JSP & Servlet

Nguyễn Thanh Quân – ntquan@fit.hcmuns.edu.vn

Ngôn ngữ HTML

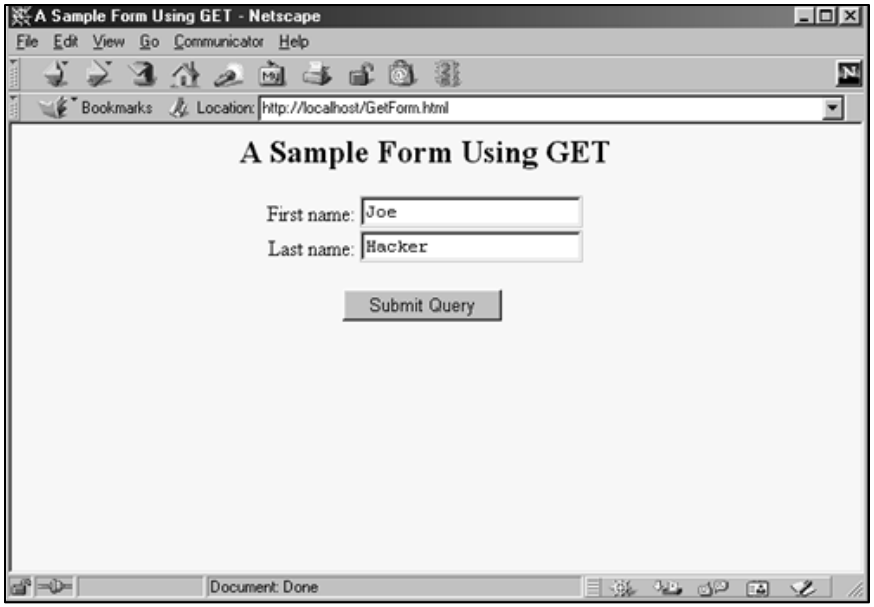
- Ngôn ngữ cơ bản nhất để hiển thị một trang web là ngôn ngữ HTML
- HTML gồm tập hợp các thẻ (tag), ví dụ để tô đậm một đoạn văn bản ta dùng thẻ : bold text
- Ngôn ngữ HTML 4.0 qui định rất nhiều thẻ, thực hiện nhiều chức năng: format text, table, input form, chèn hình ảnh, flash movies… (Tham khảo sách Core Web Programming phần HTML).
- Vai trò của các ngôn ngữ lập trình web cao cấp hầu như đóng vai trò thực thi mã lệnh của người dùng (dưới dạng Java, .NET, php script…) từ đó sau đó phát sinh HTML code và gửi về cho trình duyệt.

