleBean"/>

Init message (getProperty): <jsp:getProperty name="msg" property="message"/><br/>

(Scriptlet) : <%= msg.getMessage() %> <br/>

Set message - setProperty : <jsp:setProperty name="msg" property="message" value="I am bean" />

<jsp:getProperty name="msg" property="message"/>

**- Scriptlet** :

---

<%= msg.setMessage("I am Scriptlet"); %>

<%= msg.getMessage() %>

# ADVANCED SET PROPERTY

---

- ❑ Gán giá trị cho action setProperty là một expression
    - `String sMsg = request.getParameter("sms");`
    - `<jsp:setProperty name="msg" property="message" value="<%= sMsg %>" />`
  - ❑ Sử dụng thuộc tính param trong setProperty: đón nhận giá trị nhập từ request (của JSP khác, đối tượng khác hay trên URL)
    - `<jsp:setProperty name="msg" property="message" param="message" />`
  - ❑ Để thuộc tính có trong Java Bean trở thành tham số nhập với việc gán "\*" cho thuộc tính property trong setProperty
    - `<jsp:setProperty name="msg" property="*" />`
  - ❑ **Lưu ý:**
    - Action sẽ không được thực hiện nếu thiếu 1 trong những tham số nhập vì hệ thống sẽ không tự gán null.
    - Cơ chế tự động convert sẽ không đảm bảo các giá trị hợp lệ so với convert bằng việc lập trình bằng tay (programmatically)
    - Tên thuộc tính và tham số phải giống nhau
-

# VÍ DỤ

---

□ JSP:

```
<html>
<head>
  <title>Bean in JSP</title>
</head>
<body>
  <h1>JSP with Java Bean - Parameter</h1>
  <% String sMsg = request.getParameter("sms"); %>

  <jsp:useBean id="msg" class="JBean.SimpleBean"/>
  Init message (getProperty): <jsp:getProperty name="msg" property="message"/><br/>
    (Scriptlet) : <%= msg.getMessage() %> <br/>

  Set message - setProperty : <jsp:setProperty name="msg" property="message" value="I am bean" />
    <jsp:getProperty name="msg" property="message"/>
  - Scriptlet : <% msg.setMessage("I am Scriptlet"); %>
    <%= msg.getMessage() %><br/>
  Parameter message: <jsp:setProperty name="msg" property="message" value="<%= sMsg %>" />
    <jsp:getProperty name="msg" property="message"/>
</body>
</html>
```

# VÍ DỤ

---

□ Java Bean:

```
package JBean;
public class SimpleBean {
    private String message ="First Bean";
    private int num;

    public String getMessage(){
        return message;}

    public void setMessage(String message){
        this.message=message;}

    public int getNum(){
        return num;}

    public void setNum(int n){
        num=n;}}
```

---



# VÍ DỤ (tt)

---

□ JSP:

```
<html>
<head>
  <title>Bean in JSP</title>
</head>
<body>
  <h1>JSP with Java Bean - Set all Parameter</h1>

  <jsp:useBean id="msg" class="JBean.SimpleBean"/>
  <jsp:setProperty name="msg" property="*/>
  <jsp:getProperty name="msg" property="message"/> <jsp:getProperty name="msg"
  property="num"/>
</body>
</html>
```

---

# KHẢ NĂNG CHIA SẺ – SHARING HAY TẦM HOẠT ĐỘNG CỦA BEAN (tt)

---

## □ Lưu ý:

- Khi sử dụng các thành phần Java Bean dùng Session hay Application, các trang trong ứng dụng Web **phải khai báo các tag action jsp:useBean** với cùng **id** và **tên class**.
- Khi thực thi trang JSP, JSP/ Servlet engine sẽ thực hiện tra cứu scope, nếu instance bean đã tồn tại thì không cần tạo instance mới (**bỏ qua tag jsp:useBean**) để thực thi các phương thức của Bean. Ngược lại, **bean sẽ được tạo instance mới**.

- ## □ Ví dụ: Viết ứng dụng sử dụng scope của Java Bean đếm số lần truy cập trang Web
-

# VÍ DỤ

---

## □ Java Bean:

```
public class AccessBean{
    private String firstPage;
    private int accessCount=1;
    public String getFirstPage(){
        return firstPage;
    }
    public void setFirstPage(String firstPage){
        this.firstPage=firstPage;
    }
    public int getAccessCount(){
        return accessCount++;}
}
```

---

# VÍ DỤ

---

## □ JSP: page 1 - application

```
<h1>Access Bean - Page 1</h1>
```

```
<jsp:useBean id="count" class="ABean.AccessBean" scope="application">  
    <jsp:setProperty name="count" property="firstPage" value="ATest.jsp"  
    />
```

```
</jsp:useBean>
```

Of ATest.jsp (this page)- application, <a href="ATest1.jsp">ATest1.jsp</a>,  
<a href="ATest2.jsp">ATest2.jsp</a>,and <a  
 href="ATest3.jsp">ATest3.jsp</a>

```
<jsp:getProperty name="count" property="firstPage"/> was first page  
accessed.
```

```
<p>The four pages have been accessed <jsp:getProperty name="count"  
    property="accessCount"/> times.</p>
```

---

# Q&A

**Cảm ơn!**