Công nghệ lập trình J2EE

ts. Lâm Chí Nguyện

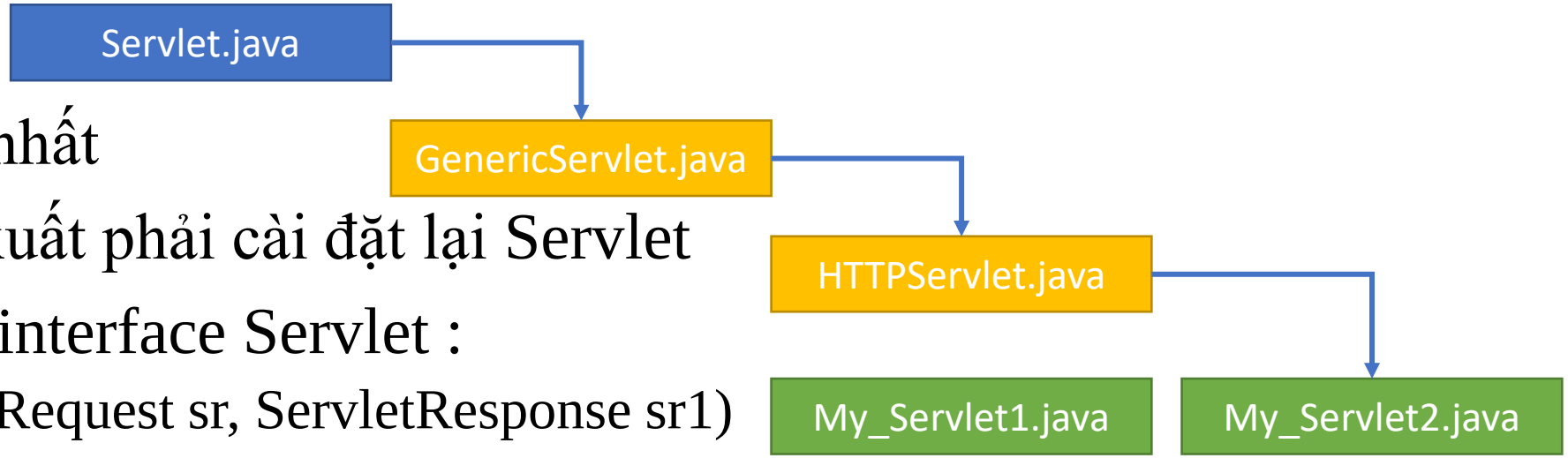
Chương Servlet

- Giới thiệu Servlet
- Giao diện **Servlet interface**
- Servlet Life cycle
- HTTPServlet
  - HTTPServletRequest
  - HTTPServletResponse
- My_Servlet
- Cấu hình Servlet trong web-container
- …..

