**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**BÁO CÁO**

**BÀI TẬP NHÓM**

**NHÓM: 24**

**ĐỀ TÀI : LẬP TRÌNH WEB VỚI JSP**

**NGƯỜI THỰC HIỆN: NGUYỄN HUY TRƯỜNG**

**PHẠM HỒNG THÁI**

**NGUYỄN KHẢ KHIÊM**

**HOÀNG THÁI SƠN**

**NGUYỄN XUÂN KHẢI**

**NGƯỜI HƯỚNG DẪN: NGUYỄN MẠNH SƠN**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(CỦA GIÁO VIÊN HƯỚNG DẪN)**

Điểm: (bằng chữ:……………………)

*Hà nội, ngày…tháng…năm 2018*

***CÁN BỘ - GIÁO VIÊN HƯỚNG DẪN***

*(Ký, ghi rõ họ tên)*

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(CỦA GIÁO VIÊN PHẢN BIỆN)**

Điểm: (bằng chữ:……………………)

*Hà nội, ngày…tháng…năm 2018*

***CÁN BỘ - GIÁO VIÊN PHẢN BIỆN***

*(Ký, ghi rõ họ tên)*

Mục lục

[I. Giới thiệu về JSP 5](#_Toc57240110)

[1. Giới thiệu chung 5](#_Toc57240111)

[2. Lợi thế của JSP so với servlet 6](#_Toc57240112)

[3. Vận hành và vòng đời của 1 trang JSP 6](#_Toc57240113)

[I. Sự khác nhau giữa JSP và HTML 10](#_Toc57240114)

[1. Sự khác biệt giữa JSP và HTML 10](#_Toc57240115)

[II. Các đối tượng ngầm của JSP và thư viện JSTL 11](#_Toc57240116)

[1. JSP Implicit Objects – Yêu cầu và phản hồi 11](#_Toc57240117)

[1.1. Request 11](#_Toc57240118)

[1.2. Response 12](#_Toc57240119)

[2. Thư viện JSTL 12](#_Toc57240120)

[2.1. Thẻ ***<c:if>*** 12](#_Toc57240121)

[***2.2.*** ***<c:forEach>*** 13](#_Toc57240122)

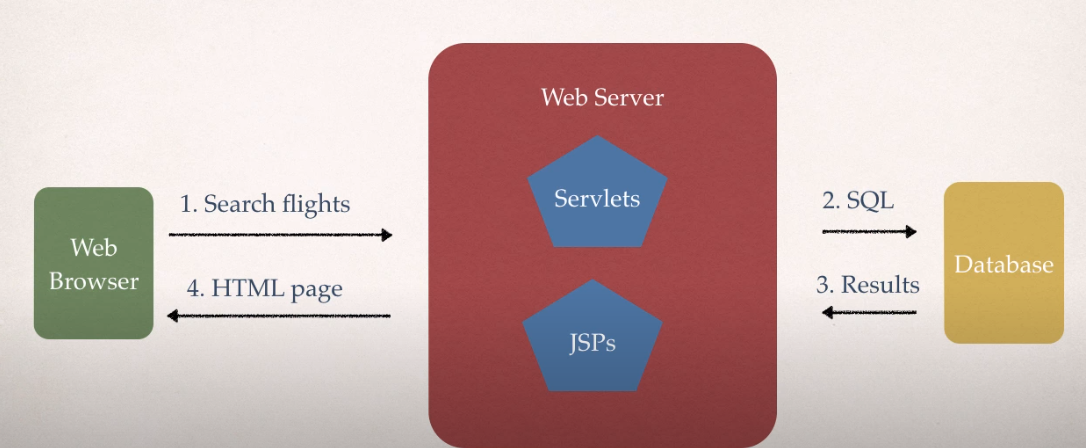
[III. Code demo 13](#_Toc57240123)

[Tài liệu tham khảo 14](#_Toc57240124)

# Giới thiệu về JSP

1. Giới thiệu chung

JSP (viết tắt của tiếng Anh JavaServer Pages) còn được biết đến với một cái tên khác là Java Scripting Preprocessor - tạm dịch là "Bộ tiền xử lý văn lệnh Java", là một công nghệ Java giúp các nhà phát triển nội dung HTML, XML hay một số định dạng khác của trang web một cách năng động, hồi âm yêu cầu của trình khách. Công nghệ này cho phép người ta nhúng mã Java và một số hành động xử lý đã được định trước (pre-defined actions) vào trong nội dung tĩnh của trang.



