

BỘ MÔN CNPM - KHOA CNTT - ĐH KHTN

J2EE PROGRAMMING TRAINING COURSE 2012 - 2.0.1

GIỚI THIỆU JSP

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

Nội dung trình bày

- Thuận lợi và bất lợi của Servlet
- Giới thiệu JSP
- JSP là Servlet
- Cú pháp cơ bản

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

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.

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

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**

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>
```

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

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

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?

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**

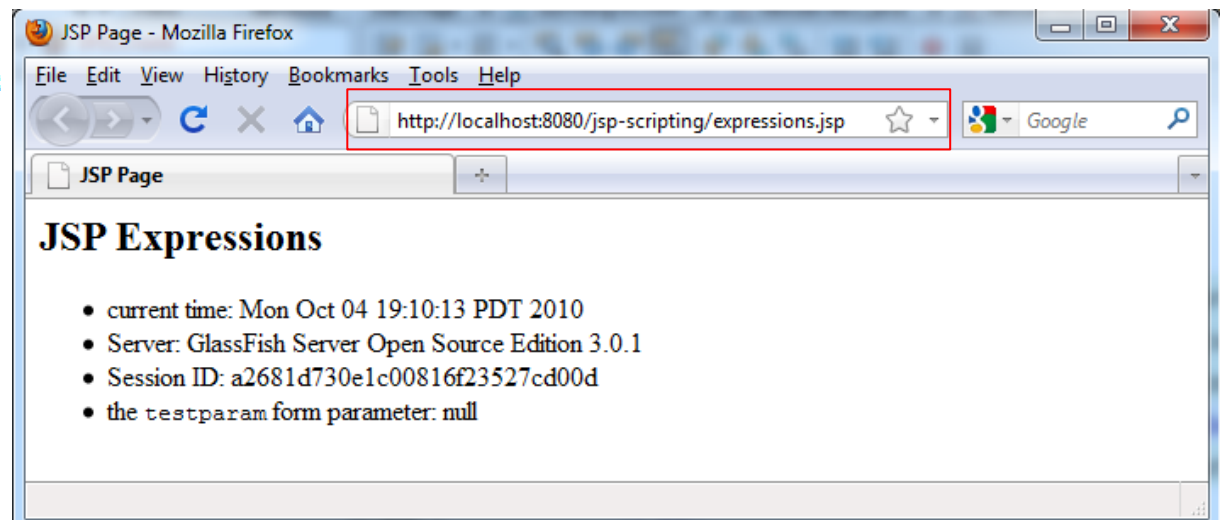
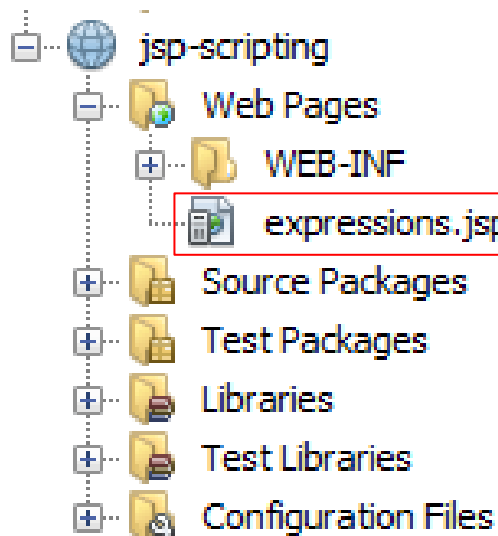
Thiết lập môi trường khi sử dụng JSP

- 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)

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>
```

JSP



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>

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>

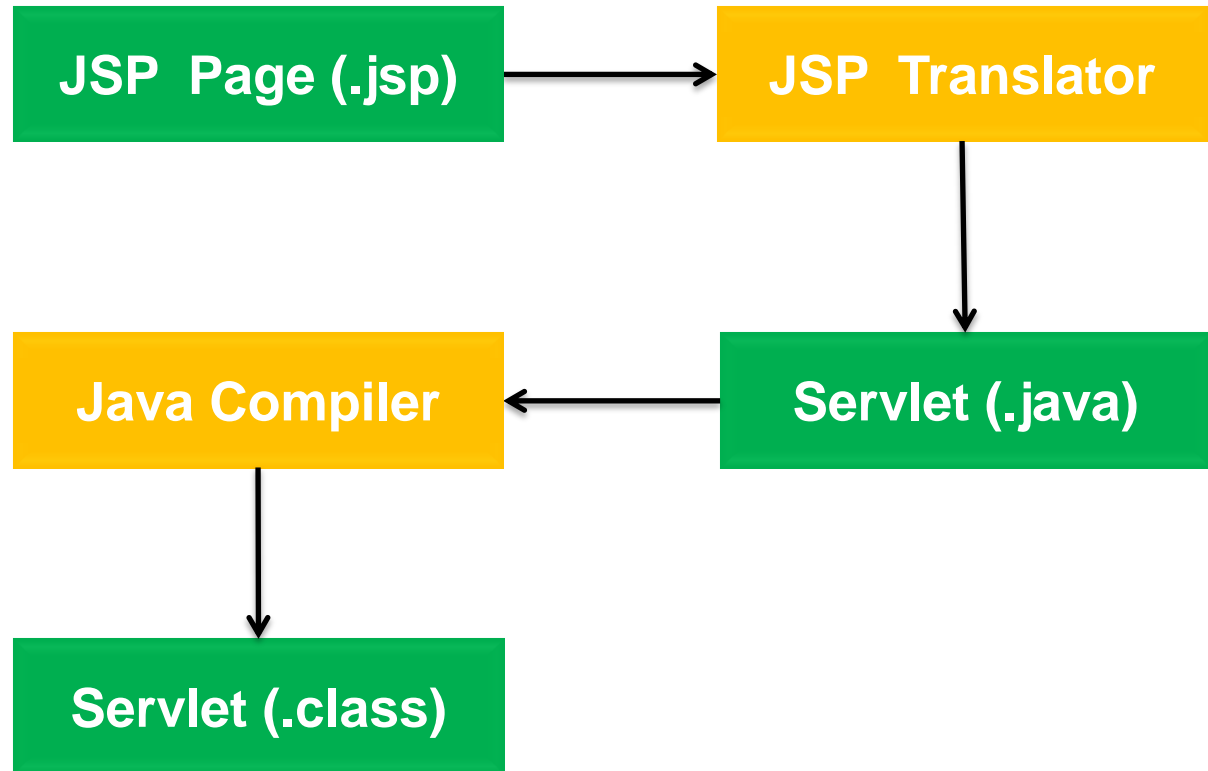
Cú pháp cơ bản

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

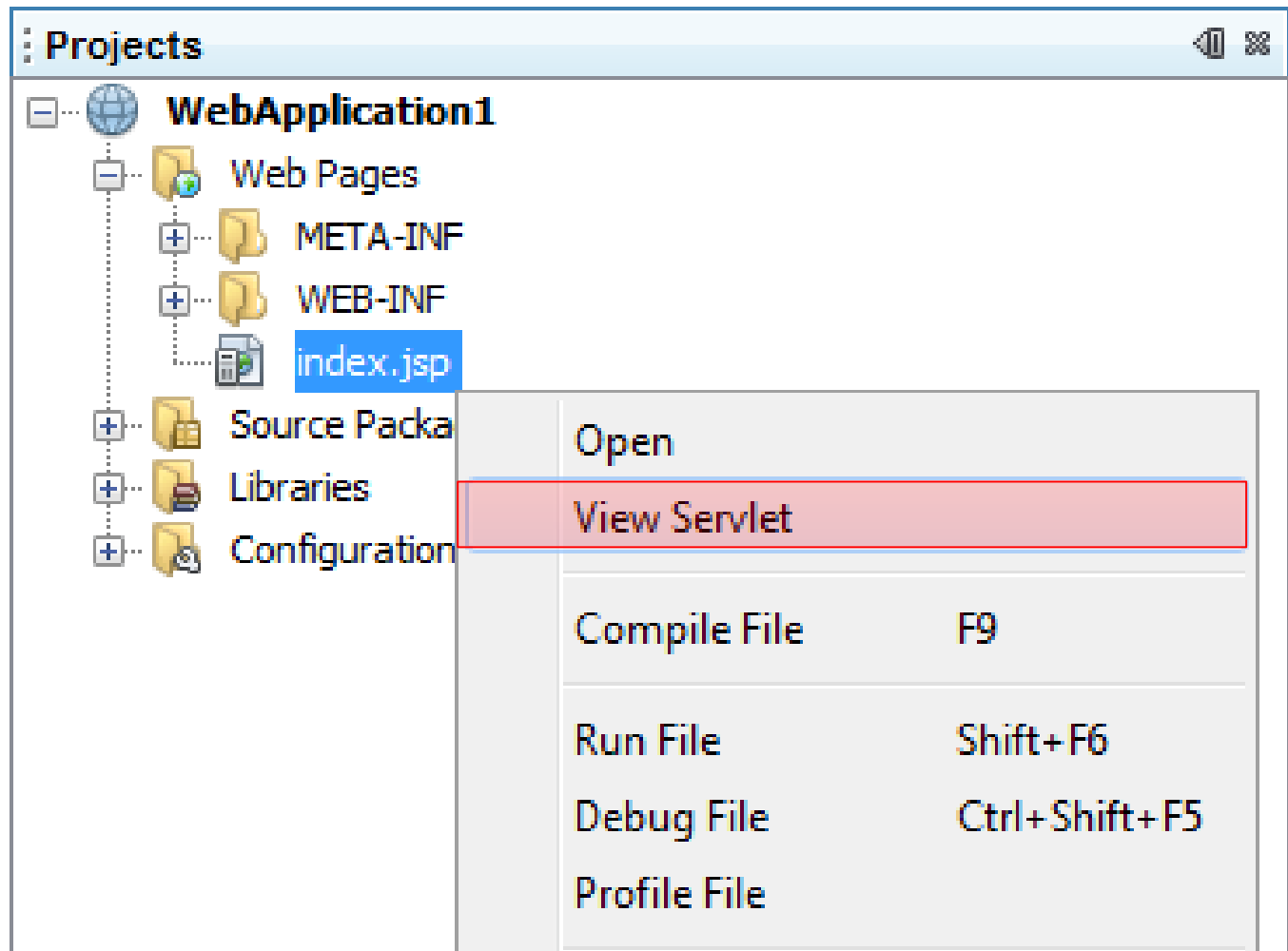
Cú pháp cơ bản

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

JSP là Servlet



JSP là Servlet



index.jsp

```
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html;
12             charset=UTF-8">
13         <title>JSP Page</title>
14     </head>
15     <body>
16         <h1>Hello World!</h1>
17         <%=new java.util.Date()%>
18     </body>
19 </html>
```

index_jsp.java - _jspService

```
1 public final class index_jsp extends HttpJspBase, JspSourceDependent {
2
3     public void _jspService(final HttpServletRequest request,
4                             final HttpServletResponse response)
5         throws IOException, ServletException {
6         final PageContext pageContext;
7         HttpSession session = null;
8         final ServletContext application;
9         final ServletConfig config;
10        JspWriter out = null;
11        final java.lang.Object page = this;
12        JspWriter _jspx_out = null;
13        PageContext _jspx_page_context = null;
14        ...
15    }
16 }
17
```

index_jsp.java - _jspService

```
1    ...
2    response.setContentType("text/html;charset=UTF-8");
3    pageContext = _jspxFactory.getPageContext(this, request, response,
4                                              null, true, 8192, true);
5    _jspx_page_context = pageContext;
6    application = pageContext.getServletContext();
7    config = pageContext.getServletConfig();
8    session = pageContext.getSession();
9    out = pageContext.getOut();
10   ...
11
```

index_jsp.java - _jspService

```
54 out.write("\n");
55 out.write("\n");
56 out.write("\n");
57 out.write("<!DOCTYPE html>\n");
58 out.write("<html>\n");
59 out.write("    <head>\n");
60 out.write("        <meta http-equiv=\"Content-Type\" content=\"text/html;\n");
61 out.write("            charset=UTF-8\">\n");
62 out.write("        <title>JSP Page</title>\n");
63 out.write("    </head>\n");
64 out.write("    <body>\n");
65 out.write("        <h1>Hello World!</h1>\n");
66 out.write("    ");
67 out.print(new java.util.Date());
68 out.write("\n");
69 out.write("    </body>\n");
70 out.write("</html>\n");
```


Các đối tượng được tạo sẵn trong trang JSP

- **application** :ServletContext
- **config** :ServletConfig
- **exception** :Throwable
- **out** :JspWriter
- **page** : Object
- **PageContext** : PageContext
- **request** : HttpServletRequest
- **response** : HttpServletResponse
- **session** : HttpSession

TIOBE Programming Community Index for September 2011

Position Sep 2011	Position Sep 2010	Delta in Position	Programming Language	Ratings Sep 2011	Delta Sep 2010	Status
1	1	=	Java	18.761%	+0.85%	A
2	2	=	C	18.002%	+0.86%	A
3	3	=	C++	8.849%	-0.96%	A
4	6	↑↑	C#	6.819%	+1.80%	A
5	4	↓	PHP	6.596%	-1.77%	A
6	8	↑↑	Objective-C	6.158%	+2.79%	A
7	5	↓↓	(Visual) Basic	4.420%	-1.38%	A
8	7	↓	Python	4.000%	-0.58%	A
9	9	=	Perl	2.472%	+0.03%	A
10	11	↑	JavaScript	1.469%	-0.20%	A

Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập bài giảng môn chuyên đề Java, 2011
- Marty Hall, Tập bài giảng J2EE, 2010



HỎI VÀ ĐÁP