Giới thiệu Servlet

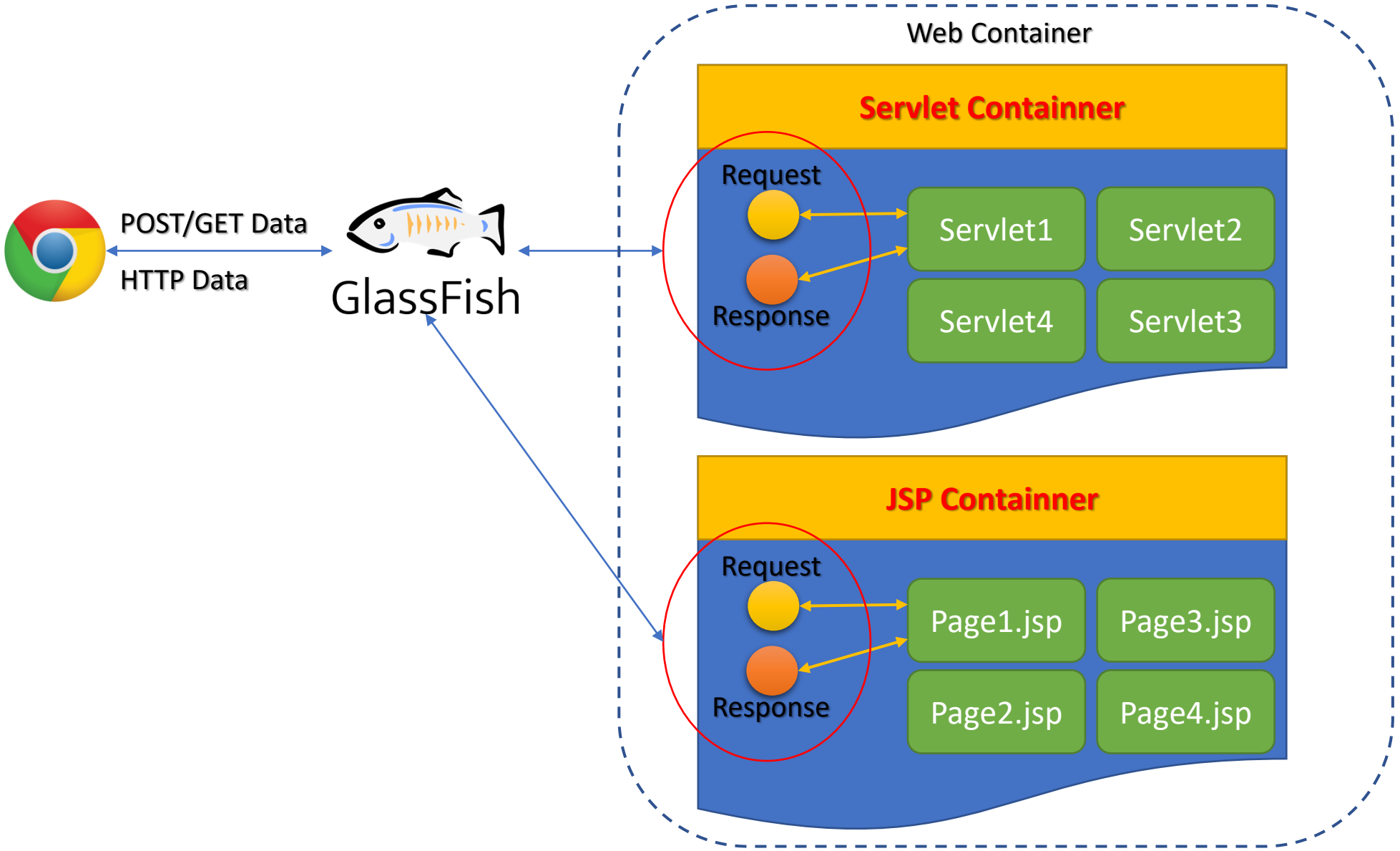
- Là một thành phần web dựa trên nền tảng các công nghệ Jakarta
- Servlet tạo ra các nội dung HTML động
- Các Servlet độc lập với các hệ điều hành
- Servlet tương tác với trình duyệt web (Webclient) thông qua hai khái niệm request/response.
- Servlet container :
  - Quản lý các servlet qua vòng đời của nó
  - Cung cấp các dịch vụ mạng mà ở đó có cài đặt sẵn các đối tượng (Request/Response)
  - Servlet container phải hỗ trợ giao thức HTTP/HTTPS

