

Bài 2: Java server pages (JSP)

Contents

1. Lý thuyết	1
2. Thẻ điều hướng.....	2
1. page directive.....	3
2. include Directive	3
3. taglib directive	3
3. Các bài demo.....	3
Bài 1:	4
Bài 2:	4
Bài 3:	Error! Bookmark not defined.
Bài 4:	5
Bài 5:	6
4. JSP Action Tags.....	7
Bài 6:	8
Java bean:.....	8
Bài 7:	8
Bài 8: (về nhà)	9

1. Lý thuyết

JSP: Java Server Page

Server (back end): jsp , servlet

Đầu tệp (pageEncoding="utf-8" – nếu tiếng Việt)

```
<%@page contentType="text/html" pageEncoding="utf-8"%>
```

Mã JSP (Scriptlets):

```
<% code java %>
```

Ví dụ viết vào văn bản:

```
<% out.println("Welcome to JSP world"); %>
```

```
<% out.write("How are you?\n");%>
```

Khai báo (Declarations):

```
<% int x=1; %>
```

Biểu thức (Expressions):

`<%= expression %>`

Chú thích (comment): đâu là chú thích??

`<!-- String name="Tô Lan Hương"; %>`

1. `<!--` đây là chú thích `-->`
2. `<p>chú ý hay nhầm lẫn : <!-- <%= name %> --></p>`
3. `<%--` hidden comment `--%>`
4. `<%//` hidden comment `%>`

Bảng sau liệt kê 9 đối tượng ẩn mà JSP hỗ trợ:

No.	Đối tượng & Mô tả
1	request Đây là đối tượng HttpServletRequest được liên kết với request.
2	response Đây là đối tượng HttpServletResponse được liên kết với response tới client.
3	out Đây là đối tượng PrintWriter được sử dụng để gửi dữ liệu tới client.
4	session Đây là đối tượng HttpSession được liên kết với request.
5	application Đây là đối tượng ServletContext được liên kết với application context.
6	config Đây là đối tượng ServletConfig được liên kết với page.
7	pageContext Điều này đóng gói việc sử dụng các tính năng cụ thể của server như JspWriters với hiệu suất cao hơn.
8	page Đơn giản là một từ đồng nghĩa với this , được sử dụng để gọi các phương thức được định nghĩa bởi lớp server được biên dịch.
9	exception Đối tượng exception cho phép các dữ liệu ngoại lệ (Exception) được truy cập bằng JSP được chỉ định.

2. Thẻ điều hướng

Có 3 loại thẻ **directive** trong JSP:

Directive	Miêu tả
<code><%@ page ... %></code>	Định nghĩa một thuộc tính page-dependent (phụ thuộc trang), như ngôn ngữ scripting, trang lỗi và các yêu cầu bộ đệm

<code><%@ include ... %></code>	Include một file trong suốt giai đoạn phiên dịch
<code><%@ taglib ... %></code>	Khai báo một thư viện thẻ, chứa các action tùy biến, được sử dụng trong trang đó

1. **page directive** được sử dụng để cung cấp các chỉ lệnh tới container mà liên quan tới JSP page hiện tại. Ta có thể mã hóa page directive bất cứ đâu trong JSP page.

Cú pháp:

```
<%@ page attribute="value" %>
```

Ví dụ:

```
<%@ page buffer="none" %>
```

```
<%@ page buffer="8kb" %>
```

```
<%@ page contentType="text/xml" %>
```

```
<%@ page contentType="text/html" %>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@ page import="java.sql.*" %>
```

```
<%@page import="java.util.Date, java.text.SimpleDateFormat"%>
```

2. **include Directive** trong JSP được sử dụng để bao một file trong suốt giai đoạn phiên dịch. Ta có thể mã hóa **include** directive bất cứ đâu trong JSP page.

```
<%@ include file="relative url" >
```

Ví dụ:

```
<%@include file="header.jsp" %>
```

```
<%@include file="logo.html" %>
```

```
<%@include file="content.jsp" %>
```

