



Bài 1

Giới Thiệu Ứng Dụng Web Tìm Hiểu Java Servlet

- ❖ Giải thích thế nào là một ứng dụng Web
- ❖ Xu hướng sử dụng ứng dụng web trong các công ty, doanh nghiệp.
- ❖ Phát triển ứng dụng web trong Java với JSP Servlet
- ❖ Vòng đời của một ứng dụng web
- ❖ Giải thích các thành phần của servlet
- ❖ Điều khiển luồng dữ liệu từ form với servlet và tới các trang jsp khác.

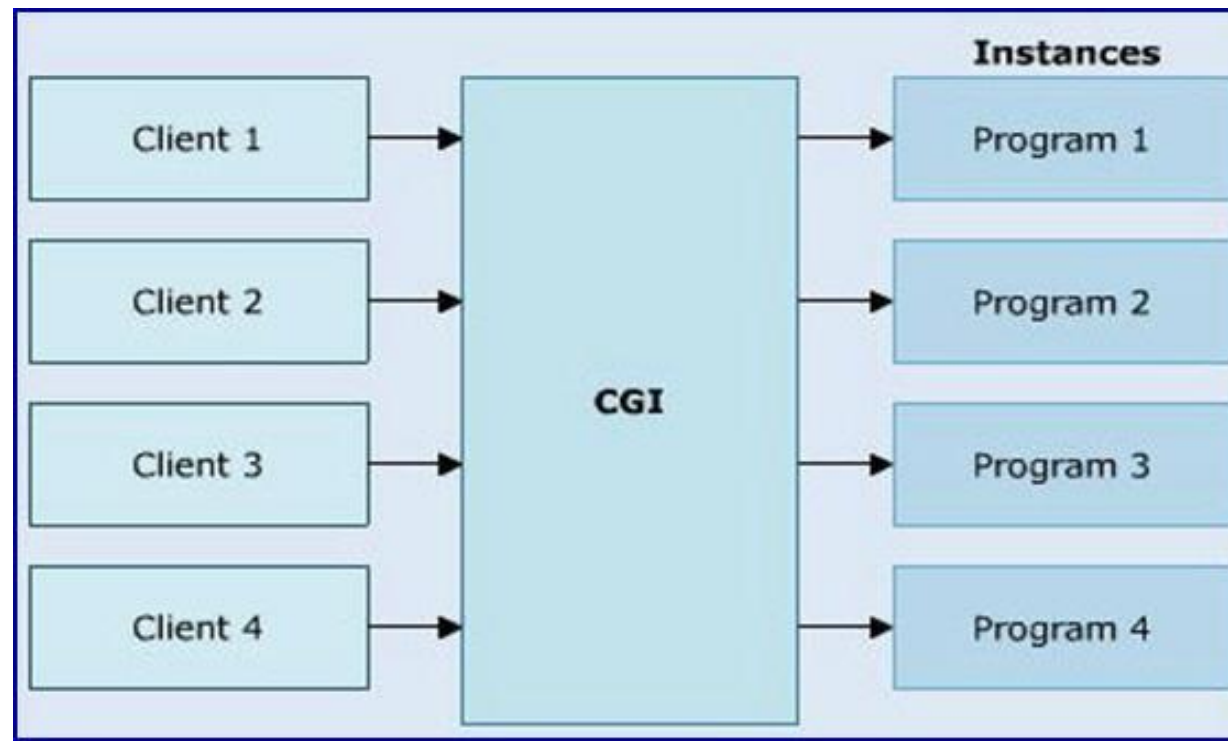
- ❖ Ứng dụng web là một ứng dụng chạy trên web server
- ❖ Bao gồm các trang web động hoặc các trang web tĩnh.
- ❖ Các trang web tĩnh được tạo sử dụng các kỹ thuật web như HTML, CSS, Javascript.
- ❖ Các trạng web động hiển thị các nội dung động, cho phép tương tác với dữ liệu đầu vào của người dùng hoặc tương tác với dữ liệu từ các hệ quản trị CSDL.

- ❖ Phần mềm để các nhân viên trong các doanh nghiệp sử dụng trong công việc hàng ngày được gọi là ứng dụng doanh nghiệp. Ngày nay hầu hết các ứng dụng doanh nghiệp đều là các ứng dụng web.
- ❖ Sử dụng ứng dụng web trong các doanh nghiệp có các lợi ích sau:
 - Chỉ cần cài đặt triển khai trên server, không phải cài đặt trên tất cả các máy nhân viên
 - Dữ liệu xử lý tập trung, dễ dàng cho doanh nghiệp tính toán, thống kê và đưa ra các giải pháp phù hợp để mang lại hiệu quả cao hơn.
 - Dễ dàng mở rộng các chi nhánh không giới hạn về khoảng cách địa lý mà vẫn quản lý và điều hành được tập trung.
 - Tiếp cận với khách hàng dễ hơn thông qua môi trường internet.

