



# LẬP TRÌNH WEB CHẠY Ở SERVER

## Bài Giảng 7



# Nội dung

- Lập trình Web với CGI
- Lập trình Web với Servlet
- Lập trình Web với JSP
- Lập trình Web với ASP
- Lập trình Web với PHP (tự học)



# Lập trình web với CGI

- Trang HTML chỉ chứa các nội dung cố định hoặc có thể thao tác thay đổi qua DHTML ở client.
- Nhu cầu sử dụng các trang web như một ứng dụng (giao tiếp với user, tìm kiếm, nội dung thay đổi, tương tác các ứng dụng, với database...)=> ứng dụng web.
- CGI (Common Gateway Interface) là một chuẩn (standard) để viết ứng dụng web.



# Lập trình web với CGI

- **Các đặc điểm**

- Là ứng dụng chạy trên nền web server.
- Nhận thông tin từ web browser và xuất thông tin để web browser hiển thị thông qua web server.
- Ứng dụng có thể được viết với bất cứ ngôn ngữ lập trình có standard input và standard output. (C/C++, Perl, Shell script, Dos shell...)

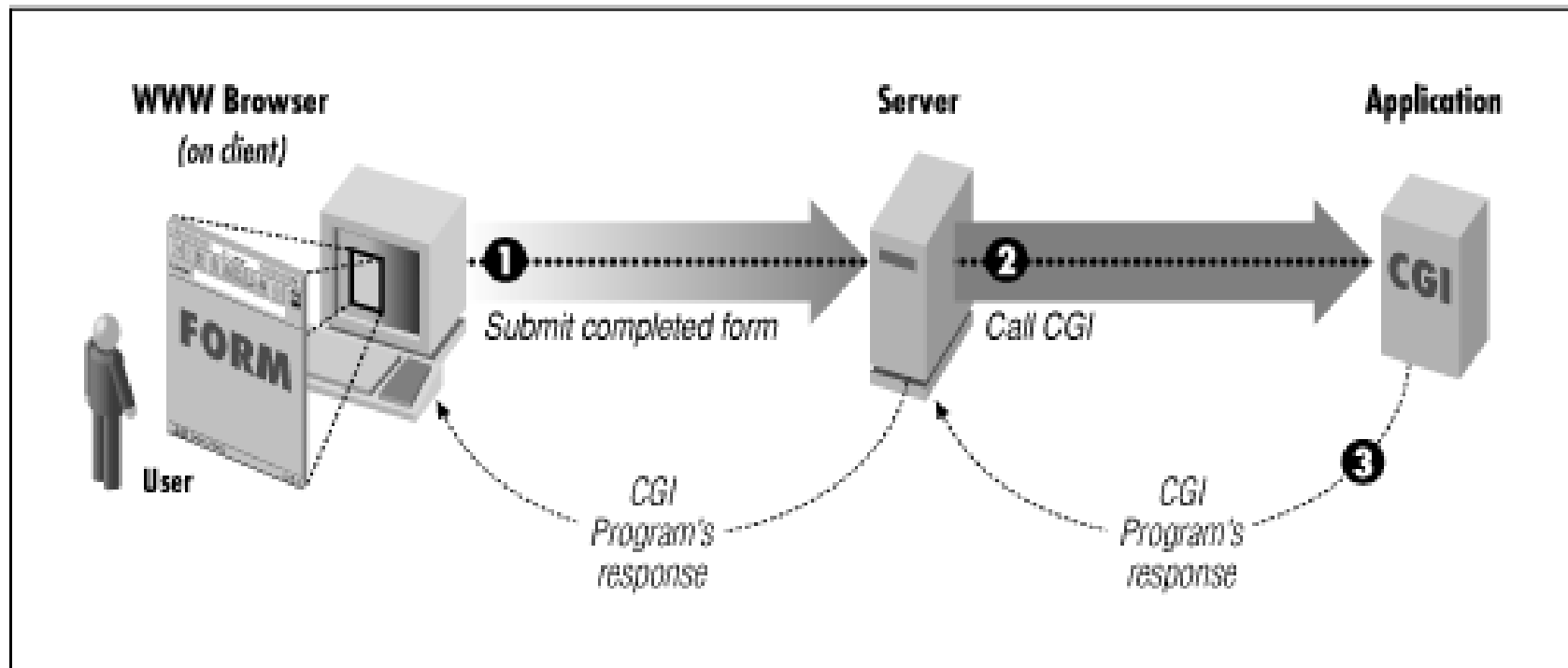


# Lập trình web với CGI

- **Cách thức hoạt động của ứng dụng web CGI**
  - Web server nhận request ứng dụng CGI từ browser.
  - Web server gọi ứng dụng CGI, truyền các thông số bằng các biến môi trường (thông qua standard input).
  - Ứng dụng web CGI xử lý, giao tiếp với các ứng dụng khác (database, mail...), xuất kết quả dạng HTML qua standard output đến web server.
  - Web server nhận kết quả và trả về cho browser.