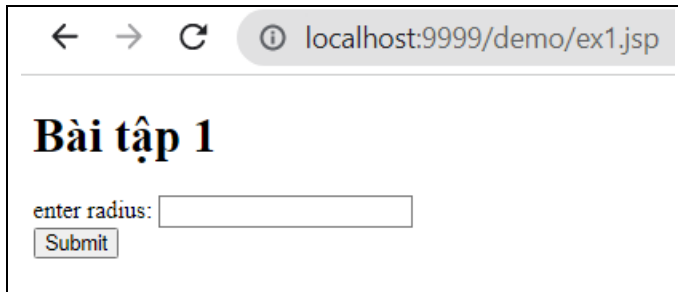
Java Server Pages API cho phép định nghĩa các Custom Tags trong JSP, mà giống như các thẻ HTML hoặc XML và một thư viện thẻ (tag library) là một tập hợp các thẻ được định nghĩa bởi người sử dụng mà triển khai các hành vi tùy biến.

3. **taglib directive** khai báo rằng: JSP page sử dụng một tập hợp các Custom Tags, thông báo vị trí của thư viện, và cung cấp một phương thức để nhận diện các Custom Tags đó trong JSP page.

```
<%@ taglib uri="uri" prefix="prefixOfTag" >
```

3. Các bài demo

Bài 1:

	<p>Có 3 cách:</p> <p>Cách 1: access to page via url: /ex1.jsp (dùng form + xử lý backend đều dùng jsp)</p> <p>Cách 2: access to page via url: /cal (làm hoàn toàn bằng servlet)</p> <p>Cách 3: access to page via url: /cal (dùng form dùng jsp, xử lý backend dùng servlet)</p>
---	---

Bài 2:

Question 1

(3 points)

You are asked to build a webpage to show your own roll number. The user can access your page via url:

/getfirst

Your Tasks include:

- (1 point) When the user access /getfirst for the first time, the system displays the form as below figure. The roll number in this form must be **your own roll number**.

HE150167

Get first letter(s)

Figure 1 – The page on first visit

- **(2 point)** When the user clicks [Get] button, the system check the input string in the textboxes. If this string representing a positive integer that called *[number]*, show the substring of first *[number]* characters of the roll number (see Figure 2). Otherwise, show the error message like Figure 3.

Note that, if *[number]* is larger than the length of the roll number, show the string of roll number only.

HE150167

Get first letter(s)

Your first 3 letter(s) is/are: HE1

Figure 2 – The page when user enter “3” and then click on button [Get]

HE150167

Get first letter(s)

Input string does not represent a positive integer

Bài 3:

Tạo form Login với username và password, username: van, password là 123. Nếu đăng nhập sai thì thông báo nhập sai (chữ màu đỏ) phía trên form nhập. Nếu nhập đúng chuyển sang trang welcome.jsp

Đăng nhập	Nhập sai
<p>← → ↻ ⓘ localhost:8084/Ex4/Login.jsp</p> <p>Example get data</p> <p>Enter your name: <input type="text"/></p> <p>Enter password: <input type="password"/></p> <p><input type="button" value="OK"/></p>	<p>← → ↻ ⓘ localhost:8084/Ex4/LoginServlet</p> <p>Example get data</p> <p>thao is invalid!</p> <p>Enter your name: <input type="text"/></p> <p>Enter password: <input type="password"/></p> <p><input type="button" value="OK"/></p>
Nhập đúng	



LoginServlet.java

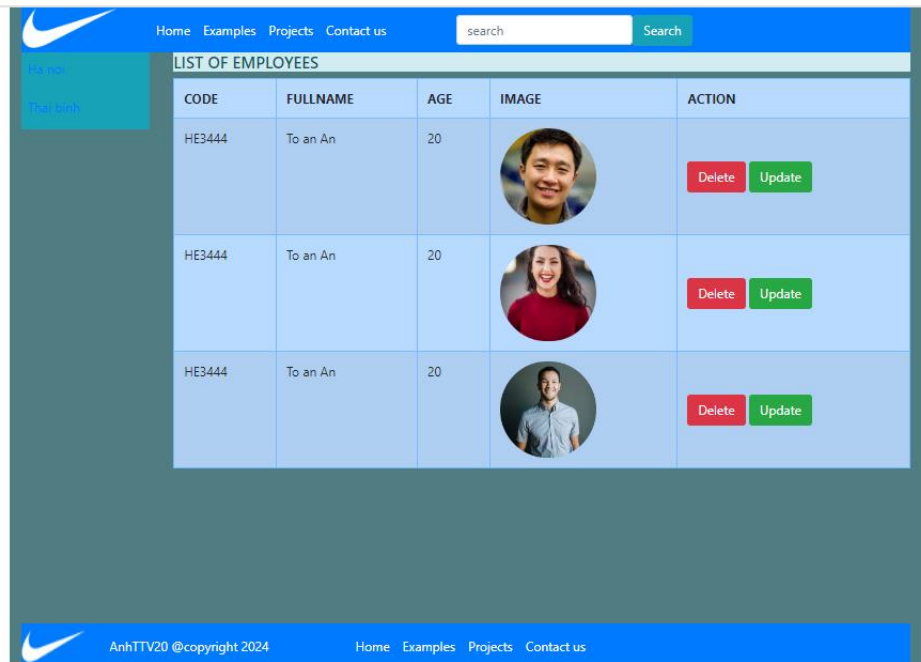
```
String u=request.getParameter("user");
String p=request.getParameter("pass");
if(u.equalsIgnoreCase("van")&& p.equals("123")){
    request.setAttribute("name", u);
    request.getRequestDispatcher("welcome.jsp").forward(request, response);
}else{
    request.setAttribute("error", "invalid");
    request.getRequestDispatcher("login.jsp").forward(request, response);
}
```

Bài 4:

Tạo trang home gồm nhiều file

Ví dụ:

localhost:9999/bootstrap/home

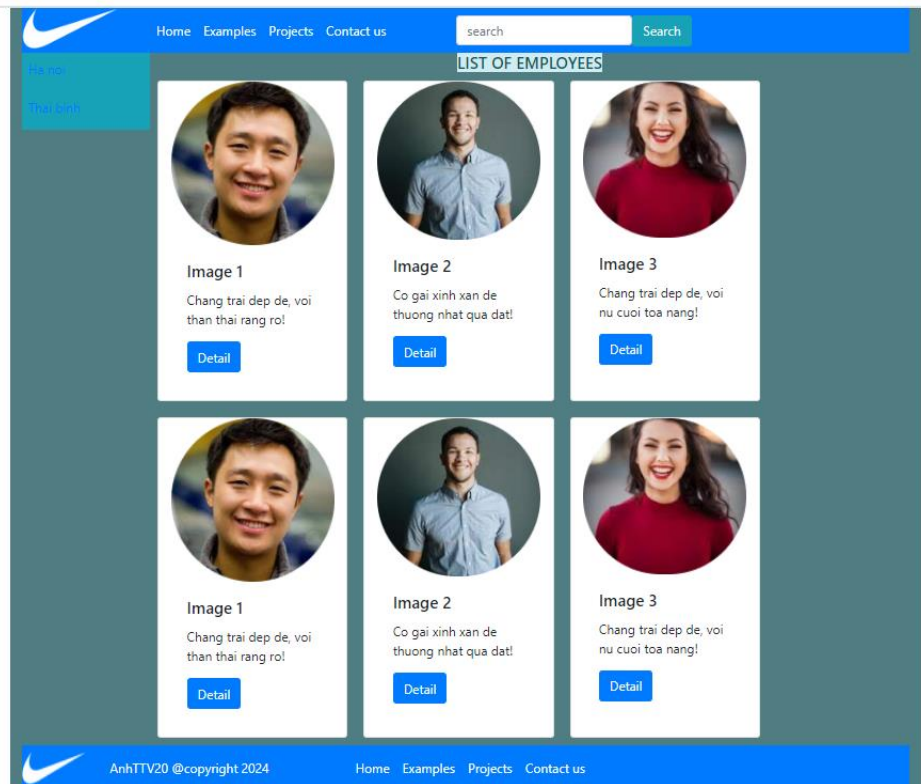


Trang home gồm các file:

Header.jsp, tableft.jsp và footer.jsp

Trang danh sách list giống hệt trang home chỉ khác phần content:

localhost:9999/bootstrap/list



4. JSP Action Tags

JSP action sử dụng các chỉ lệnh trong cú pháp XML để điều khiển hành vi của Servlet engine. Có thể chèn động một file, tái sử dụng các thành phần JavaBeans, chuyển người dùng tới trang khác, hoặc tạo HTML cho Java plugin.

JSP Action Tags	Description
jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resource.
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:plugin	embeds another components such as applet.
jsp:param	sets the parameter value. It is used in forward and include mostly.

jsp:fallback

can be used to print the message if plugin is working. It is used in jsp:plugin.

forward action trong JSP kết thúc action của trang hiện tại và chuyển request tới nguồn khác chẳng hạn như một Static page, JSP page khác, hoặc một Java Servlet.

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

`<jsp:include>` Action trong JSP

Action này cho phép bạn chèn các file vào trong các trang đang được tạo. Cú pháp như sau:

`<jsp:include page="relative URL" flush="true" />`

Bài 5: làm lại bài tập 5 dùng `<jsp:include>` Action

Java bean:

`<jsp:useBean>` Action trong JSP

useBean action trong JSP là khá linh hoạt. Đầu tiên nó tìm kiếm một đối tượng đang tồn tại đang sử dụng thuộc tính id và scope. Nếu một đối tượng không được tìm thấy, thì nó cố gắng tạo đối tượng đã xác định đó.

Một **JavaBean** là một **lớp Java** được xây dựng một cách đặc biệt trong Java và được mã hóa theo JavaBeans API Specifications.

Sau đây là các đặc trưng duy nhất giúp phân biệt một JavaBean với các lớp khác trong Java:

Nó cung cấp một constructor **mặc định**, không có tham số.

Nó có thể xếp thứ tự và triển khai **Serializable** interface.

Nó có thể có một số đặc tính mà có thể được đọc và được viết.

Nó có thể có một số phương thức "getter" và "setter" cho các đặc tính.

Cách đơn giản nhất để tải một Bean như sau:

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

Trong đó scope="page" là mặc định, ngoài ra "application", "session" hoặc "request"

`<jsp:useBean id="students" class = "bean.StudentsBean"/>`

Tương đương

`<% bean.StudentsBean students = new bean.StudentsBean(); %>`

Khi lớp Bean đã được tải, có thể sử dụng các `<jsp:setProperty>` action và `<jsp:getProperty>` action để sửa đổi và thu nhận các đặc tính của Bean.

`<jsp:getProperty name="students" property="firstName"/>`

Tương đương `<%= students.getFirstName() %>`

`<jsp:getProperty name="students" property="*" />`

`<jsp:setProperty name="students" property="firstName" value="Ha"/>`

Tương đương `<%= students.setFirstName("Ha") %>`

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

Bài 6:

Làm lại bài tính số học dùng java Bean (JavaBeanEx)

A Simple Calculator

Enter Operand1:

Enter Operand2:

Select Operation:

Bước 1: tạo java bean (lớp – java)

Bước 2: jsp (form) – viết ra kết quả

Bài 7: (về nhà)

1. Cho Project: Ex-DrawCircle với các file có sẵn. file index.html khi chạy được

Enter the number of Circles:

Tạo servlet có tên là CircleServlet và url-pattern /circle, servlet sinh ngẫu nhiên ra các đường tròn lưu trữ trong ServletScope với tên “cirs” sao đó được hiển thị bởi /cirView

2. Cho Project: Ex-DrawRect với các file có sẵn. file index.html khi chạy được

Enter the number of Rectangles:

Tạo servlet có tên là RectServlet và url-pattern /rect, servlet sinh ngẫu nhiên ra các đường chữ nhật lưu trữ trong ServletScope với tên “rects” sao đó được hiển thị bởi /rect.jsp

3. Làm bài Q1_XXX

/tinh.html

/tinh.jsp

/tinh

enter money (VND):

☒ to USD ☐ to Yen

Change

init parameter rateUSD (23000) and rateJPY (121) in context parameter. if input not number print "enter digits", if number print change money to USD or JPY

enter money (VND):

change to USD

change to JPY

init parameter rateUSD (23000) and rateJPY (121) in context parameter. if input not number print "enter digits", if number print change money to USD or JPY