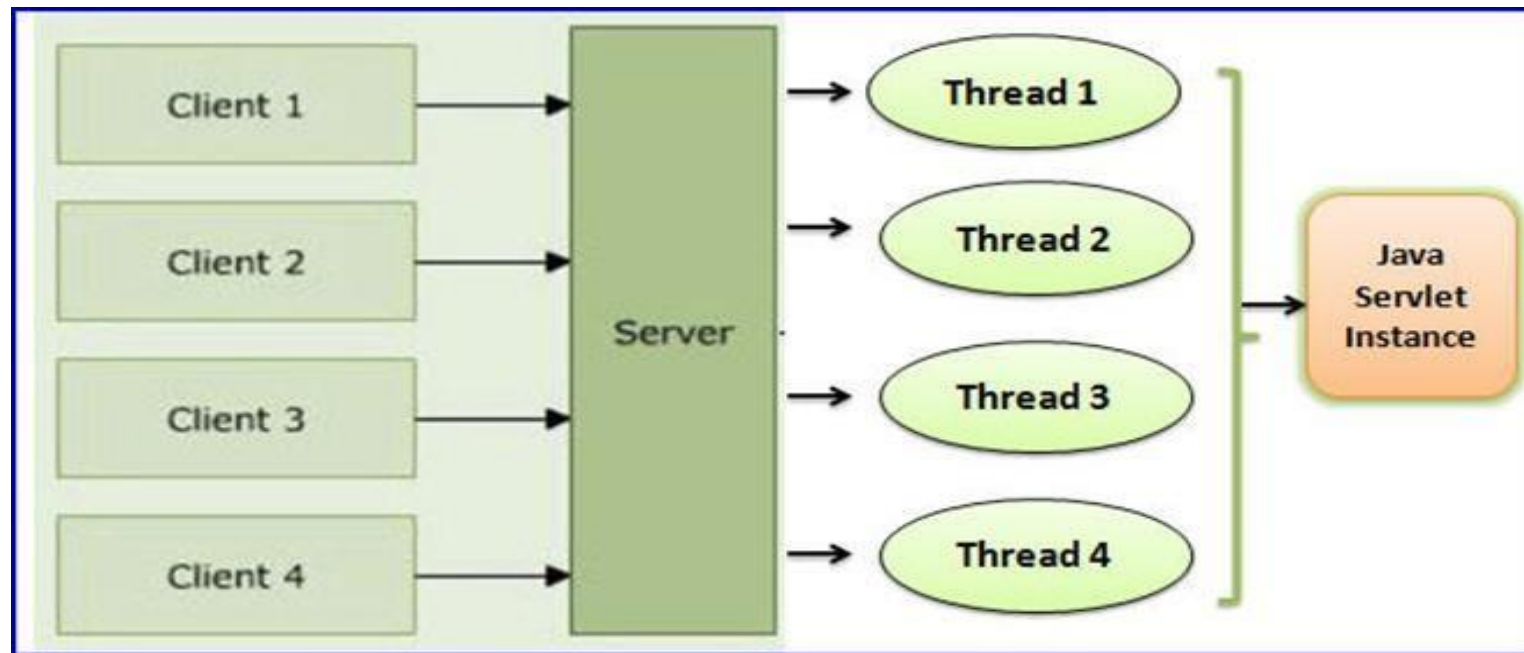
- ❖ **Nền tảng của ứng dụng web trong Java là JSP và Servlet**
- ❖ **JSP là các trang hiển thị để tương tác với người dùng, bao gồm các thẻ HTML, các định dạng css, Javascript, các mã lệnh Java nhúng vào gọi là các scriptlet, các thẻ chuẩn, thẻ tùy biến của Java.**
- ❖ **Servlet là các lớp mà xử lý các yêu cầu của người dùng, gửi các kết quả phản hồi và thực thi tất cả các tính toán của ứng dụng web bên server.**

- ❖ Trước khi sử dụng Servlet để xử lý các yêu cầu của người dùng. Bên phía server người ta sử dụng kỹ thuật được gọi là CGI. Đặc điểm của CGI là mỗi khi nhận được một yêu cầu từ người dùng, nó sẽ khởi tạo một chương trình độc lập, dịch và xử lý yêu cầu sau đó trả về kết quả tới máy khách.