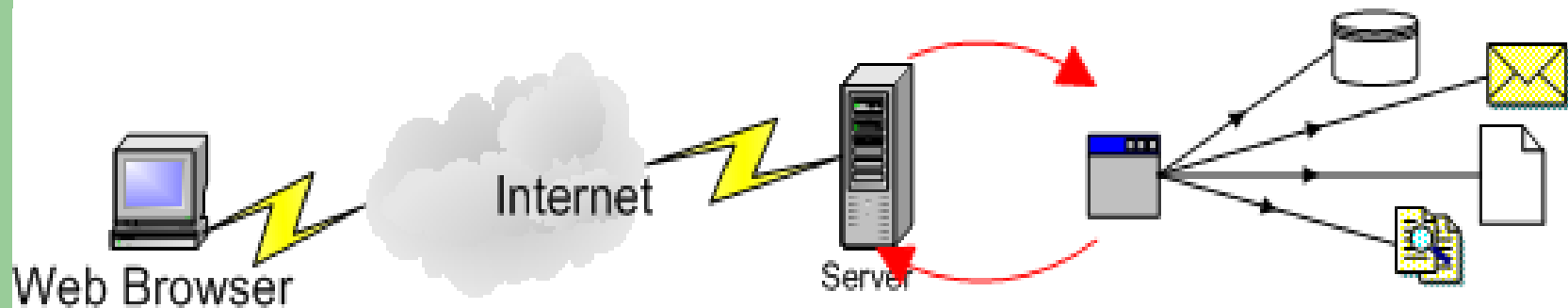
# Lập trình web với CGI

- Hình vẽ minh họa hoạt động CGI



# Lập trình web với CGI

- Hình vẽ minh họa hoạt động CGI (Perl)



Client sends a standard HTTP request encoded with special variables and their values. These variables are commonly called "CGI variables"

The web server receives the HTTP request and using special rules, determines that the request is destined for a CGI program and not for an HTML file.

The Perl interpreter accepts the input from the server, parses the variables, applies some programming logic to the data and returns a document back to the webserver.

Perl is able to call libraries and other network functions to complete the request.



# Lập trình web với CGI

- Ví dụ (Perl)

```
#!/perl/bin/perl
#Remember : this path will vary depending on
#where Perl is located
print "Content-type:text/html\n\n";
print "<html><head><title>HELLO!</title></head>";
print "<body>\n";
print "<h2>Hello!</h2>\n";
foreach $key (sort(keys %ENV)) {
    print "VARIABLE $key = $ENV{$key}<br>\n";
}
print "</body></html>\n";
```



# Lập trình web với CGI

- Các biến môi trường

Variable	Data
DOCUMENT_ROOT	The root directory of the web server
HTTP_COOKIE	The visitor's cookie, if one is set
HTTP_HOST	The name of your server
HTTP_REFERER	The page that called the script
HTTP_USER_AGENT	The browser the visitor is using
HTTPS	Set "on" if it is called through a secure server
PATH	The system path your server is running on
QUERY_STRING	See the GET method of receiving data below
REMOTE_ADDR	The visitors IP address
REMOTE_HOST	The hostname of the visitor
REMOTE_PORT	The port on your web server the visitor is connected to
REMOTE_USER	The visitors username
REQUEST_METHOD	GET or POST - see below
REQUEST_URI	The interpreted pathname of the requested document or CGI
SCRIPT_FILENAME	The full path name of the CGI
SCRIPT_NAME	The interpreted pathname of the CGI

# Lập trình web với CGI

- **Các biến môi trường**

SERVER_ADMIN	The email address of the webmaster
SERVER_NAME	Your domain name
SERVER_PORT	The port the server is set to
SERVER_SOFTWARE	Your server type eg. Apache 1.3.11

- **Ví dụ**

Để in ra địa chỉ IP của máy khách:

```
print "IP Address = $ENV{'REMOTE_ADDR'}\n";
```

- **Một số các ví dụ cụ thể**

- Dùng phương thức GET: biến môi trường lưu ý là QUERY\_STRING
- Dùng phương thức POST.
- Lưu các biến dùng COOKIE hoặc Session.
- Lập trình CGI kết nối với CSDL.



# Lập trình web với CGI

- **Tài liệu tham khảo**

- Perl Programming

- [http://inconnu.isu.edu/~ink/perl\\_cgi/](http://inconnu.isu.edu/~ink/perl_cgi/)

- <http://www.bewley.net/perl/>

- [http://jan.netcomp.monash.edu.au/ecommerce/perl\\_session.html](http://jan.netcomp.monash.edu.au/ecommerce/perl_session.html)