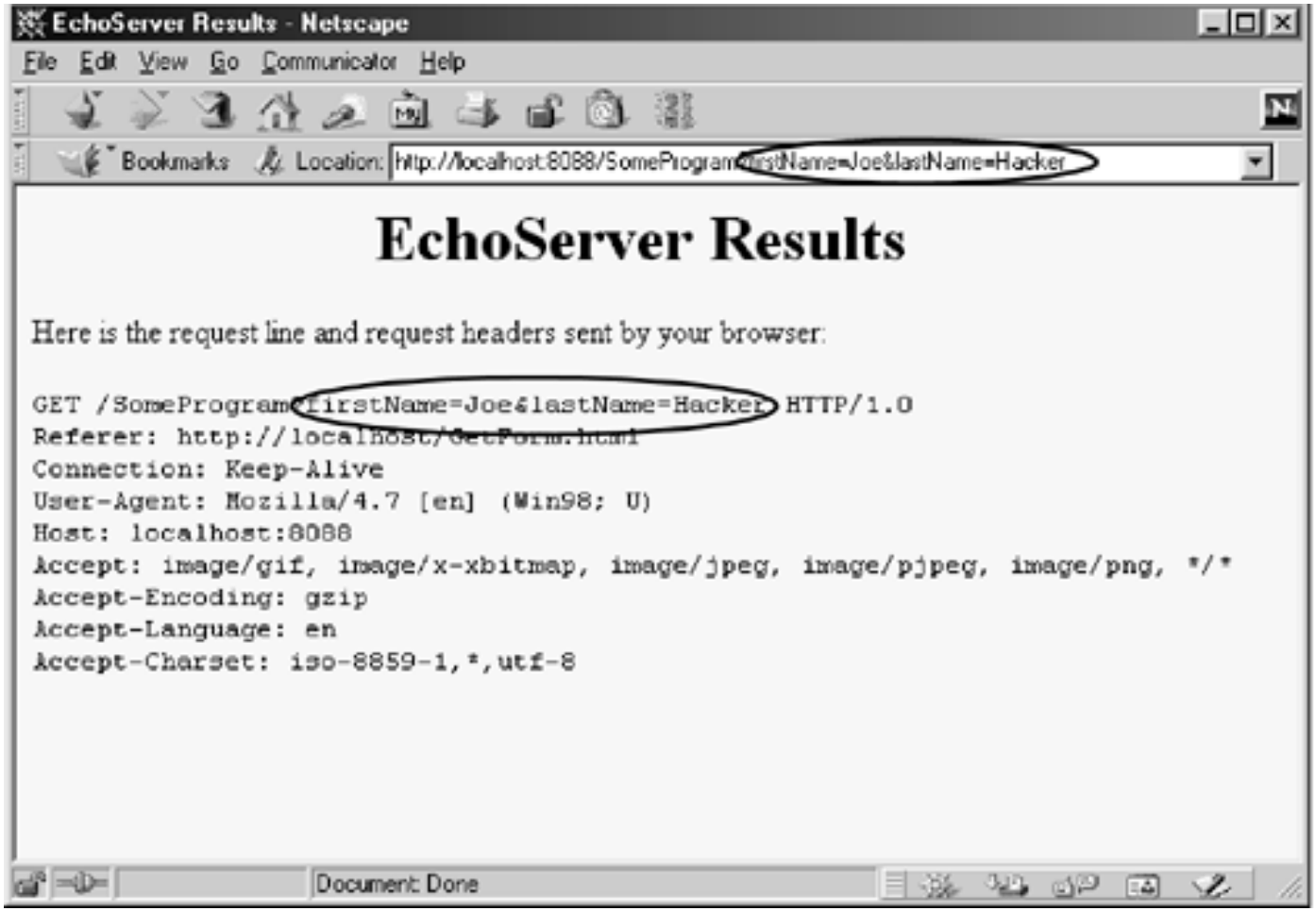
HTML Form

- Get: Các parameter được gửi kèm với URL
- Post: Các parameter gửi bên trong gói tin request

GET

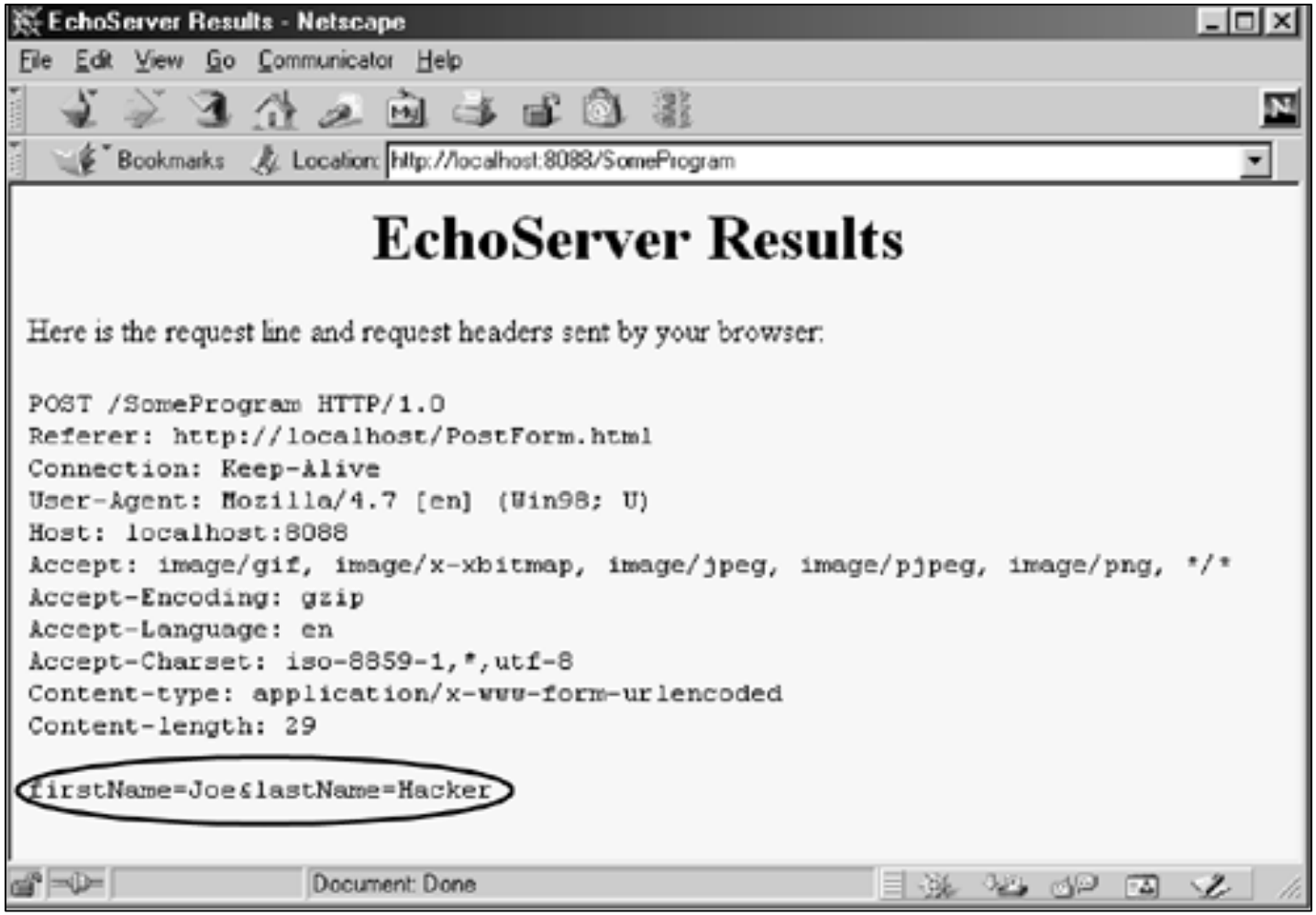
```
<FORM ACTION="http://localhost:8088/SomeProgram">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" id="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" id="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT"> <!-- Press this button to submit form -->
  </CENTER>
</FORM>
```




EchoServer Results - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://localhost:8088/SomeProgram?firstName=Joe&lastName=Hacker
EchoServer Results
Here is the request line and request headers sent by your browser:
GET /SomeProgram?firstName=Joe&lastName=Hacker HTTP/1.0
Referer: http://localhost/getForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Document: Done

POST

```
<FORM ACTION="http://localhost:8088/SomeProgram"
      METHOD="POST">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" id="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" id="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT">
  </CENTER>
</FORM>
```


EchoServer Results - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://localhost:8088/SomeProgram
EchoServer Results
Here is the request line and request headers sent by your browser:
POST /SomeProgram HTTP/1.0
Referer: http://localhost/PostForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Content-type: application/x-www-form-urlencoded
Content-length: 29
firstName=Joe&lastName=Hacker
Document: Done

Các thành phần của FORM

- Text
- Button
- Radio Button & Check box
- Combo Box & List
- Password Field
- ……..

  Tham khảo: Core Web Programming Phần “HTML Form”

Client-side & Server-side programming

- Client-side: các đoạn code sẽ chạy ở phía trình duyệt của người dùng. Điển hình nhất là các mã lệnh javascript

Tham khảo thêm sách : CSS & Java Script

- Server-side: Các ngôn ngữ lập trình web cao cấp (JSP/Servlet, ASP.NET, PHP…) sẽ được thực thi ở phía Web Server đảm nhận các vai trò phức tạp như kết nối CSDL, tính toán … sau đó phát sinh các đoạn mã HTML và gửi về cho trình duyệt.

Servlet

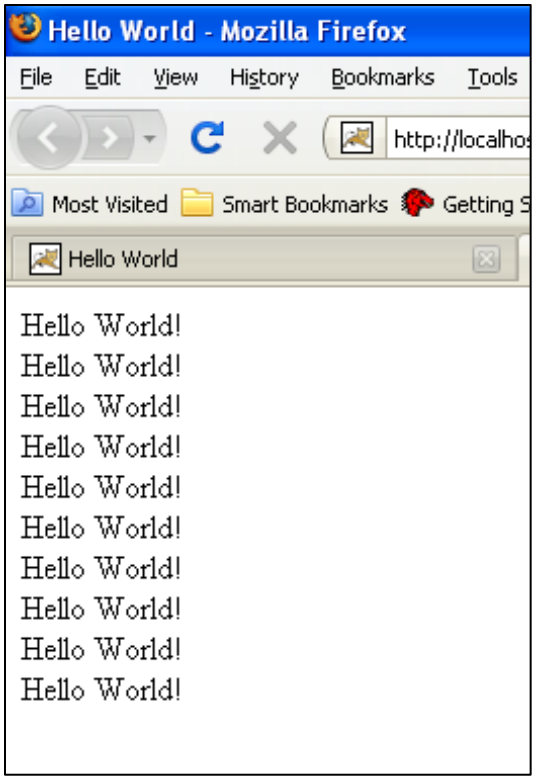
- Là một class Java hoàn chỉnh, gồm các phần khai báo biến, các hàm …
- Servlet tương ứng với một trang web để người dùng truy xuất đến.
- Trong một servlet không có hàm main, phục vụ người dùng thông qua 2 hàm:
  - doPost(): nếu người dùng truy xuất bằng giao thức post
  - doGet(): nếu người dùng truy xuất bằng giao thức get

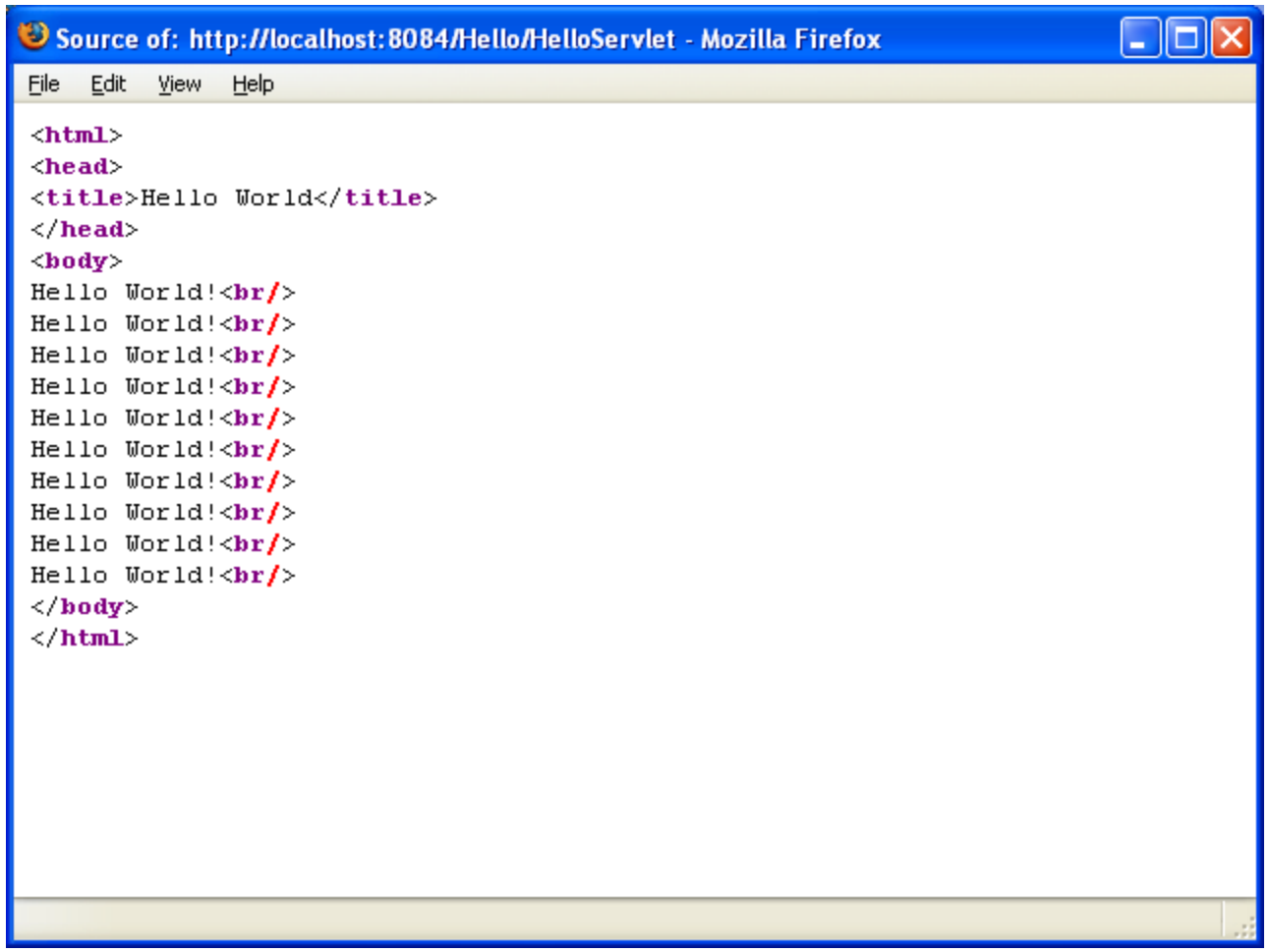
Ví dụ 1 servlet:

- Servlet sau sẽ in 10 dòng “Hello World” lên màn hình của trình duyệt

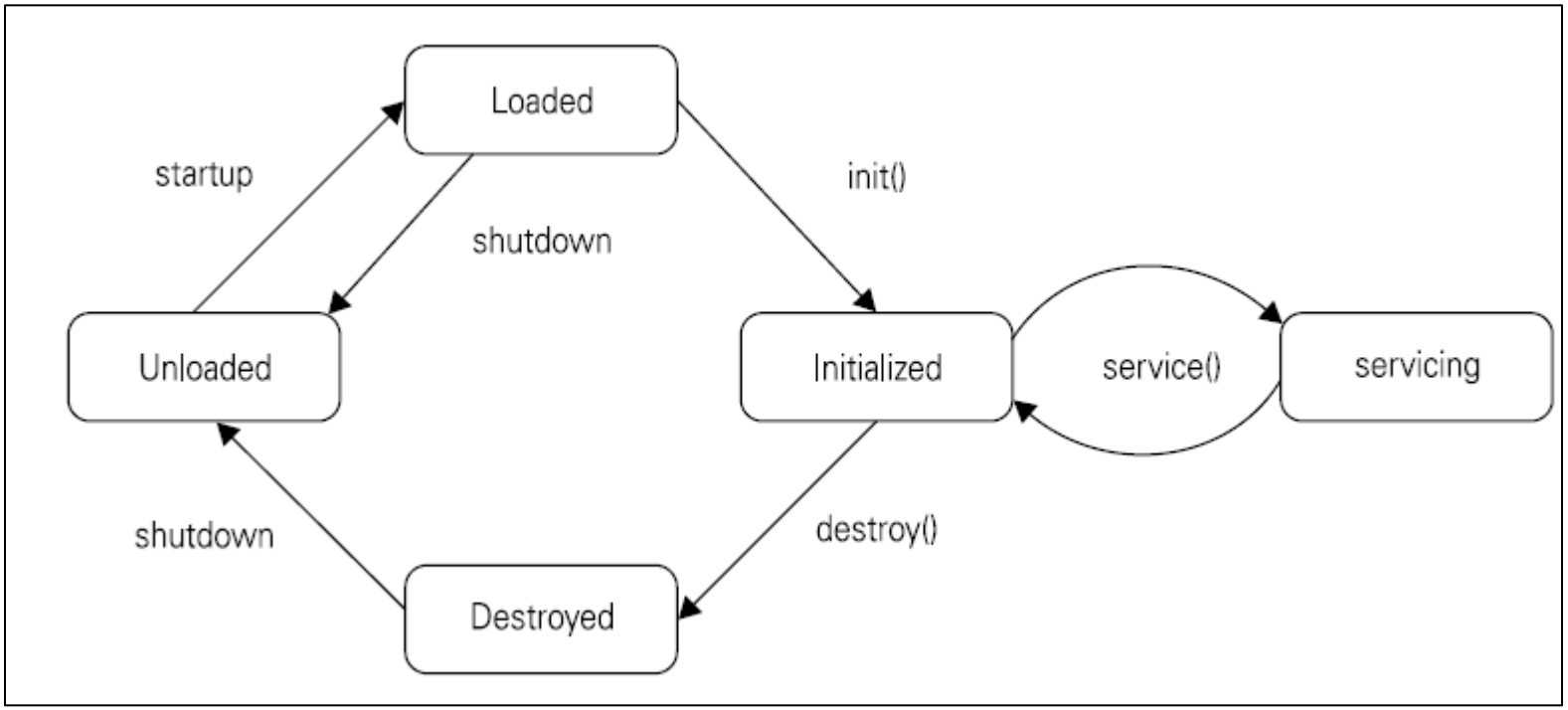
```
public class HelloServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World</title>");
        out.println("</head>");
        out.println("<body>");
        for (int i=0;i<10;i++){
            out.println("Hello World!<br/>");
        }
        out.println("</body>");
        out.println("</html>");
    }
}
```

Nội dung trang / View page source





Chu trình sống của một servlet



Các hàm quan trọng

- init(): Servlet được nạp khi web server khởi động, hàm init chỉ được gọi một lần ứng với mỗi servlet.
- destroy(): trước khi trang servlet bị hủy, hàm destroy của servlet tương ứng sẽ được gọi
- service(): mỗi khi trang web được truy xuất (thông qua các web browser của người dùng), hàm service được gọi. Tùy theo giao thức truy xuất là post/get hàm service sẽ quyết định gọi hàm doPost() hoặc doGet()
- getServletConfig(): lấy các biến được lưu trong file web.xml

Các đối tượng dùng để truy xuất dữ liệu, điều khiển trang web

- Request: Lấy dữ liệu gửi lên từ phía trình duyệt
- Reponse: Trả dữ liệu về trình duyệt

HttpServletRequest

- HttpServletRequest phân tích thông tin incoming của HTML form data và lưu dưới dạng các parameters.
- Hỗ trợ sử dụng cookies và session, cung cấp khả năng truy xuất các thông tin headers.
- Được truyền vào dưới dạng tham số của các hàm doGet(…), doPost(…) hoặc được tạo sẵn trong trang jsp với tên request.

Các phương thức phổ biến

- public abstract Cookie[] getCookies();
- public abstract String getHeader(String name);
- public abstract HttpSession getSession(boolean create);
- public String getParameter(String name);
- public Enumeration getParameterNames();
- public String[] getParameterValues(String name);

HttpServletResponse

- Cũng được truyền vào trang servlet thông qua các hàm doGet(…), doPost(…); trong trang jsp được cài đặt sẵn với tên response.
- Được dùng để tạo các thông tin phản hồi từ server về phía browser
- Chuyển hướng trình duyệt, thêm cookies, thêm http headers.

Các phương thức phổ biến

- public abstract void addCookie(Cookie cookie);
- public abstract void addHeader(String name, String value);
- PrintWriter getWriter();
- public void sendRedirect(String location) throws IOException;

JSP Page

- Có thể xem JSP là một dạng trình bày ngắn gọn hơn của servlet
- Trong một file JSP, các mã HTML được viết đan xen với mã JAVA
- Mã Java được khai báo trong tag

<%

Java code

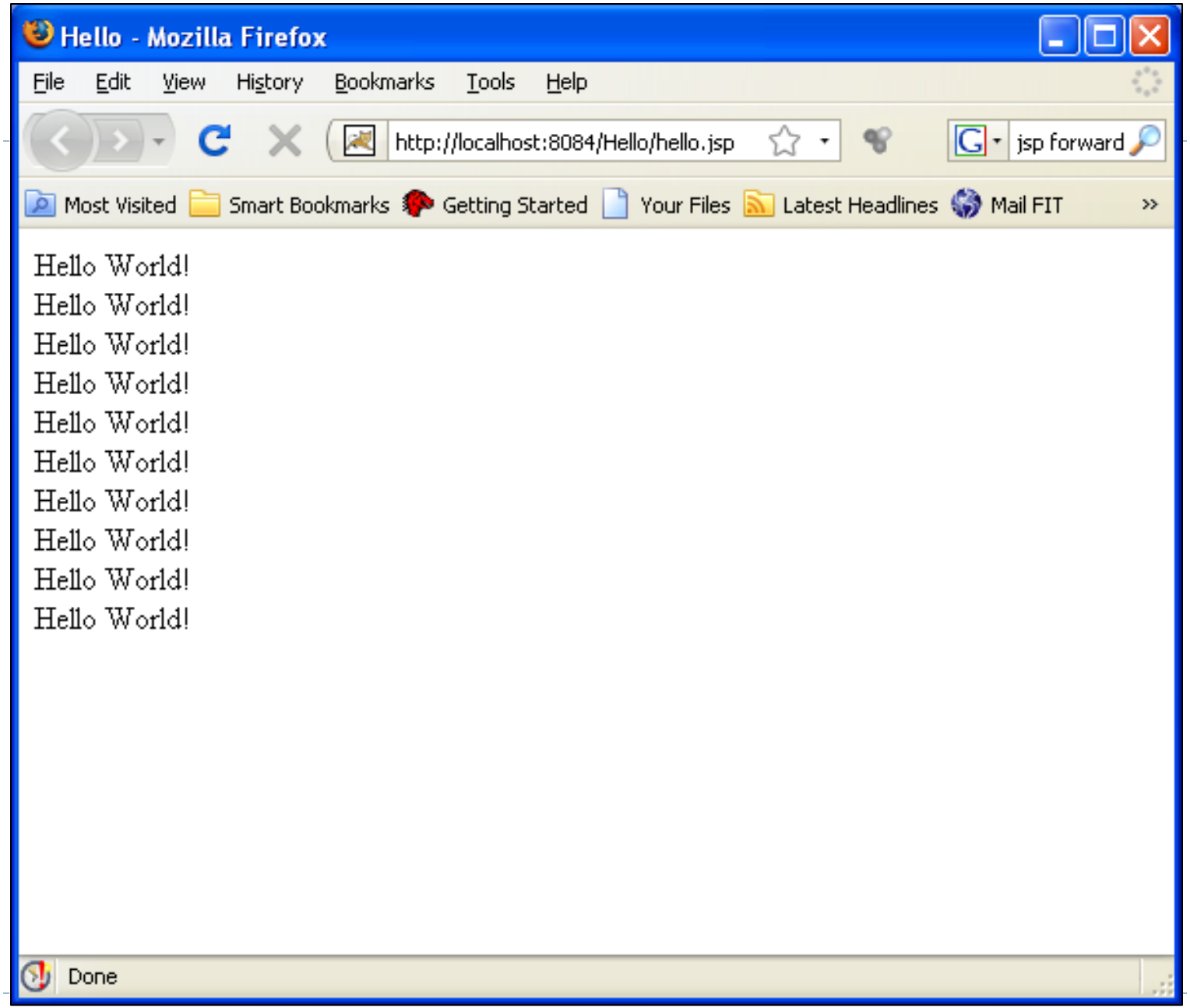
…….

%>

Ví dụ trang jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Hello</title>
    </head>
    <body>
        <%
          for (int i=0;i<10;i++){
              out.println("Hello World!<br/>");
          }
        %>
    </body>
</html>
```


Hello - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/Hello/hello.jsp
jsp forward
Most Visited Smart Bookmarks Getting Started Your Files Latest Headlines Mail FIT
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Done

JSP vs Servlet

- JSP thực sự cũng là một trang servlet. Khi trang jsp được triệu gọi, trước tiên sẽ được biên dịch thành servlet
- Quá trình biên dịch này do các web server đảm nhận, không phải lập trình viên

Chu trình sống

Phase Name	Description
Page translation	The page is parsed and a Java file containing the corresponding servlet is created.
Page compilation	The Java file is compiled.
Load class	The compiled class is loaded.
Create instance	An instance of the servlet is created.
Call jspInit()	This method is called before any other method to allow initialization.
Call _jspService()	This method is called for each request.
Call jspDestroy()	This method is called when the servlet container decides to take the servlet out of service.

Khai báo biến

- Có 2 dạng khai báo: biến của lớp, biến của hàm service
- Biến của lớp:

<%! int n; %>

Biến n được khai báo ở mức class (biến của lớp), được khởi tạo giá trị = 0

- Biến cục bộ của hàm _jspservice()

<% int n; %>

Biến n được xem như khai báo cục bộ trong hàm _jspservice(), không được khởi gán giá trị mặc định

Chèn java code:

Sử dụng một trong 2 cách:

- <% code %>
- <jsp:scriptlet>Code</jsp:scriptlet>
- Ví dụ:

```
<% String query = request.getQueryString(); %>
```

Lấy giá trị biến, biểu thức

- <%= expression %>
- <jsp:expression>Expression </jsp:expression>
- Ví dụ

Hostname: <%= request.getRemoteHost() %>

Count = <%= count %>

- Chú ý: Không có dấu ; ở cuối dòng

Các biến có sẵn

- request - HttpServletRequest (1st doGet)
- response - HttpServletResponse (2nd doGet)
- session - HttpSession
- out - JspWriter
- application - ServletContext

Directive:

- Import:

  <%@ page import="package.class" %>

  <%@ page import="package.class1,...,package.classN" %>

- Format

  <%@ page contentType="MIME-Type" %>

  <%@ page contentType="MIME-Type; charset=Character-Set"%>

```
<%@ page contentType="text/html;charset=ISO-8859-1" %>
```

<jsp:include>

- <%@ include file="Relative URL" %>
- <jsp:include page="Relative URL" flush="true" />
- Mục đích
  - Trang được include sẽ được thực thi trước, lấy kết quả chèn vào trang gọi chỉ thị include
- Với servlet có thể dùng hàm include của lớp RequestDispatcher

<%@include %>

- Khác với jsp:include, chỉ thị này include trực tiếp source code của một file jsp khác vào file này. Ví dụ:
- <%@ include file="connection.jsp"%>

<%jsp forward %>

- Chuyển hướng sang 1 trang web khác:
  - <jsp:forward page="/servlet/login" />
- Truyền tham số theo kèm:

```
<jsp:forward page="/servlet/login">
<jsp:param name="username" value="jsmith" />
</jsp:forward>
```

Chuyển trang web bằng cách khác

- RequestDispatcher rd = request.
  getRequestDispatcher("pathToResource");
  rd.forward(request, response);
- Hoặc dùng response.sendRedirect()

Cấu trúc ứng dụng Web

Trong thư mục project do Netbeans tạo ra sẽ có:

Build (thư mục gốc của web application)

- -> *.html, *.jsp, *.shtml (các file không phải là class java)
- -> Web-inf (thư mục)
  - -> web.xml (file định dạng của ứng dụng : các thong số cấu hình)
  - -> classes (thư mục chứa các java class và servlet class)
  - -> lib (thư mục chứa các tập tin thư viện mà ứng dụng import)

Các bạn cũng có thể vào thư mục "/dist" để xem cấu trúc ứng dụng ở dạng file nén .war, nội dung file này sẽ có đầy đủ các thư mục nêu trên.

Có thể chép file war lên webserver, server sẽ tự giải nén và deploy ứng dụng web

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
<servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>web.HelloServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/HelloServlet</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>        30        </session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Tài liệu tham khảo

- Core Web Programming
- SCWCD