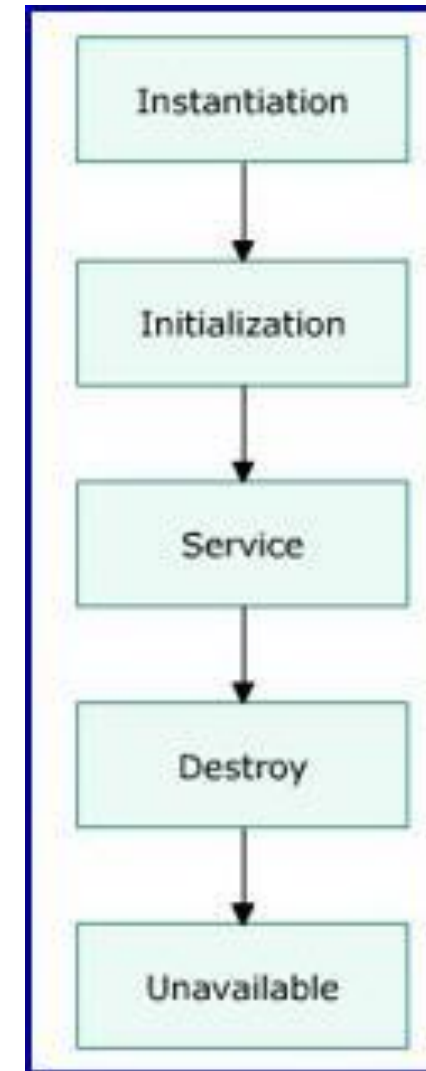
- ❖ Mô hình hoạt động của CGI



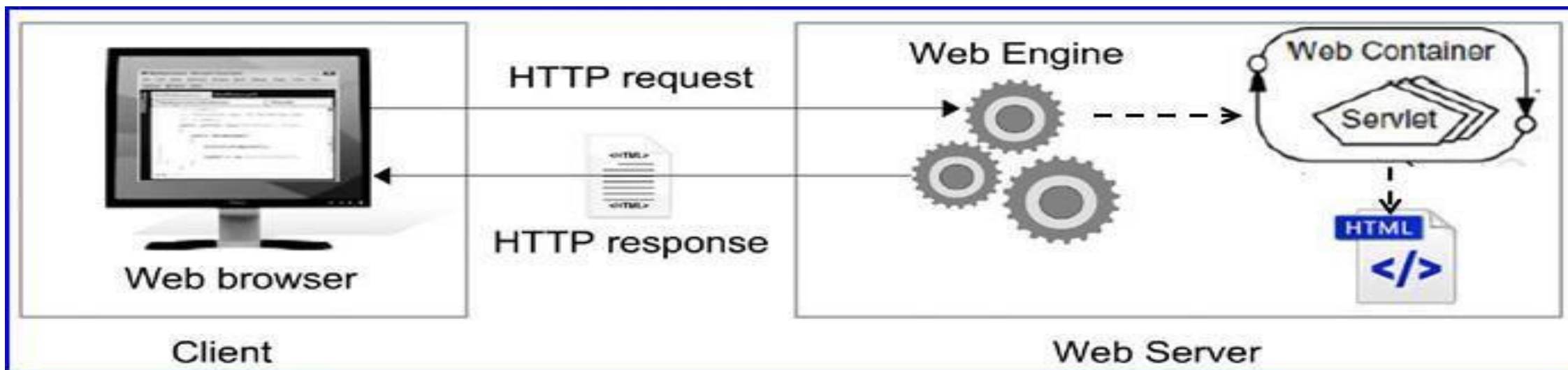
- ❖ Tương tự như xử lý như với CGI, nhưng servlet áp dụng kỹ thuật lập trình đa luồng (Thread) và nó chỉ cần khởi tạo các thể hiện của lớp mà không cần phải tạo thành một chương trình độc lập như CGI.
- ❖ Mô hình hoạt động của Servlet