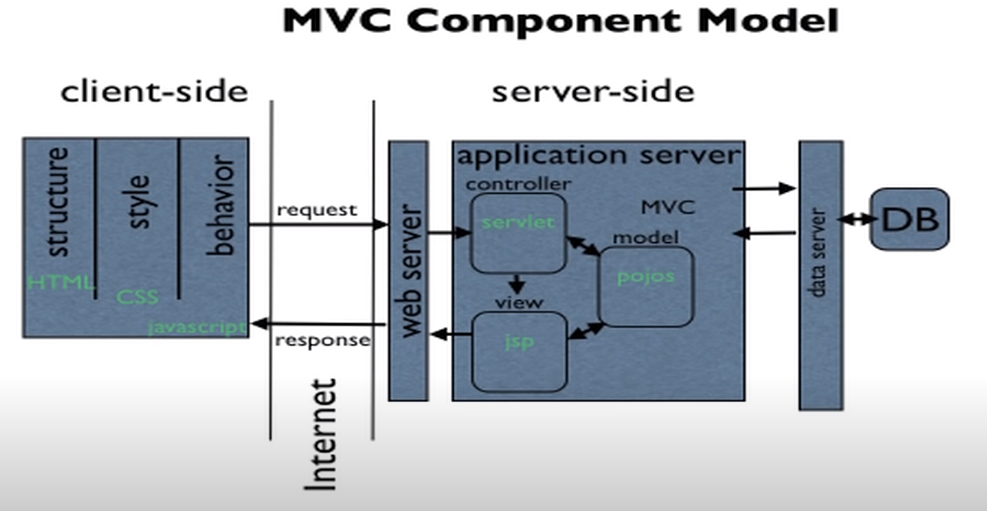
Hình 1.1. JSP và Web Server

JSP là một công nghệ phía máy chủ, được sử dụng để tạo ứng dụng web hoặc tạo nội dung các trang web động và độc lập với nền tảng. JSP được sử dụng để chèn mã Java vào các trang HTML. Nó là một phiên bản nâng cao của Servlet.

**JSP được biên dịch nội bộ thành Java Servlet**

Có nghĩa là, bất cứ điều gì có thể được thực hiện bằng cách sử dụng JSP cũng có thể được thực hiện bằng cách sử dụng Java servlet. Tuy nhiên, điều quan trọng cần lưu ý là các servlet và JSP là các công nghệ bổ sung , KHÔNG thay thế nhau. Servlet có thể được xem như " HTML bên trong Java ", tốt hơn cho việc triển khai logic nghiệp vụ - vì nó là Java chiếm ưu thế. Mặt khác, JSP là " Java bên trong HTML ", ưu việt hơn trong việc tạo bản trình bày - vì nó chiếm ưu thế trong HTML. Trong một ứng dụng Model-View-Control (MVC) điển hình , các servlet thường được sử dụng cho Controller (C), liên quan đến logic lập trình phức tạp. JSP thường được sử dụng cho View (V), chủ yếu đề cập đến việc trình bày. Model (M) thường được triển khai bằng JavaBean hoặc EJB.(Hình 1.2)

Trước khi hiển thị ra trình duyệt, tập tin JSP phải được biên dịch thành Servlet, dùng bộ biên dịch JSP (JSP compiler). Bộ biên dịch JSP có thể tạo servlet thành mã nguồn Java trước, rồi biên dịch mã nguồn ra tập tin .class dùng bộ biên dịch Java, hoặc có thể trực tiếp tạo mã byte code cho servlet từ trang JSP.



