**CÔNG NGHỆ JSP**

**1. Giới thiệu**

- File jsp:

file văn bản trộn lẫn lệnh của HTML và lệnh của Java.

tên file có đuôi là jsp

lưu trên máy chủ (máy cài web server)

mục đích để tạo trang web động.

- Mô hình:

Browser

Web Server

Jsp engine

File Jsp

CSDL

Nhận trang web động

Y/c file Jsp

- Cơ chế hoạt động:

Khi browser yêu cầu file jsp, web server gọi Jsp Engine.

Jsp Engine sẽ nạp file jsp vào bộ nhớ, biên dịch thành file servlet

Cho thực thi servlet, tạo ra một trang web động và gởi về cho browser.

Browser thông dịch trang web động, hiện kết quả trên cửa sổ browser.

Lưu ý:

Jsp engine chỉ cần biên dịch file jsp sang file servlet.class chỉ một lần.

Các yêu cầu sau, Jsp engine chỉ cần thực thi servlet, nên tốc độ nhanh hơn nhiều so với lần đầu.

JSP Engine

Web Server

file mới hoặc đã cập nhật?

thực thi servlet có sẵn

biên dịch jsp thành servlet

thực thi servlet mới

Yes

No

**2. Các thẻ lệnh:**

+ Chú thích: <%-- Các lệnh html --%>

+ Lệnh java: <% Các lệnh java %>

+ Hiện giá trị biểu thức: <% = biểu thức %>

+ Định nghĩa hàm hoặc khai báo biến application: <%! Hàm hoặc biến application %>

+ Nhúng file:

Nhúng nội dung: <%@ include file=”filename” %>

Nhúng kết quả: <jsp:include page=”filename” />

hoặc

<jsp:include page=”filename” >

<jsp:param name=”varname” value=”value”/>

</jsp:include>

+ Nhúng lớp: <%@ page import="classname"%>

+ Chuyển trang:

- Tự chuyển tham số:

<jsp:forward page=”filename”/> hoặc

- Không tự chuyển tham số:

response.sendRedirect(“filename?varname=value”);

**3. Các đối tượng webserver tạo sẵn**

+ Đối tượng out: dùng để gởi dữ liệu cho browser

out.print(“lệnh html”);//gởi lệnh html cho browser

+ Đối tượng request: dùng để chứa dữ liệu do browser gởi

String giatri=request.getParameter(“tên tham số”);//lấy giá trị tham số cho browser gởi

request.setAttribute(“tên thuộc tính”,giatri);//thiết lập thuộc tính=giatri

String giatri=(String)request.getAttribute(“tên thuộc tính”);//lấy giá trị thuộc tính

+ Đối tượng response: dùng để gởi dữ liệu cho browser

+ Đối tượng session: dùng để lưu các biến session

+ Đối tượng application: dùng để lưu các biến application

**4. Đối tượng cookie, session, application**

Khi browser gởi yêu cầu đến web server, browser sẽ thiết lập kết nối với webserver.

Khi nhận được trang web thì kết nối sẽ kết thúc.

Nếu browser muốn yêu cầu trang web khác thì phải thiết lập kết nối mới.

Để lưu lại thông tin của các lần kết nối trước đó, web server hỗ trợ các đối tượng session, application, cookies.

**4.1 Đối tượng session**

- Phiên làm việc tính từ lúc gởi yêu cầu đầu tiên cho đến khi đóng browser.

- Webserver sẽ cấp cho mỗi phiên một đối tượng session dùng để lưu dữ liệu của phiên (gọi là các biến session).

- Dữ liệu lưu trong đối tượng session sẽ được dùng chung cho các lần yêu cầu trong cùng phiên.

- Đối tượng session sẽ tồn tại cho đến khi kết thúc phiên làm việc (browser được đóng lại) hoặc khi hết thời gian sống do web server qui định, khi đó tất cả các biến session lưu trong đối tượng session sẽ mất.

- Các phương thức:

+ Đặt biến session tên là name, giá trị là value vào trong đối tượng session

public void setAttribute(String name, Object value)

ví dụ: session.setAttribute(“masv”,”001”);

+ Lấy giá trị biến session tên là name trong đối tượng session

public Object getAttribute(String name)

ví dụ: String masv=session.getAttribute(“masv”);

+ Hủy biến session tên là name trong đối tượng session

public void removeAttribute(String name)

ví dụ: session.removeAttribute(“masv”);

+ Hủy đối tượng session, khi đó tất cả các biến session trong đối tượng session cũng bị hủy

public void invalidate()

ví dụ: session.invalidate();

**4.2 Đối tượng application**

- Khi web server được cho thực thi, webserver sẽ tạo một đối tượng application dùng chung cho tất cả các phiên làm việc và cho mọi NSD.

- Đối tượng application sẽ tồn tại cho đến khi web server được khởi động lại, khi đó tất cả các biến application lưu trong đối tượng application sẽ mất.

- Các phương thức: giống như các phương thức trên đối tượng session

ví dụ:

application.setAttrbute(“count”,new Interger(1));

Integer count= (Integer)application.getAttribute(“count”);

application.removeAttrbute(“count”);

application.invalidate();

**4.3. Đối tượng cookie**

- Biến cookie được web server lưu ở máy client, biến session/application được lưu ở máy server.

- Khi browser gởi yêu cầu đến web server thì cũng gởi tất cả các biến cookie cho web server (chỉ gởi các biến cookie do web server đó tạo ra)

- Biến cookie sẽ bị hủy khi hết thời gian sống do người lập trình qui định hoặc do browser qui định

- Các phương thức:

+ Tạo biến cookies:

Cookie c=new Cookie(“name”,value);

+ Đặt thời gian sống cho biến cookie

c.setMaxAge(time);// time tính bằng giây, time=0 là hủy biến cookie

+ Ghi biến cookie vào máy client

response.add(c);

+ Lấy tất cả các biến cookie do client gởi

Cookies [] cs=request.getCookies();

+ Lấy tên biến và giá trị biến cookie thứ i

cs[i].getName;

cs[i].getValue;

**5. Chuyển tham số:**

url:

<url>?<name>=<value> [&[…]]

neu su dung form thì method phải là post, nhưng value không có dấu tiếng việt

form:

<form action=<html/jsp/servlet> method=post/get>

<input type=text name=”tents” value=”giatri”>

<input type=input value=”OK”>

</form>

hidden:

<form action=<html/jsp/servlet> method=post/get>

<input type=hidden name=”tents1” value=”giatri”>

<input type=text name=”tents2” value=”giatri”>

<input type=input value=”OK”>

</form>

session:

session.setAttribute(“tents”,”giatri”);

String giatri=session.getAttribute(“tents”);

**6 Java Bean**

- Giới thiệu:

+ Là đối tượng có thể truy xuất bằng cách dùng thẻ lệnh của Jsp

+ Trong bean cần có các phương thức gán giá trị cho thuộc tính (phương thức set) và các phương thức lấy giá trị thuộc tính (phương thức get)

+ Bean dùng để che dấu mã java trong Jsp và dễ dàng sử dụng bean như biến session, application.

- Các thẻ lệnh Jsp truy xuất bean

+ Khai báo sử dụng bean

<jsp:useBean id=”beanname” scope=”page/session/application” class=”classname”>

</jsp:useBean>

+ Gán giá trị cho thuộc tính của bean

<jsp:setProperty name=”beanname” property=”propertyname” value=”value”/>

+ Lấy giá trị thuộc tính của bean

<jsp:getProperty name=”beanname” property=”propertyname”/>