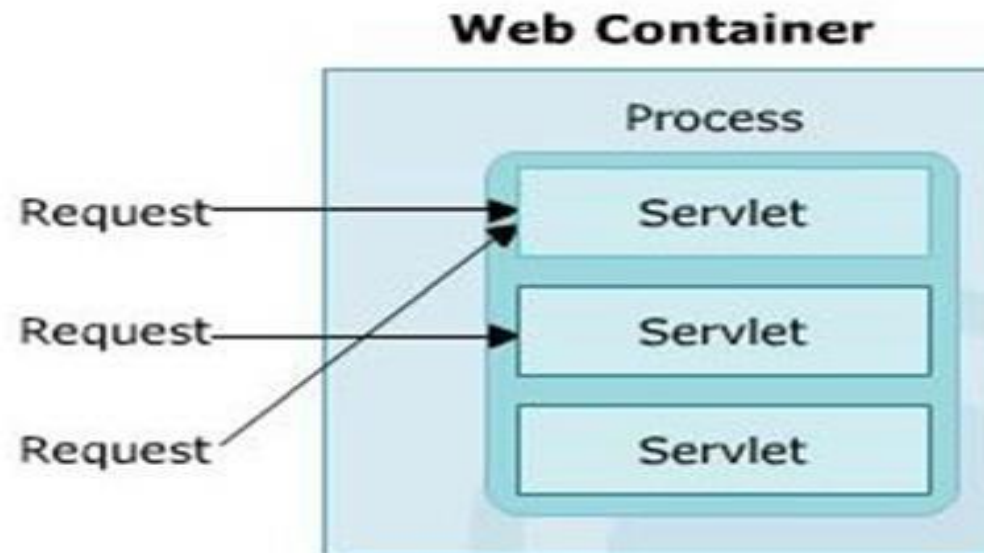
- ❖ Vòng đời của servlet được định nghĩa theo các phương thức:
- ❖ `init()`: Khởi tạo servlet
- ❖ `service()`: Xử lý yêu cầu của client
- ❖ `destroy()`: Hủy khởi tạo của servlet khi nó đã xử lý xong yêu cầu của client



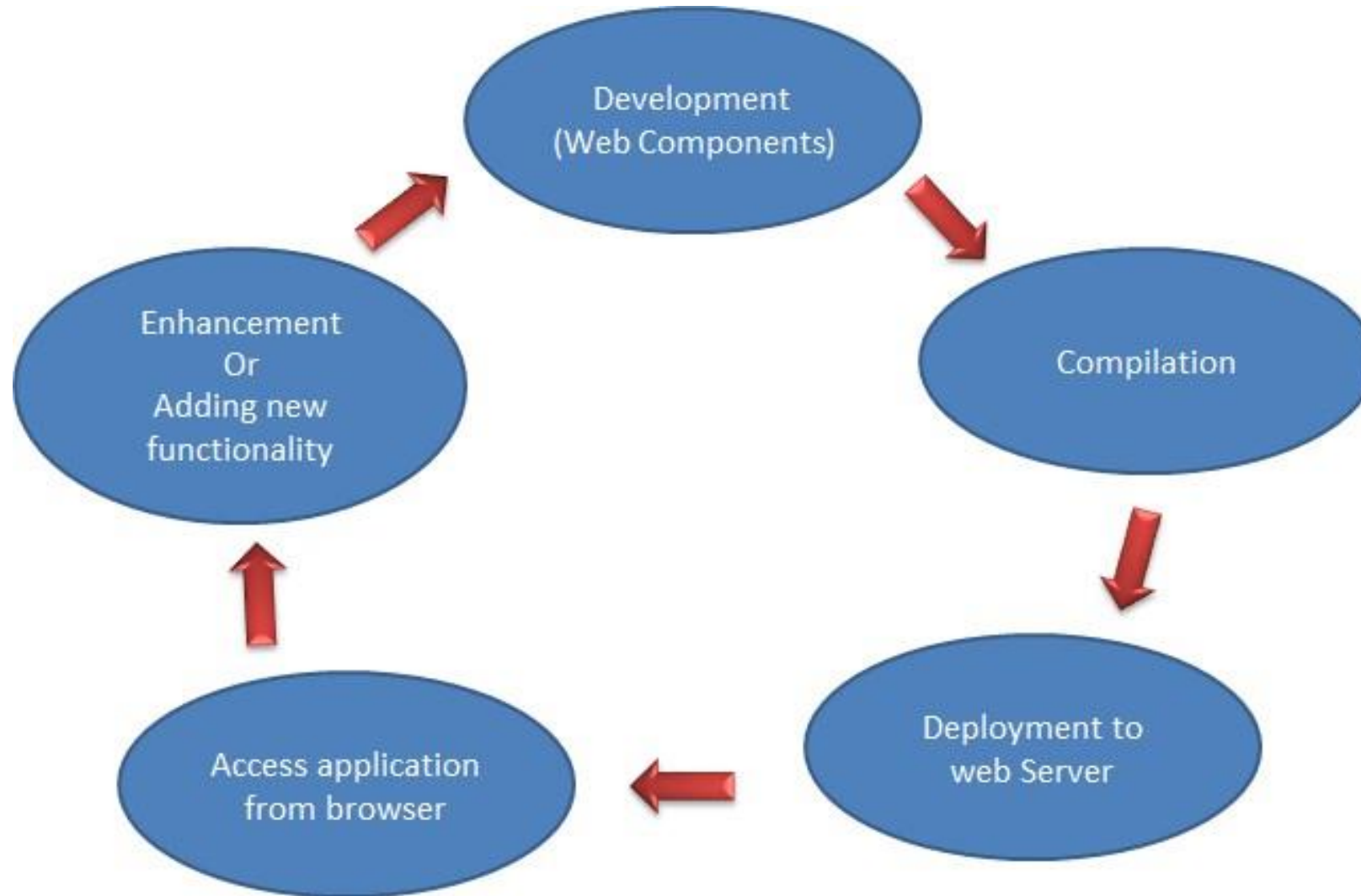
- ❖ Web server chứa một web container chịu trách nhiệm để quản lý các thành phần web ở trên server