1.2. JSP và view trong MVC

1. Lợi thế của JSP so với servlet

* JSP là một tính năng được phát triển bởi Sun microsystems để cải tiến cho Servlets, che đậy tất cả các lỗ hổng của Servlets.
* Nếu bạn chưa biết, Servlets được sử dụng để chứa cả code logic và code giao diện. Nhưng trong JSP, hai thành phần này được tách ra. (Tách biệt logic nghiệp vụ và phần trình bày để dễ quản lý). Trong JSP, chúng ta định nghĩa thẻ bằng "<% %>".
* JSP rất dễ bảo trì và khi chạy không cần biên dịch lại hoặc triển khai lại.
* Sử dụng JSP, chúng ta có thể triển khai API Java theo mặc định vào chương trình web.
* Và không chỉ API Java, JSP còn hoạt động với các thẻ HTML và XML, điều mà trước đây không thể thực hiện được.

1. Vận hành và vòng đời của 1 trang JSP

***Để hiểu rõ hơn về hoạt động của 1 file JSP ta có 1 ví dụ về JSP.***

1. <html>
2. <head> <title> JSP đầu tiên </title> </head>
3. <body> <%
4. double num = Math.random ();
5. if (num> 0.95) {
6. %>
7. <h2> Bạn sẽ có một ngày may mắn! </h2> <p> ( <% = num%> ) </p>
8. <%
9. } else {
10. %>
11. <h2> Chà , cuộc sống vẫn tiếp diễn ... </h2> <p> ( <% = num%> ) </p>
12. <%
13. }
14. %>
15. <a href=" <%= request.getRequestURI() %> "> <h3 > Thử lại </h3> </a>
16. </body>
17. </html>

Hãy bắt đầu với một ví dụ JSP đơn giản. Chúng ta sẽ sử dụng ứng dụng web có tên " hello" mà chúng ta đã tạo trong bài tập trước đó của chúng tôi. Sử dụng trình chỉnh sửa văn bản lập trình để nhập các mã HTML / JSP sau và lưu dưới dạng " first.jsp" (loại tệp " .jsp" là bắt buộc) trong thư mục chính của webapp (ngữ cảnh web) của bạn (tức là " webapps\hello"). Để thực thi tập lệnh JSP: Chỉ cần khởi động máy chủ Tomcat và sử dụng trình duyệt để cấp URL để duyệt trang JSP (tức là <http://localhost:8080/hello/first.jsp>).

**Giải thích:**

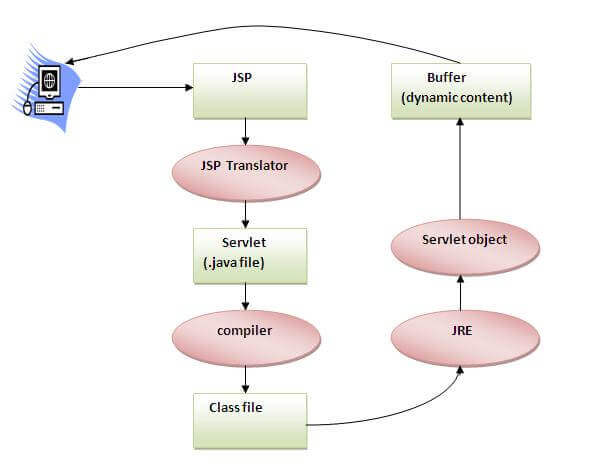
* Tập lệnh JSP là một trang HTML thông thường chứa các chương trình Java. Nhớ lại rằng JSP là " Java bên trong HTML " (trong khi servlet là " HTML bên trong Java "). Các câu lệnh Java được bao bọc bởi <% ... %>(được gọi là JSP scriptlet ) hoặc <%= ... %>(được gọi là biểu thức JSP ).
* JSP Scriptlet <% ... %> được sử dụng để bao gồm các câu lệnh Java.
* JSP Expression <%= ... %> được sử dụng để đánh giá một biểu thức Java đơn và hiển thị kết quả của nó.
* Phương thức request.getRequestURI()được sử dụng để truy xuất URL của trang hiện tại . Điều này được sử dụng trong thẻ liên kết để làm mới trang để lấy một số ngẫu nhiên khác.

Khi một JSP được truy cập lần đầu tiên, Tomcat chuyển đổi JSP thành một servlet, biên dịch servlet và thực thi servlet. Kiểm tra servlet được tạo cho " first.jsp" và nghiên cứu chuyển đổi JSP-sang-servlet. Tìm " work\Catalina\localhost\hello" cho " first\_jsp.java" của Tomcat .

Qua đó nhìn chung, các trang JSP tuân theo các giai đoạn sau:

* Translation
* Complilation
* Classloading - Tải lớp (trình nạp lớp tải tệp lớp)
* Instantiation - Thuyết minh (Đối tượng của Servlet được tạo được tạo).
* Initialization - Khởi tạo (vùng chứa gọi phương thức jspInit ()).
* Request processing - Yêu cầu xử lý (vùng chứa gọi phương thức \_jspService ()).
* Destroy - Hủy bỏ (vùng chứa gọi phương thức jspDestroy ()).

Chúng ta hãy xem quy trình thực thi của một chương trình JSP. Khi chúng ta triển khai các chương trình JSP, trách nhiệm của JSP Container là chuyển đổi các chương trình JSP thành mã Servlet tương ứng . Mỗi vùng chứa JSP đều có một Trình biên dịch JSP. Trình biên dịch JSP có trách nhiệm chuyển đổi / chạy chương trình JSP. Hình sau cho thấy các bước sẽ được thực hiện khi chúng ta triển khai JSP trên đầu máy chủ.



Quy trình thực thi của 1 file hello.jsp :

* Máy khách gửi yêu cầu đến máy chủ.
* Trình biên dịch JSP tạo (chuyển đổi JSP thành servlet) mã servlet của  one.jsp và đặt nó vào một tệp có tên hello.jsp.java.
* Bây giờ Trình biên dịch Java đã hình thành. JavaC chuyển đổi mã Java thành mã Byte, tức là nó tạo ra tệp .class .
* Bây giờ máy chủ tạo đối tượng servlet và xử lý yêu cầu của khách hàng.

Máy chủ sẽ thực hiện tất cả các bước trên khi nhận được yêu cầu lần đầu tiên. Khi máy khách gửi cùng một yêu cầu lần thứ hai, máy chủ chỉ thực thi đối tượng servlet của chương trình JSP tương ứng.

Một trang JSP bao gồm :

* **Thẻ root**: Thẻ này sẽ chứa các thuộc tính, thông tin của trang JSP.
* **Comment**: Cũng như trang HTML, trong JSP, chúng ta có thể comment với kí hiệu *<! your comment ->*
* **Declaration**: khai báo biến hoặc phương thức của java trong trang JSP.
* **Expression**: Được sử dụng để chèn 1 giá trị vào trang một cách trực tiếp
* **Thẻ biểu thức JSP**: Được sử dụng để đánh giá một biểu thức và định hướng các output đến một trình duyệt web phù hợp. Cú pháp khai báo là: *<%= your code %>*
* **Scriptlet Tag:** cho phép bạn viết mã java trong trang JSP. Cú pháp

*<% <i> mã java </ i>%>*

# Sự khác nhau giữa JSP và HTML

HTML(Hypertext markup language) là ngôn ngữ đánh dấu Siêu văn bản, ngôn ngữ đánh dấu tiêu chuẩn cho các tài liệu được thiết kế để hiển thị và xem trực tuyến trong trình duyệt cũng giúp tạo cấu trúc của trang web. bởi vì nó là một ngôn ngữ đánh dấu, nó bao gồm nhiều thẻ. Có các thẻ để hiển thị văn bản, bảng, danh sách có thứ tự và danh sách không có thứ tự, v.v. Có hai phần chính trong trang HTML: phần đầu và phần nội dung. Dữ liệu mô tả trang còn được gọi là siêu dữ liệu nằm bên trong phần đầu trong khi phần thân bao gồm tất cả các thẻ cần thiết để thể hiện nội dung hiển thị của trang web.

1. Sự khác biệt giữa JSP và HTML

|  |  |  |
| --- | --- | --- |
| STT | JSP | HTML |
| 1 | JSP được cung cấp bởi Sun Micro System. | HTML được tạo ra bởi Tim Berners-Lee vào năm 1991. |
| 2 | JSP cung cấp một giao diện động cho dữ liệu thay đổi liên tục và nó cũng tự động gọi các hành động của máy chủ. | HTML cung cấp một phương tiện để đưa ra một tài khoản chi tiết về cấu trúc của thông tin dựa trên văn bản trong một tài liệu. |
| 3 | JSP chỉ tạo các trang web động. | Trong khi Html chỉ tạo ra các trang web tĩnh. |
| 4 | Cần có JSP container để thực thi mã Jsp. | Cần có Trình thông dịch Html để thực thi các mã này. |
| 5 | JSP cấp quyền đặt mã java bên trong các trang JSP. | Trong HTML, nó không cho phép đặt mã java bên trong các trang Html. |
| 6 | JSP là một công nghệ được sử dụng để tạo các ứng dụng web động. | HTML là một ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để tạo cấu trúc của các trang web. |
| 7 | JSP cho phép đặt thẻ tùy chỉnh hoặc thẻ của bên thứ ba. | Nó không cho phép đặt thẻ tùy chỉnh hoặc thẻ của bên thứ ba. |
| 8 | JSP chạy thẳng trên Máy chủ Web và JVM cục bộ. | HTML chạy trong Trình duyệt web. |
| 9 | JSP được gọi là ngôn ngữ kịch bản phía máy chủ. | HTML được gọi là ngôn ngữ kịch bản phía máy khách. |
| 10 | JSP cần một khoảng thời gian để tải vì nó cần tương tác với Máy chủ Web. | Vì HTML chạy trên máy cục bộ, nó được tải rất nhanh hơn. |

1. Lợi thế của JSP so với HTML

Khi sử dụng JSP, người dùng không cần viết code HTML và Java riêng biệt mà tất cả ở trong file .jsp, điều này giúp cho việc quản lí và thao tác code dễ dàng hơn. Giúp spilt code, tách riêng các phần theo chức năng

JSP là biên dịch động . Điều này có nghĩa là, bất cứ khi nào tệp .jsp được sửa đổi trong môi trường phát triển, nó phải được biên dịch trước khi được gọi trong máy khách.

# Các đối tượng ngầm của JSP và thư viện JSTL

1. JSP Implicit Objects – Yêu cầu và phản hồi

JSP cung cấp tổng số 9 đối tượng Implicite như sau:

1. Request
2. Response
3. Config
4. Application
5. Session
6. Page context
7. Page object
8. Exception
9. Out

Chúng ta sẽ đi tìm hiểu về 2 đối tượng chính yêu cầu và request và response

* 1. Request

Yêu cầu JSP là đối tượng ngầm được cung cấp bởi HttpServletRequest. Trong servlet, trước tiên chúng ta phải nhập javax.servlet.http.HttpServletRequest sau đó chúng ta phải tạo đối tượng của nó để lấy đầu vào từ bất kỳ biểu mẫu HTML nào dưới dạng

import javax.servlet.http.HttpServletRequest;

public class LoginServlet extends HttpServlet

{

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

HttpSession session = request.getSession()

}

}

Trong JSP, đối tượng yêu cầu được định nghĩa ngầm để bạn không phải tạo bất kỳ đối tượng nào. Đối tượng yêu cầu JSP được tạo bởi vùng chứa web cho mỗi yêu cầu của khách hàng. Nó được sử dụng để lấy giá trị tham số, tên máy chủ, cổng máy chủ, v.v.

* 1. Response

Đây là đối tượng HttpServletResponse được liên kết với phản hồi cho máy khách. Đối tượng phản hồi cũng xác định các giao diện xử lý việc tạo tiêu đề HTTP mới. Thông qua đối tượng này, lập trình viên JSP có thể thêm cookie mới hoặc tem ngày, mã trạng thái HTTP, v.v.

Trong JSP, đối tượng phản hồi được định nghĩa ngầm nên bạn không phải tạo bất kỳ đối tượng nào. Đối tượng phản hồi JSP được tạo bởi vùng chứa web cho mỗi yêu cầu của khách hàng. Về cơ bản, nó được sử dụng để chuyển hướng đến bất kỳ tài nguyên nào khác.

1. Thư viện JSTL

**JSP Standard Tag Library (JSTL)** là thư viện thẻ chuẩn cung cấp các thẻ để kiểm soát hành vi trang, lặp đi lặp lại và các lệnh điều khiển, các thẻ quốc tế hóa, và các thẻ SQL.

**JSTL** là một phần của **Java EE API** và trong hầu hết các container servlet.

* 1. Thẻ ***<c:if>***

Đánh giá một biểu thức và chỉ hiển thị nội dung của nó nếu biểu thức đánh giá là true.

Ví dụ: Kiểm tra nếu carts khác 0 thì in ra.

<c:if test="${not empty carts && sessionScope.carts.size()!= 0}">

<div class="cart-alert">

<span>${sessionScope.carts.size()}</span>

</div>

</c:if>

* 1. ***<c:forEach>***

Thẻ này tồn tại như một giải pháp thay thế tốt cho việc nhúng Java vòng lặp for, while hoặc do- while thông qua một scriptlet

Ví dụ: Hiển thị ra danh sách sản phẩm

<c:forEach var="item" items="${items}">

<div class="product-item productG">

<div class="product-name">

<a href="/san-pham?id=${item.id}">${item.name}</a>

</div>

</div>

</c:forEach>

# Code demo

* Tên : Sảm phẩm web bán mỹ phẩm
* Chức năng :

+ Client: Hiển thị, giới thiệu sản phẩm, thông tin giỏ hàng, thông tin thanh toán…

+ Admin: Thêm, sửa, xóa, quản lý sản phẩm…

* Công nghệ sửa dụng :

+ Mô hình MVC

+ Java Spring boot, JSP

+ Html/Css/Js

* Link code github: [https://github.com/truong02bp/dulich-spring-boot](https://github.com/truong02bp/dulich-spring-boot?fbclid=IwAR3eXkvqDqhuRXvst3uFb49_W2O4k8z3xw1X2UU0_f7NDy_e7Fx8sV57wuw)

# Tài liệu tham khảo

X`**2012.** *Getting Starting with JSP with example.* [Online] 10 2012. [Cited: 10 21, 2020.] https://www3.ntu.edu.sg/home/ehchua/programming/java/JSPByExample.html.

**edureka. 2015.** edureka. [Online] 1 9, 2015. [Cited: 10 21, 2020.] https://www.youtube.com/watch?v=\_8\_kQz1m-0U&t=960s&ab\_channel=edureka%21.

**2018.** Geeksforgeeks. [Online] 7 2, 2018. [Cited: 10 20, 2020.] https://www.geeksforgeeks.org/introduction-to-jsp/.

**2018.** Geeksforgeeks. [Online] 7 5, 2018. [Cited: 10 20, 2020.] https://www.geeksforgeeks.org/difference-between-servlet-and-jsp/.

**2018.** Guru99. [Online] 2018. [Cited: 10 21, 2020.] https://www.guru99.com/jsp-tutorial.html.

**Iuv2code. 2017.** Youtube. [Online] 2 9, 2017. [Cited: 10 20, 2020.] https://www.youtube.com/watch?v=40KM8IdneLg&list=PLEAQNNR8IlB588DQvb2wbKFQh2DviPApl&ab\_channel=luv2code.

**Piercy, Craig. 2014.** Craig Piercy. [Online] 1 26, 2014. [Cited: 10 20, 2020.] https://www.youtube.com/watch?v=fZMUMNqf6mg&t=325s&ab\_channel=CraigPiercy.