# Giới thiệu Servlet

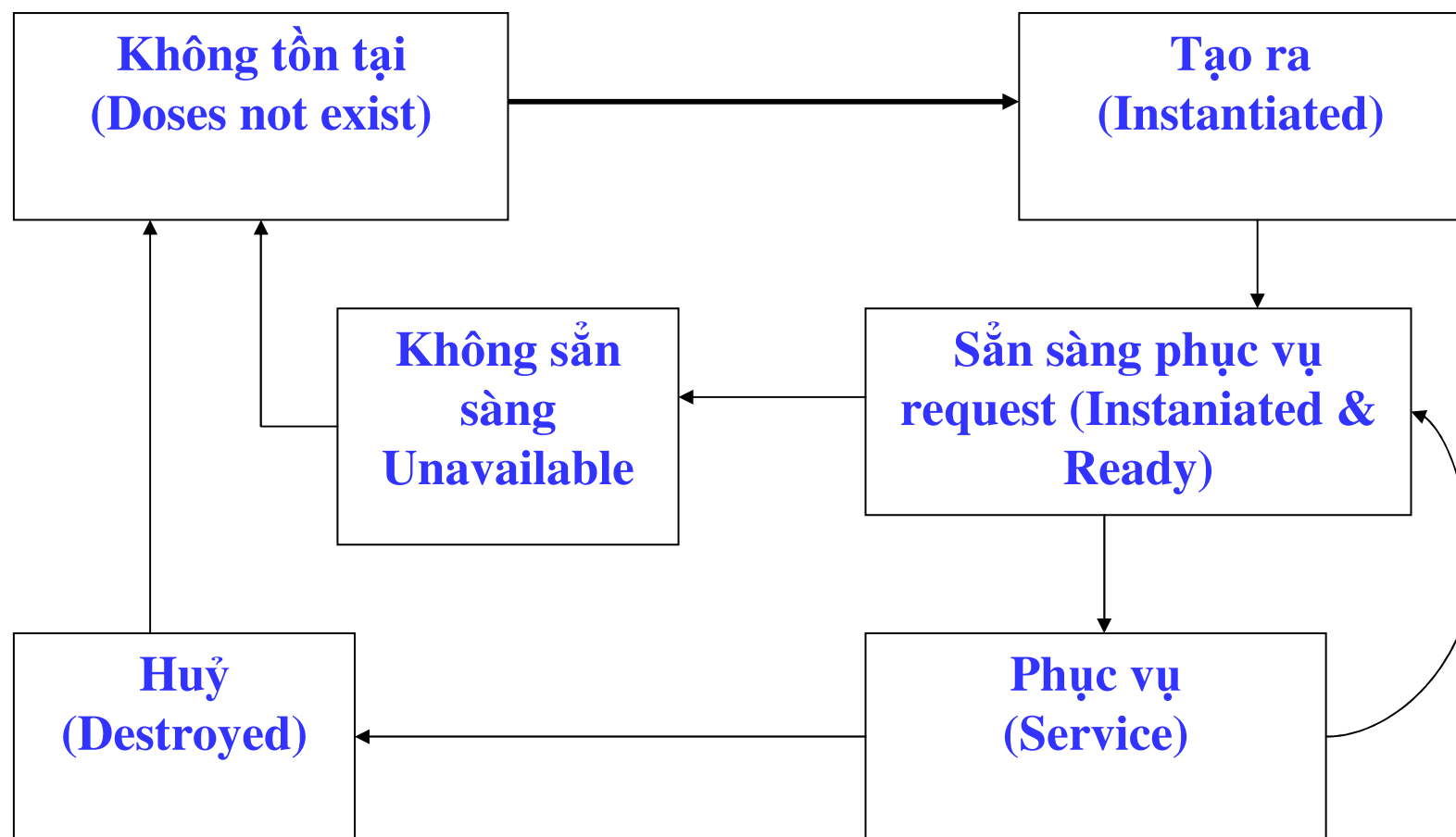
- Java servlet là những chương trình độc lập platform và nhỏ chạy ở phía server được lập trình để mở rộng chức năng của web server (Web server phải hỗ trợ Java).
- Cơ chế hoạt động theo mô hình CGI mở rộng.
- Chương trình servlet:
  - Thường extends class `HttpServlet`. Không có method *main*.
  - Phải được dịch ra ở dạng byte-code (.class) và khai báo với web server.



# Giới thiệu Servlet

- Cần có package `javax.servlet` để biên dịch (<http://java.sun.com/products/servlet/>)
- Các server hiện hỗ trợ Servlet:
  - Apache Tomcat (<http://jakarta.apache.org>)
  - Sun's Java Web Server, free, hiện không cho download (<http://www.sun.com/software/jwebserver/>)
  - New Atlanta's ServletExec, tích hợp ServletEngine vào các web server(<http://newatlanta.com>)
  - <http://www.macromedia.com/software/jrun/trial/>
  - ...
- Tham khảo các tài liệu về Servlet:  
<http://java.sun.com/products/servlet/docs.html>

# Vòng đời của Servlet





# Lập trình với Servlet API (t.t)

- **Java Servlet API gồm:**
  - **javax.servlet.\***
    - Chứa các class và interface độc lập với HTTP
  - **javax.servlet.http.\***
    - Chứa các class cụ thể cho HTTP
- Tất cả servlet đều phải hiện thực giao diện **javax.servlet.Servlet**.
- Tuy nhiên do chúng ta dùng giao thức HTTP nên **javax.servlet.http.HttpServlet** là lớp hiện thực giao diện Servlet, nên
- Vì vậy ta thường phải kế thừa từ lớp này và override các method **doGet()/doPost()**.



# Lập trình với Servlet API (t.t)

Mục đích	Class/Interface
Hiện thực servlet	<code>javax.servlet.Servlet</code> <code>javax.servlet.SingleThreadModel</code> <code>javax.servlet.GenericServlet</code> <code>javax.servlet.http.HttpServlet</code>
Cấu hình sevlet	<code>javax.servlet.ServletConfig</code>
Servlet Exception	<code>javax.servlet.ServletException</code>
Requests	<code>javax.servlet.ServletRequest</code> <code>javax.servlet.http.HttpServletRequest</code> <code>javax.servlet.ServletInputStream</code>





# Lập trình với Servlet API (t.t)

Mục đích	Class/Interface
Responses	<code>javax.servlet.ServletResponse</code> <code>javax.servlet.http.HttpServletResponse</code> <code>javax.servlet.ServletOutput</code>
Session tracking	<code>javax.servlet.http.HttpSession</code> <code>java.servlet.http.HttpSessionBindingListener</code>
Servlet context	<code>javax.servlet.ServletContext</code>
Servlet collaboration	<code>javax.servlet.RequestDispatcher</code>
Khác	<code>javax.servlet.http.Cookie</code> <code>java.servlet.http.HttpUtils</code>

# Cơ chế hoạt động của servlet

- Web server nhận yêu cầu triệu gọi servlet từ client.
- Nếu servlet chạy lần đầu, web server load file servlet tương ứng, khởi tạo các thông số bằng qua method **init()**
- Nếu servlet đã được khởi tạo, tạo một thread để xử lý yêu cầu.
- Gọi methods **doXXX()** để xử lý các request tương ứng theo giao thức HTTP:
  - **doGet** cho HTTP GET, **doPost** cho HTTP POST



# Cấu trúc đơn giản của một Servlet

```
import java.io.*;
import java.servlet.*;
import java.servlet.http.*;
public class Sample extends HttpServlet{
    public doGet(HttpServletRequest request,
                  HttpServletResponse response)
        throws ServletException, IOException{
        //dùng đối tượng "request" để đọc dữ liệu từ client
        //dùng đối tượng "response" để xuất dữ liệu cho client
        PrintWriter out = response.getWriter();
        //dùng đối tượng out để ghi (method print) dữ liệu cho client
    }
}
```



# Một Servlet đơn giản

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```



# Lập trình Web với Servlet

- **API cho HttpServletRequest**
  1. Object `getAttribute(String name)`
  2. void `setAttribute(String name, Object att )`
  3. Enumeration `getAttributeNames()`
  4. int `getContentLength()`
  5. String `getParameter(String name)`
  6. Enumeration `getParameterNames()`
  7. Enumeration `getParameterValues()`
  8. String `getRemoteHost()`
  9. RequestDispatcher `getRequestDispatcher( String path )`
  10. boolean `isSecure()`
  11. ServletInputStream `getServletInputStream()`
  12. int `getContentLength()`
  13. ...



# Lập trình Web với Servlet

- **Lấy dữ liệu từ web client gửi đến bằng servlet**

- Dùng đối tượng của class *HttpServletRequest*

- Các methods để lấy thông số:

- *getParameter("para-name")*

- *getParameterValues("para-name")*

```
String username =  
    request.getParameter("username");
```

```
String[] choice =  
    request.getParameterValues("comments");
```

- Dùng đối tượng của class *HttpServletRequest* để lấy các thông tin HTTP header

- Ví dụ: Request Parameter



# Lập trình Web với Servlet

- **Lấy các thông số HTTP request header**
  - class *HttpServletRequest* cung cấp các method để lấy các thông số request header.
  - *String getHeader(header-name)*: lấy nội dung của header-name
  - *Enumeration getHeaderNames()*: lấy tất cả các header-name.
  - Một số method điển hình:
    - *Cookie[] getCookies()*: dãy Cookie từ client
    - *int getContentLength()*: trả giá trị Content-Length
    - *int getContentType()*: trả giá trị Content-Type
    - *int getRemoteUser()*: giá trị username nếu có authenticate
  - Ví dụ: Request Header



# Lập trình Web với Servlet

- **Lấy các thông số HTTP request header (t.t)**

- Lấy các giá trị của biến môi trường CGI:
  - QUERY\_STRING: *getQueryString()*
  - REMOTE\_ADDR: *getRemoteAddr()*
  - REMOTE\_HOST: *getRemoteHost()*
  - REQUEST\_METHOD: *getMethod()*
  - PATH\_INFO: *getPathInfo()*
  - SCRIPT\_NAME: *getServletPath()*
  - SERVER\_NAME: *getServerName()*
  - SERVER\_PORT: *getServerPort()*
  - HTTP\_XXX\_YYY: *getHeader("Xxx-Yyy")*
  - ...

- Ví dụ : [Request Info](#)





# Lập trình Web với Servlet

- **API cho HttpServletResponse**

1. `String getCharaterEncoding()`
2. `ServletOutputStream getOutputStream()`
3. `PrintWriter getWriter()`
4. `setContentLength( int len )`
5. `setContentType( String type )`
6. `setBufferSize( int len )`
7. `int getBufferSize()`
8. `flushBuffer()`
9. ...



# Lập trình Web với Servlet

- **Gửi dữ liệu cho web client**
  - Dùng đối tượng của class *HttpServletResponse*.
  - Tạo đối tượng *PrintWriter* để ghi dữ liệu gửi
    - `PrintWriter out = response.getWriter();`
  - Xử lý các mã HTTP trả về với các method của class *HttpServletResponse*:
    - `void setStatus(int statusCode)`: gửi các mã response
    - `void sendError(int errorCode, String msg)`: gửi mã lỗi theo giao thức HTTP và message
    - `void sendRedirect(String URL)`: chuyển đến một trang URL khác
    - Các mã có thể theo giao thức HTTP hoặc dùng các hằng số trong class *HttpServletResponse*.



# Lập trình Web với Servlet

- **Xử lý Cookie với web browser**
  - Chức năng Cookie
    - Kết hợp với web browser để lưu các thông số cần thiết.
    - Thông tin có thể dùng để thiết lập phiên làm việc (session) trong các ứng dụng thương mại điện tử(e-commerce).
    - Lưu trữ username, password
    - Thông tin để customize web site cho user hay dùng cho cơ chế personalization
    - ...