- ❖ Quản lý việc thực thi của các servlet và các trang jsp
- ❖ Các trang jsp được dịch thành các mã lệnh của servlet, xử lý xong và tạo lại các trang jsp khác như là phản hồi của server tới client.



- ❖ Hình sau mô tả vòng đời của một ứng dụng web



❖ Là giao diện định nghĩa các phương thức để xử lý request từ client. Các phương thức được định nghĩa trong giao diện này:

- ❑ `public String getParameter(Strings)`
- ❑ `public java.lang.String[] getParameterValues(java.lang.Stringname)`
- ❑ `public java.util Enumeration getParameterNames()`
- ❑ `public String getAttribute(Strings)`
- ❑ `public int getContentLength()`
- ❑ `public ServletInputStream getInputStream() throws IOException`
- ❑ `public String getServerName()`

- ❖ Là giao diện định nghĩa các phương thức để xử lý request từ client sử dụng với giao thức Http.

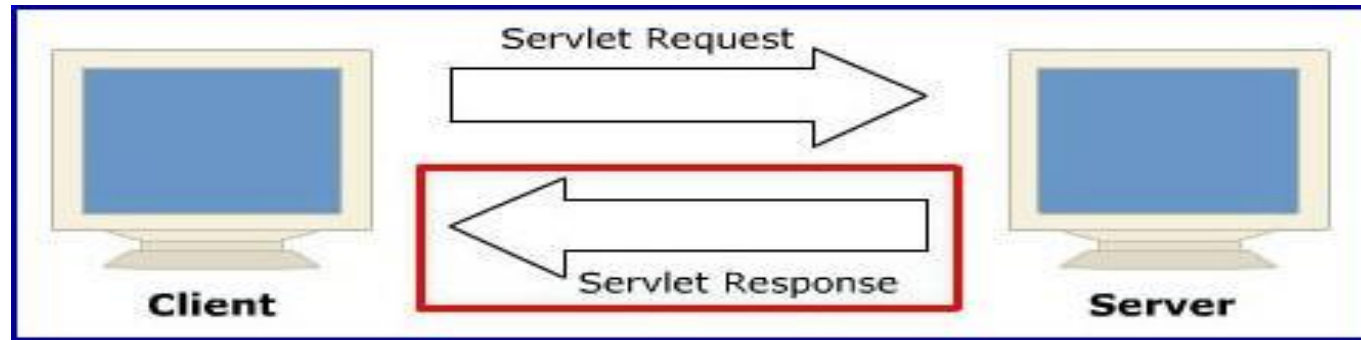
Các phương thức được định nghĩa trong giao diện này:

- ❑ `public Cookie[] get_cookies()`
- ❑ `public String getHeader (String name)`
- ❑ `public String getMethod()`
- ❑ `public String getPathInfo()`
- ❑ `public String getAuthType()`

- ❖ Dữ liệu request từ client được truyền đi theo giao thức Http. Theo giao thức này một số thông tin từ client được mang đi thêm dưới dạng các tiêu đề của HttpRequest.
- ❖ Thông tin của các tiêu đề HttpRequest có dạng như sau:



- ❖ Định nghĩa các phương thức để xử lý kết quả phản hồi về theo yêu cầu của client



- ❖ Các phương thức của giao diện này:

- `public java.lang.String getContentType()`
- `public PrintWriter getWriter() throws IOException`
- `public ServletOutputStream getOutputStream() throws IOException`
- `public void setContentType(java.lang.String str)`

- ❖ Kế thừa giao diện ServletResponse để truyền đi các dữ liệu phản hồi từ server tới client thông qua giao thức Http.
- ❖ Các phương thức của giao diện này:
 - `public void addCookie(Cookie cookie)`
 - `public void addHeader(java.lang.String name, java.lang.String value)`
 - `public boolean containsHeader(String name)`
 - `public boolean containsHeader(String name)`
 - Các phương thức để xử lý tiêu đề: `addHeader()` , `addDateHeader()` ,
`addIntHeader()` , `containsHeader()`

- ❖ Các tiêu đề của HttpResponse được gửi trả về cho client như sau:

```
$ curl -I searchengineland.com
HTTP/1.1 200 OK
Date: Sat, 06 Aug 2011 03:57:00 GMT
Server: Apache
Last-Modified: Sat, 06 Aug 2011 00:09:14 GMT
ETag: "374c333-13de3-4a9cb062e5e80"
Accept-Ranges: none
Content-Length: 81379
Cache-Control: max-age=300, must-revalidate
Expires: Sat, 06 Aug 2011 04:02:00 GMT
Vary: Accept-Encoding, Cookie
Content-Type: text/html; charset=UTF-8
```

❖ Điều hướng qua đối tượng request:

❖ `request.getRequestDispatcher("target").forward(request, response);`

- Cho phép điều hướng từ vị trí hiện tại đi đến thành phần “target” (có thể là các trang jsp, các servlet khác) và mang theo dữ liệu của các đối tượng request, response.

❖ `request.getRequestDispatcher("target").include(request, response);`

- Cho phép thêm vào dữ liệu của request và response từ thành phần “target” khác

❖ Điều hướng qua đối tượng response:

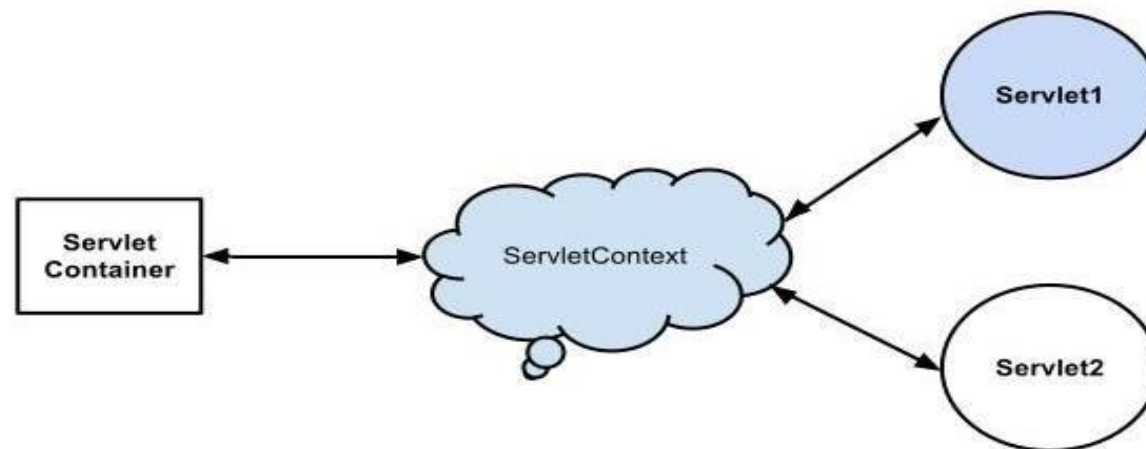
❖ `response.sendRedirect("target");`

- ❖ Gửi một phản hồi trực tiếp tới client theo vị trí của target

❖ `response.encodeRedirectURL(String url);`

- ❖ Mã hóa url để sử dụng cho phương thức `sendRedirect()`

- ❖ Cung cấp truy xuất tới tất cả các thành phần trên ứng dụng web và không giới hạn theo người dùng.



- ❖ Các phương thức của ServletContext:

- `public String getInitParameter(String name)`
- `public void setAttribute(String name, Object object)`
- `public Object getAttribute(String name)`
- `public Enumeration getInitParameterNames()`
- `public void removeAttribute(String name)`

HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn