

CHUYÊN ĐỀ JAVA

GIỚI THIỆU JSP (JAVA SERVER PAGE)

Nguyễn Hoàng Anh – nhanh@fit.hcmus.edu.vn

1

Nội dung trình bày

- Thuận lợi và bất lợi của Servlet
- Giới thiệu JSP
- Chu kỳ sống của trang JSP
- Cú pháp cơ bản

2

Servlet

- Thuận lợi
 - Đọc dữ liệu từ Form
 - Đọc các HTTP Request Header
 - Gán HTTP Status Code và Response Header
 - Sử dụng Cookie và Session
 - Chia sẻ dữ liệu giữa các Servlet
 - Xử lý cơ sở dữ liệu, ...
- Bất lợi
 - Sử dụng câu lệnh println để phát sinh HTML

3

Servlet

- Bạn có nghĩ bạn sẽ tạo được giao diện web đẹp với Servlet?
- Bạn có thể đọc được một trang web được tạo ra với Servlet?
- Bạn sẽ bảo trì các giao diện web này như thế nào?

→ Servlet gây khó khăn rất lớn cho lập trình viên khi phải tạo ra giao diện web.

4

Servlet

- Servlet rất mạnh về xử lý và điều phối, nhưng Servlet lại rất yếu về tạo giao diện
- Khắc phục điểm yếu của Servlet như thế nào?

→ Sử dụng công nghệ JSP

5

JSP

- Ý tưởng
 - Thiết kế các trang web sử dụng HTML chuẩn
 - Vị trí nào cần tạo ra nội dung động chỉ cần chèn các thẻ Java vào bên trong HTML.
 - Toàn bộ trang JSP được thông dịch sang Servlet (một lần) và Servlet được thực thi khi yêu cầu của client gửi đến
- Dễ dàng hơn rất nhiều cho lập trình viên

6

JSP

```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <html>
3  <head>
4      <meta http-equiv="Content-Type"
5          content="text/html; charset=UTF-8">
6      <title>JSP Page</title>
7  </head>
8  <body>
9      <h1> Xin chào:
10         <%=request.getParameter("tenNguoiDung")%>
11     </h1>
12 </body>
13 </html>

```

7

JSP

- Thuận lợi
 - Thuận tiện khi tạo ra trang web HTML
 - Dễ đọc và dễ bảo trì trang web HTML
- Có nhiều công cụ hỗ trợ thiết kế HTML
 - Dream Weaver
 - Adode GoLive,...
- Phân cách thiết kế web và xử lý mã nguồn java
 - Đội ngũ thiết kế HTML chuyên nghiệp hơn lập trình viên java

8

8

JSP

- Thuận lợi hơn Servlet
 - Thuận tiện trong việc tạo ra trang web HTML
 - Sử dụng các công cụ thiết kế như DreamWeaver
 - Phân cách xử lý và giao diện
- JSP ra đời để thay thế Servlet?
 - Lập trình viên JSP rất cần kiến thức về lập trình web với Servlet

9

Xây dựng web với java

- Servlet mạnh về xử lý nghiệp vụ và điều phối nhưng lại rất yếu về hiển thị
- JSP mạnh về xử lý hiển thị nhưng lại yếu về xử lý nghiệp vụ và điều phối
- Vậy chọn công nghệ nào? JSP hay Servlet? Tại sao?

10

Xây dựng web với java

- Thường thì chúng ta sẽ **kết hợp** sức mạnh của **Servlet** và **JSP** vào mô hình **MVC** (Model-View-Controller)
 - Các **Servlet** đóng vai trò làm **Controller**
 - Các trang **JSP** đóng vai trò làm **View**

11

11

Thiết lập môi trường khi sử dụng JSP

- Thiết lập CLASSPATH: **Không**
- Biên dịch: **Không**
- Sử dụng các Packaged để tránh đụng độ tên: **Không**
- Trang JSP phải để vào thư mục đặc biệt: **Không**
 - Để vào Web Pages trong netbeans, giống với HTML, CSS, JavaScript, GIF, JPEG, ...
- Sử dụng các URL đặc biệt cho trang JSP: **Không**
 - Giống như URL của trang HTML (ngoại trừ phần mở rộng của trang)

12

12

JSP

```

1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type"
5   content="text/html; charset=UTF-8">
6 <title>JSP Page</title>
7 </head>
8 <body>
9 <h2>JSP Expressions</h2>
10 <ul>
11 <li>current time: <%= new java.util.Date() %>
12 <li>Server: <%= application.getServerInfo() %>
13 <li>Session ID: <%= session.getId() %>
14 <li>the <code>testparam</code> form parameter:
15   <%= request.getParameter("testparam") %>
16 </ul>
17 </body>
18 </html>

```

13

JSP

The screenshot shows a web browser window titled "JSP Page - Mozilla Firefox". The address bar displays the URL "http://localhost:8080/jsp-scripting/expressions.jsp". The page content, titled "JSP Expressions", lists the following information:

- current time: Mon Oct 04 19:10:13 PDT 2010
- Server: GlassFish Server Open Source Edition 3.0.1
- Session ID: a2681d730e1c0816f23527cd00d
- the testparam form parameter: null

In the background, a file explorer window shows the project structure with "expressions.jsp" highlighted under the "WEB-INF" directory.

14

Chu kỳ sống của trang JSP

	Request #1	Request #2	Request #3	Request #4	Request #5	Request #6
JSP được thông dịch sang Servlet	Yes	No	No	No	Yes	No
Servlet được biên dịch	Yes	No	No	No	Yes	No
Khởi tạo và nạp thể hiện của Servlet lên bộ nhớ server	Yes	No	Yes	No	Yes	No
Phương thức init được gọi	Yes	No	Yes	No	Yes	No
Phương thức doGet, (hoặc doPost, ...) được gọi	Yes	Yes	Yes	Yes	Yes	Yes

15

15

Cú pháp cơ bản

Các thẻ HTML	
Mô tả	Khi chuyển từ server đến client, nội dung HTML không thay đổi
Ví dụ	<code><h1> Xin chào </h1></code>
Ghi chú trong HTML	
Mô tả	Thẻ ghi chú được gửi đến client nhưng không được hiển thị
Ví dụ	<code><!-- Nội dung ghi chú --></code>
Ghi chú trong JSP	
Mô tả	Thẻ ghi chú của lập trình viên không được gửi đến client
Ví dụ	<code><%-- Nội dung ghi chú --></code>

16

16

Cú pháp cơ bản

JSP Expression	
Mô tả	Biểu thức được tính toán và gửi kết quả về khi client request
Ví dụ	<code><%= Giá trị %></code>
JSP Scriptlet	
Mô tả	Thực thi các câu lệnh khi client request
Ví dụ	<code><% Câu lệnh hoặc các câu lệnh%></code>
JSP Declaration	
Mô tả	Khai báo thuộc tính và định nghĩa phương thức trong trang
Ví dụ	<code><%! Khai báo thuộc tính %></code> <code><%! Định nghĩa phương thức %></code>

17

17

Cú pháp cơ bản

JSP Directive	
Mô tả	<ul style="list-style-type: none"> - Thông tin về cấu trúc trang : <code>page</code> - Chèn trang khác vào : <code>include</code> - Các thẻ do người dùng định nghĩa : <code>taglib</code>
Ví dụ	<code><%@ directive att="val" %></code>
JSP Action	
Mô tả	Thực thi Action khi client request
Ví dụ	<code><jsp:forward page="success.jsp" /></code>
JSP Expression Language Element	
Mô tả	Viết tắt của JSP Expression
Ví dụ	<code>\${biểu thức}</code>

18

18

Cú pháp cơ bản

Custom Tag	
Mô tả	Thẻ do người dùng định nghĩa
Ví dụ	<pre><prefix:name> . . . </prefix:name></pre>

19

19

Các đối tượng được tạo sẵn trong trang JSP

- **application** : javax.servlet.ServletContext
- **config** : javax.servlet.ServletConfig
- **exception** : java.lang.Throwable
- **out** : javax.servlet.jsp.JspWriter
- **page** : java.lang.Object
- **PageContext** : javax.servlet.jsp.PageContext
- **request** : javax.servlet.ServletRequest
- **response** : javax.servlet.ServletResponse
- **session** : javax.servlet.http.HttpSession

20

20

10 website phổ biến nhất năm 2010 - Alexa.com

- | | |
|----------------------------|----------------------------|
| 1. Google | 6. Baidu |
| Java (Web), C++ (Indexing) | Unknown |
| 2. Facebook | 7. Wikipedia |
| PHP | PHP |
| 3. YouTube | 8. Blogger |
| Flash, Python, Java | Java |
| 4. Yahoo | 9. MSN |
| PHP, Java | .NET |
| 5. Microsoft Live.com | 10. Twitter |
| .NET | Ruby on Rails, Scala, Java |

21

21

10 ngôn ngữ phổ biến nhất thế giới năm 2010

1. Java
2. C
3. C++
4. PHP
5. Visual Basic
6. C#
7. Python
8. Objective C
9. Perl
10. Ruby

Nguồn: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

22

22

Tham khảo

- Marty Hall, Larry Brown, Core Servlets and JavaServer Pages™: Volume 1: Core Technologies, 2nd Edition (2003)
- <http://courses.coreservlets.com/Course-Materials/csajsp2.html> (2010)

23

23



HỎI VÀ ĐÁP

24

24