# Lập trình Web với Servlet

- **Ghi thông tin Cookie lên máy client**
  - Thông tin được truyền đi trong field `Set-Cookie` HTTP header
  - Dùng method `addCookie(Cookie cookie)` của class `HttpServletResponse`.
  - Các thuộc tính quan trọng trong class `Cookie`:
    - Cookie name: `setName(String name) – getName()`
    - Cookie value: `setValue(String value) – getValue()`
    - Max Age: `setMaxAge(int seconds) – getMaxAge()`
  - Ví dụ: Cookies



# Lập trình Web với Servlet

- Ví dụ

```
String user="",pass="";
Cookie[] cookies = request.getCookies();
if(cookies.length==0){
    user = request.getParameter("username");
    pass = request.getParameter("password");
    Cookie name_cookie = new Cookie("username",user);
    response.addCookie(name_cookie);
    Cookie pass_cookie = new Cookie("password",pass);
    response.addCookie(pass_cookie);
}
else{
    for(int i=0;i<cookies.length;i++){
        Cookie cookie = cookies[i];
        if(cookie.getName().equals("username"))
            user=cookie.getValue();
        if(cookie.getName().equals("password"))
            pass=cookie.getValue();
    }
}
```



# Lập trình Web với Servlet

- Lưu thông tin về phiên làm việc của user
  - class *HttpSession*.
  - Có thể dùng để lưu bất kỳ đối tượng nào.
  - Đối tượng của class *HttpSession* được trả về từ method *getSession()* của class *HttpServletRequest*.
  - Các method thường sử dụng:
    - *Object getValue(String name)* [2.2: *getAttribute*]
    - *void putValue(String name, Object object)* [2.2: *putAttribute*]
    - *void removeValue(String name)* [2.2: *removeAttribute*]
    - *String[] getValueNames()* [*Enumeration getAttributeNames()*]
    - *String getId()*
    - *void setMaxInactiveInterval(int seconds)*
  - Ví dụ: Session



# Lập trình Web với Servlet

- Ví dụ lưu ShoppingCart vào session

```
HttpSession session = request.getSession(true);
ShoppingCart cart =
    (ShoppingCart)session.getValue("ShoppingCart");
if(cart==null){
    cart = new ShoppingCart();
    session.putValue("ShoppingCart", cart);
}
//process(cart)
```



# Lập trình Web với Servlet

- **Kết nối cơ sở dữ liệu**
  - Dùng JDBC (Java DataBase Connectivity) để kết nối và thao tác với database.
  - Quy trình xử lý:
    - Tạo JDBC driver và URL database.
    - Thiết lập connection đến URL database.
    - Tạo đối tượng statement.
    - Thực thi các câu lệnh SQL.
    - Xử lý kết quả thực thi.
    - Đóng kết nối đến database.





# Lập trình Web với Servlet

- **Kết nối đến database thông qua ODBC.JDBC trên Windows**

- Tạo DataSourceName trong ODBC.JDBC

```
Connection con = null;
Statement stmt = null;
ResultSet rs = null;
String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
String databaseURL = "jdbc:odbc:DataSourceName";
try{
    Class.forName(driver);
    con = DriverManager.getConnection(databaseURL);
    stmt = con.createStatement();
    rs = stmt.executeQuery("SELECT * FROM Products");
    while(rs.next()){
        out.println(rs.getString[1]+"-"
            getInt("quantity")); //...
    }
    con.close();
}catch(SQLException se){ con.close(); }
```



# Lập trình Web với Servlet

- **Kết nối database server bất kỳ**

- Cần có driver hỗ trợ.

- Kết nối đến Oracle Database Server:

```
driver="oracle.jdbc.driver.OracleDriver"
```

```
databaseURL ="jdbc:oracle:thin@localhost:1521:" + dbName
```

```
con = DriverManager.getConnection(databaseURL, user, password)
```

- Kết nối đến Sysbase:

```
driver="com.sysbase.jdbc.SysDriver"
```

```
databaseURL=
```

```
    "jdbc:sysbase:Tds:localhost:1521?SERVICENAME=" + dbName
```

```
con = DriverManager.getConnection(databaseURL, user, password)
```



## Giới thiệu về JSP

- Là một trang HTML có xen các đoạn mã Java.
- Được web server biên dịch thành bytecode.
- Cơ chế hoạt động tương tự như Servlet.
- Được tạo sẵn các đối tượng của các class *HttpServletRequest* và *HttpServletResponse* để xử lý giao tiếp với web client.

# Lập trình web với JSP

- **Các đối tượng được xây dựng sẵn**
  - *request*: đối tượng class *HttpServletRequest*
  - *response*: đối tượng class *HttpServletResponse*
  - *out*: đối tượng class *PrintWriter*
  - *session*: đối tượng class *HttpSession* tạo ra từ method *getSession()* của đối tượng *request*
  - *application*: đối tượng class *ServletContext*
  - *config*: đối tượng class *ServletConfig*



# Ví dụ về lập trình web với JSP

```
<html>
<body bgcolor="white">
<h1> Request Information </h1>
<font size="4">
JSP Request Method: <%= request.getMethod() %>
<br>
Request URI: <%= request.getRequestURI() %>
<br>
Request Protocol: <%= request.getProtocol() %>
<br>
Servlet path: <%= request.getServletPath() %>
<br>
Path info: <%= request.getPathInfo() %>
<br>
Path translated: <%= request.getPathTranslated() %>
<br>
</body>
</html>
```

# Lập trình web với JSP

- **Các đoạn mã Java trong trang JSP được đặt trong tag**  
`<% .... %>`
  - Giá trị biểu thức: `<%= biểu thức %>`
  - Khai báo các biến `<%! [type variable;]+ %>`
  - Java code `<% Java code %>`
  - Chú thích `<%-- comments --%>`
- **Các khai báo chỉ thị (directive)**
  - `<%@ page import="[package][,package]*" %>`
  - `<%@ page isThreadSafe="{truefalse}" %>`
  - `<%@ page session="{truefalse}" %>`
  - `<%@ include file="filename" %>`



## Tạo JSP dùng JSP wizard (JBuilder)

- Tạo một Project (`jsptutorial`)
- Chọn một server (Tomcat 5.0)
- Tạo một Web module (`jspwebmodule`)
- Tạo JSP(`JSPWithCounter`)
- Chỉnh sửa trên JavaBean
- Thay đổi JSP code
- Thực thi JSP
- Deploy JSP



# Tạo JSP dùng JSP wizard (JBuilder)

- Tập tin JSPWithCounterBean.java

```
package jsptutorial;
public class JSPWithCounterBean {
    /**initialize variable here*/
    private int myCount=0;
    private String sample = "Start value";
    // Access sample property
    public String getSample() { return sample; }
    // Access sample property
    public void setSample(String newValue) {
        if (newValue!=null) { sample = newValue; }
    }
    /**New method for counting number of hits*/
    public int count() {
        return ++myCount;
    }
}
```





# Tạo JSP dùng JSP wizard (JBuilder)

- **Tập tin JSPWithCounter.jsp**

```
...
<body>
<h1>
JBuilder Generated JSP
</h1>
<form method="post" action="JSPWithCounter.jsp">
<br>Enter new value: <input name="sample"><br>
<br><br>
<input type="submit" name="Submit" value="Submit">
<input type="reset" value="Reset">
<br>
<p>This page has been visited: <%= jSPWithCounterBeanId.count () %>
times.</p>
</form>
</body>
</html>
```



## Giới thiệu về ASP

- Là một trang HTML có chứa các mã script (VBScript hay JavaScript).
- Các script này có thể:
  - Lấy thông tin từ user.
  - Sinh nội dung động.
  - Thao tác với database.
  - ...
- Trang ASP được chạy trên web-server có hỗ trợ (server-side)



# Giới thiệu về ASP

- **Các công cụ**
  - **Soạn thảo**
    - Có thể dùng trình soạn thảo văn bản text bất kỳ.
    - Các công cụ trực quan : Ms. Visual InterDev, Ms. FrontPage, Macromedia Dreamweaver...
  - **Web server**
    - Win9x : Personal Web Server
    - Win NT/2000 : Internet Information Services.
    - Chili!Soft : <http://www.ChiliSoft.com> (UNIX support)
    - HalcyonSoft : <http://www.halcyonsoft.com>

# Lập trình web với ASP

- **Thêm script xử lý vào trang ASP**

- Đặt mã trong cặp dấu : `<%...%>`

- Dùng tag `SCRIPT`

- Ví dụ

```
<% @language=jscript %>
<%
// cong voi "" de dam bao hovaten la kieu chuoii
hovaten="" + Request("HoTen");
%>
<HTML>
<BODY>
<% if (hovaten != "undefined") { // Greeting duoc yeu cau
tu form %>
<FONT SIZE=4 COLOR="#0000FF">
Xin chao ban: <%=hovaten %>
</FONT><BR>
<% } %>
```

# Lập trình web với ASP

- **Thêm script xử lý vào trang ASP (tiếp theo)**

```
<FORM ACTION="greeting.asp" METHOD="Get">
```

Xin cho biết họ tên của bạn:

```
<INPUT NAME="HoTen" TYPE="text">
```

Hãy bấm nút SUBMIT:

```
<INPUT NAME="button1" TYPE="submit"  
  VALUE="SUBMIT">
```

```
</FORM>
```

```
</BODY>
```

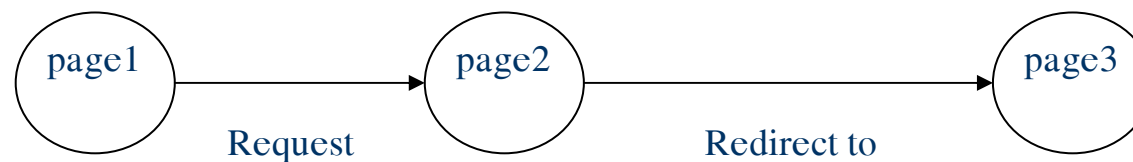
```
</HTML>
```

# Lập trình web với ASP

- **Đối tượng xử lý request là Request**
  - Lấy thông tin người dùng nhập từ Form với POST
    - Request.Form(“field\_name”)
    - Request.Form(“field\_name”)(index)
    - Request.Form(“field\_name”).Count
  - Lấy thông tin người dùng nhập từ Form với GET
    - Request(“field\_name”)
    - Request.QueryString(“field\_name”)(index)
    - Request.QueryString(“field\_name”).Count
  - *Lưu ý về một số field đặc biệt:*
    - checkbox, radio, các field đặt cùng tên...

# Lập trình web với ASP

- Đối tượng xử lý response là **Response**
  - Các phương thức chính
    - Response.Write("String")
    - Response.Redirect("URL")



# Lập trình web với ASP

- Sử dụng **Cookie**

- Thiết lập giá trị Cookie

**Response.Cookies(cookie)[(key).attribute] = value**

- Nhận các giá trị Cookie

**Request.Cookies(cookie)[(key).attribute]**

Ví dụ:

```
<%
```

```
Response.Cookies("firstname")="Alex"
```

```
Response.Cookies("user")("firstname")="John"
```

```
Response.Cookies("user")("lastname")="Smith"
```

```
Response.Cookies("user")("country")="Norway"
```

```
Response.Cookies("user")("age")="25"
```

```
%>
```



# Lập trình web với ASP

- **Đối tượng Session**

- Quản lý thông tin về phiên làm việc(session).
- Có thể lưu dữ liệu, đối tượng của một session.
- Biến trong Session:

*Session("Ten\_Bien")=Gia\_Tri\_Khoi\_Tao;*

*Ten\_Bien = Session ("Ten\_Bien");*

- Ví dụ

```
<%
```

```
Session("username")="Donald Duck"
```

```
Session("age")=50
```

```
%>
```

# Lập trình web với ASP

- **Đối tượng *Application***
  - Quản lý thông tin về ứng dụng.
  - Có thể dùng để lưu trữ dữ liệu, đối tượng.
  - Chia sẻ giữa các người sử dụng.
  - *Khởi tạo:*  
*Application("Ten\_Bien")=Gia\_Tri;*
  - *Thao tác cập nhật:*
    - Gọi *Application.Lock()*;
    - Thực hiện các thao tác với các biến *Application*.
    - Gọi *Application.Unlock()*;



# Lập trình web với ASP

- **Đối tượng `Server`**
  - Làm việc với các thành phần server.
  - Có hai phương thức thường dùng của đối tượng này là:
    - Phương thức `MapPath(URL)`: Xác định đường dẫn vật lý của trang được đặc tả bởi URL.
    - Phương thức `CreateObject(progID)` dùng để tạo một thể hiện của thành phần COM server.



# Lập trình web với ASP

- **Thao tác với CSDL**

- Có thể dùng ADO để thao tác.
- Ví dụ về đọc dữ liệu từ CSDL:

```
Dim objRecordset
Set objRecordset =
    Server.CreateObject("ADODB.Recordset")
objRecordset.Open "table-name", "DSN=dsn"
Do While NOT objRecordset.EOF
    Response.Write objRecordset("field-name")
    objRecordset.MoveNext
Loop
```



# Lập trình web với ASP

- **Đọc dữ liệu bằng câu lệnh SQL**

```
strSQL = "Select * FROM table WHERE ..."
```

```
objRecordset.Open strSQL, "DSN=dsn"
```

- In dữ liệu theo dạng bảng, danh sách, listbox...
- Có thể dùng cách này để hiện thực việc tìm kiếm.

- **Cập nhật thông tin trong record**

```
strSQLUpdate = "UPDATE table SET  
field1=..., field2=... WHERE ..."
```

```
objRecordset.Excute strSQLUpdate
```



# Lập trình web với ASP

- **Thêm mới record**

```
Set obj-Recordset-name =  
Server.CreateObject ("ADODB.Recordset")  
objRecordset-name.Open "table-name",  
    "DSN=dsn", _ adOpenDynamic,  
    adLockOptimistic  
obj-Recordset-name.AddNew  
obj-Recordset-name.Fields ("field-name")  
= ...  
obj-Recordset-name.Update  
\ ...  
obj-Recordset-name.Close
```



# Tổng kết

- Lập trình Web với CGI
  - Perl
  - Apache Web Server 2.0
- Lập trình Web với Servlet
  - Java
  - Apache Tomcat 4.1
- Lập trình Web với JSP
  - Java
  - Apache Tomcat 4.1
- Lập trình Web với PHP
  - PHP
  - Apache & MySQL
- Lập trình Web với ASP
  - Jscript
  - IIS 5.1



## Bài tập: Ứng dụng tra cứu DBĐT bằng servlet

- Viết một **InputForm** servlet để tạo một HTML form nhập để nhận dữ liệu tra cứu danh bạ điện thoại.
- Viết một **SearchAccount** servlet để tra cứu thông tin danh bạ.





# Viết servlet InputForm

```
1. import javax.servlet.*; import javax.servlet.http.*;
2. import java.io.IOException;

3. public class InputForm extends HttpServlet {
4.     static final private String CONTENT_TYPE = "text/html";
5.     //Initialize global variables
6.     public void init() throws ServletException {
7.         // TO DO ...
8.     }

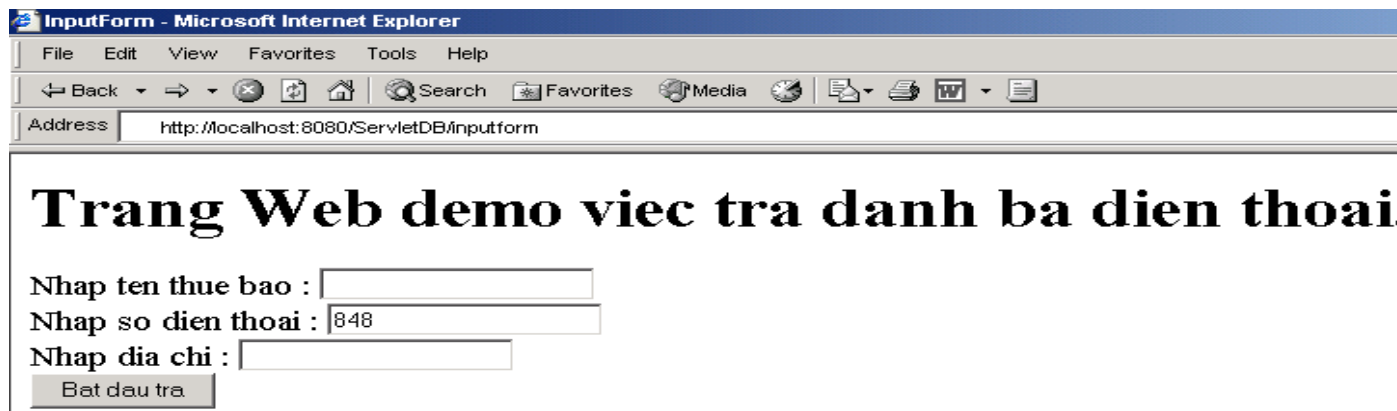
9.     public void doGet (HttpServletRequest request,
10.                        HttpServletResponse response)
11.        throws ServletException, IOException {
12.         // TO DO ...
13.     }
14. }
```



# Viết servlet InputForm (t.t)

```
1. //Process the HTTP Get request
2. public void doGet ( HttpServletRequest request, HttpServletResponse
   response)
3.         throws ServletException, IOException {
4.     response.setContentType(CONTENT_TYPE);
5.     PrintWriter out = response.getWriter(); // Write to client
6.     out.println("<html><head><title>InputForm</title></head>");
7.     out.println("<body>");
8.     out.println("<h1>Trang Web demo viec tra danh ba dien thoai.</h1>");
9.     out.println("<form action='searchaccount' method='post'>");
10.    out.println("Nhap ten thue bao: <INPUT NAME='ttbao'><br>");
11.    out.println("Nhap so dien thoai: <INPUT NAME='sodthoai'><br>");
12.    out.println("Nhap dia chi: <INPUT NAME='diachi'> <br>");
13.    out.println("<input type='submit' name='Submit' value='Bat dau tra'>");
14.    out.println("</form>"); out.println("</body></html>");
15. }
```

# Kết quả hiển thị trên trình duyệt



The screenshot shows a Microsoft Internet Explorer browser window titled "InputForm - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/ServletDB/inputform". The main content area contains the following text and form elements:

**Trang Web demo viec tra danh ba dien thoai.**

Nhap ten thue bao :

Nhap so dien thoai :

Nhap dia chi :

```
<html>
<head><title>InputForm</title></head>
<body> <h1>Trang Web demo viec tra danh ba dien thoai.</h1>
<form action=searchaccount method='post'>
    Nhap ten thue bao : <input name='ttbao' ><br>
    Nhap so dien thoai : <input name='sodthoai' ><br>
    Nhap dia chi      : <input name='diachi' ><br>
    <input type='submit' name='Submit' value='Bat dau tra'>
</form>
</body></html>
```



# Viết servlet SearchAccount

```
1. import java.io.*; import java.util.*; import java.sql.*;
2. public class SearchAccount extends HttpServlet {
3.     static final private String CONTENT_TYPE = "text/html";
4.
5.     // Process the HTTP Post request
6.     public void doPost (HttpServletRequest request,
7.                         HttpServletResponse response)
8.         throws ServletException, IOException
9.     {
10.        // TODO...
11.        // Lay gia tri tu cac textbox tren InputForm
12.    }
13. }
```



# Viết servlet SearchAccount (t.t)

```
1. //Process the HTTP Post request
2. public void doPost (HttpServletRequest request, HttpServletResponse response)
3.     throws ServletException, IOException {
4.     response.setContentType(CONTENT_TYPE);
5.     PrintWriter out = response.getWriter();
6.     out.println("<html><head><title>SearchAccount</title></head>");
7.     out.println("<body><h1>Ket qua tra dien thoai theo yeu cau cua ban :</h1>");
8.     out.println("<table border=1 cellPadding=1 cellSpacing=1>");
9.     String tentbao = request.getParameter("ttbao");

10.    // xay dung lenh SQL
11.    String newSQL = "SELECT * FROM CUSTOMER";
12.    if (tentbao != null && tentbao.length() != 0) {
13.        newSQL = newSQL + " where tentbao like '%" + tentbao + "%'";
14.    }
15.    String conStr = "jdbc:odbc:DanhBaDT";
16.    Statement stmt = null;    ResultSet rs = null;
17.    .....
```

# Viết servlet SearchAccount

```

1. try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
2.     con = DriverManager.getConnection(conStr, "", "");
3.     stmt = con.createStatement();
4.     rs = stmt.executeQuery(newSQL);
5.
6.     out.println("<tr><th>So thu tu</th><th>Ten thue bao</th><th>So dien" +
7.         " thoai</th><th>Dia chi</th></tr>\n");
8.     if (rs != null) {
9.         for ( int i = 1; rs.next(); ) {
10.            out.println("<tr><td>" + i + "</td><td>" + rs.getString(2) +
11.                "</td><td>" + rs.getString(3) + "</td><td>" +
12.                rs.getString(4) + "</td></tr>\n");
13.        } // end for
14.    } // end if
15.    out.println("</table>");
16.    rs.close();    stmt.close();    con.close();
17. } catch (Exception e) {    System.out.println("Error : " + e); }
18. ...
19. } // End doPost()
    
```

# Kết quả chạy thử

**InputForm - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

← Back → Search Favorites Media

Address <http://localhost:8080/ServletDB/inputform>

**Trang Web demo viec tra danh ba dien thoai.**

Nhap ten thue bao :

Nhap so dien thoai :

Nhap dia chi :

**SearchAccount - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address <http://localhost:8080/ServletDB/searchaccount>

**Ket qua tra dien thoai theo yeu cau cua ban :**

So thu tu	Ten thue bao	So dien thoai	Dia chi
1	Xi Muoi	08-8481234	123 Ly Thuong Kiet
2	Nguyen Quang Hung	84-8481235	234 Lac Long Quan

[Quay ve trang chu](#)