Giao diện Servlet

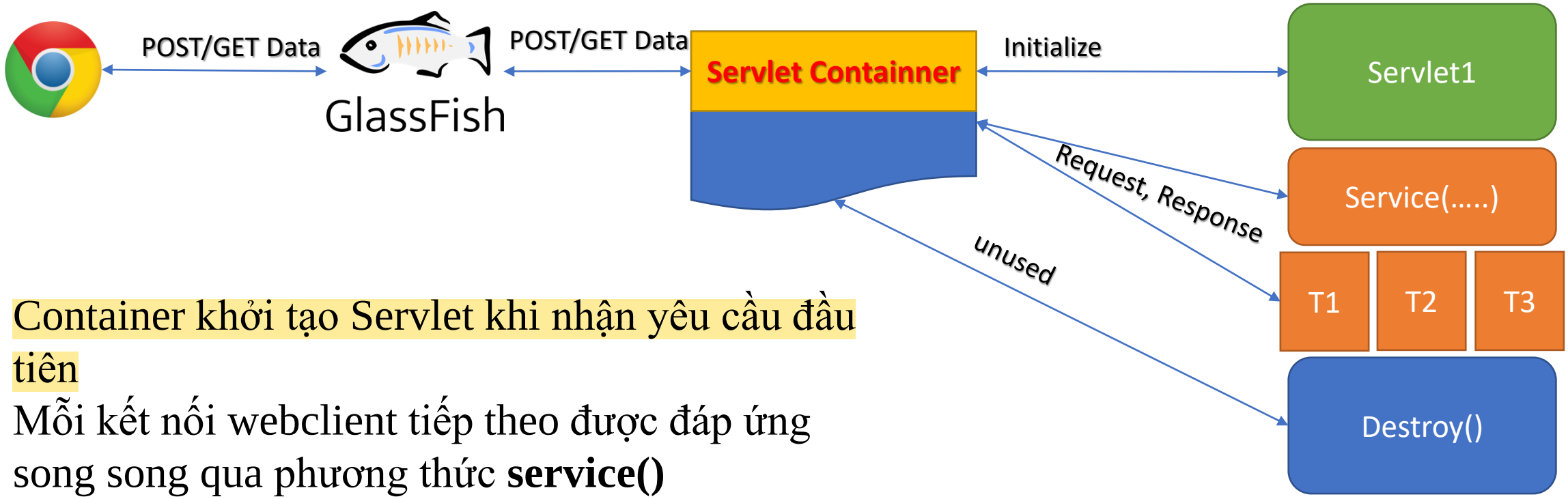


- Là giao diện cơ bản nhất
- Các lớp servlet dẫn xuất phải cài đặt lại Servlet
- Các hàm cơ bản của interface Servlet :
  - void **service**(ServletRequest sr, ServletResponse sr1)
  - String **getServletInfo**()
  - ServletConfig **getServletConfig**()
- Để có thể thiết kế servlet, ta thừa kế mở rộng từ **HTTPServlet**

Một hoạt vụ đơn giản



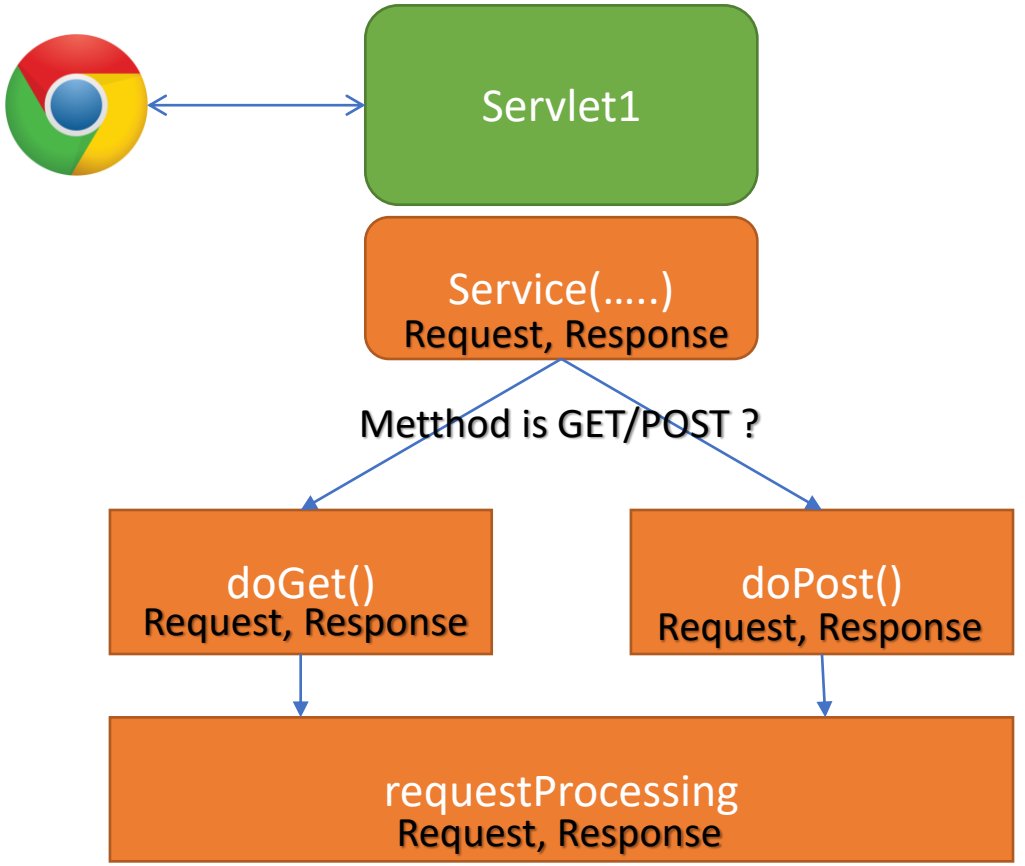
Servlet Life cycle



- Container khởi tạo Servlet khi nhận yêu cầu đầu tiên
- Mỗi kết nối webclient tiếp theo được đáp ứng song song qua phương thức **service()**
- Các dữ liệu giữa web server và servlet thông qua :
  - **Request :** dữ liệu gởi từ webclient
  - **Response :** nội dung được tạo ra bởi Servlet
- Khi không còn bất kỳ nối kết nào đến servlet thì container gọi phương thức **destroy()**

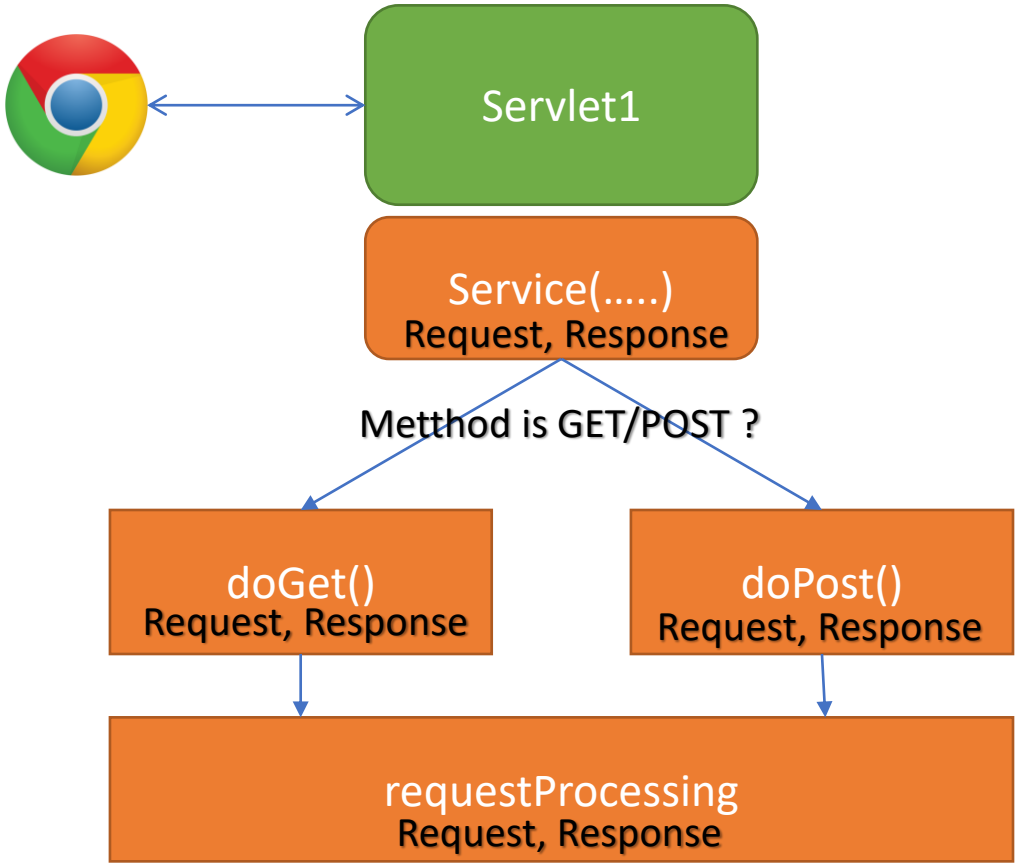
HTTPServlet

- HTTPServlet :
  - lớp abstract mô tả các phương thức của Servlet
  - sử dụng giao thức HTTP để truyền nhận dữ liệu
- Các phương thức abtract :
  - doGet()
  - doPost()
  - **requestProcessing()**
    - Kiểm tra dữ liệu nhận được
    - Xử lý các yêu cần cần thiết
    - Tạo nội dung gởi về webclient



HTTPServlet

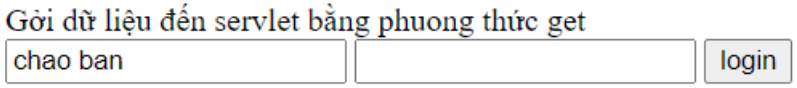
- Truyền nhận dữ liệu :
  - HTTPServletRequest – request
    - getParameterNames
    - getParameter
    - getParameterValues
    - getParameterMap
  - HTTPServletResponse – response
    - Thao tác trên Header
    - Thao tác trên Cookie
    - Thao tác trên nội dung (<body>)
    - Chuyển hướng đến trang khác



Data Form (HTML)

- Công cụ nhập dữ liệu và gởi dữ liệu đến server xử lý
- Các thẻ HTML dùng trong Form
  - <form> </form>
  - <input />
  - …

```
<form name = "login" method="POST" action="./Hello">
      <input type="text" name="username">
      <input type="password" name="password">
      <input type="submit" value="login">
</form>
```



login

Get Data Form From HTTPServletRequest

- request.getParameterNames() → cho một mảng tên của các tham số
- request.getParameter("ParameterName") → giá trị của một tham số
- request.getParameterValues("ParameterName") → các giá trị của một tham số

```
Enumeration parameters = request.getParameterNames();
while (parameters.hasMoreElements()) {
      String parameter = (String) parameters.nextElement();
      String value     = request.getParameter(parameter);
      //code here…..
}
```

login

Get Data Form From HTTPServletRequest

- request.getParameterMap()

```
Map m =request.getParameterMap();
Set s = m.entrySet();
Iterator it =s.iterator();
while (it.hasNext()){
        Map.Entry<String, String[]> entry;
        entry = (Map.Entry<String,String[]>)it.next();
        String key        = entry.getKey();
        String [] values = entry.getValue();
        //code here…
}
```

login

Get HTML Header From HTTPServletRequest

- request.getHeaderNames()
- request.getHeaders ()
- request.getHeaderValues()

```
 Enumeration headers = request.getHeaderNames();
while (headers.hasMoreElements()) {
      String header = (String) headers.nextElement();
      String values     = request.getHeaders(header);
      //code here…..
}
```

login

Tạo nội dung bằng HTTPServletResponse

- response.setContentType("text/html;charset=UTF-8");
- response.addHeader(key, values);
- request.getHeaderValues();
- response.getWriter();
  - Out.println();

Tạo một Servlet

- Tạo My_Servlet.java thừa kế từ HttpServlet

```
@WebServlet(urlPatterns = {"/My_Servlet"})
public class My_Servlet extends HttpServlet{
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

Tạo một Servlet

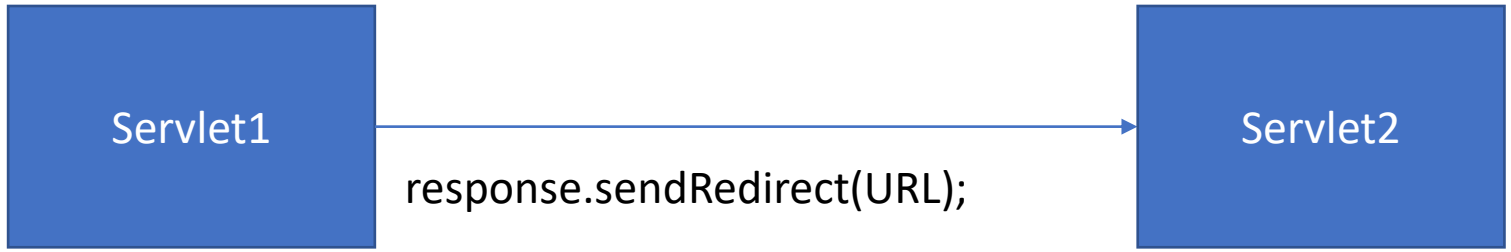
- Cấu hình Servlet trong web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <servlet>
        <servlet-name>vidu1</servlet-name>
        <servlet-class>vidu1</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>vidu1</servlet-name>
        <url-pattern>/Hello</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

Tuần sau

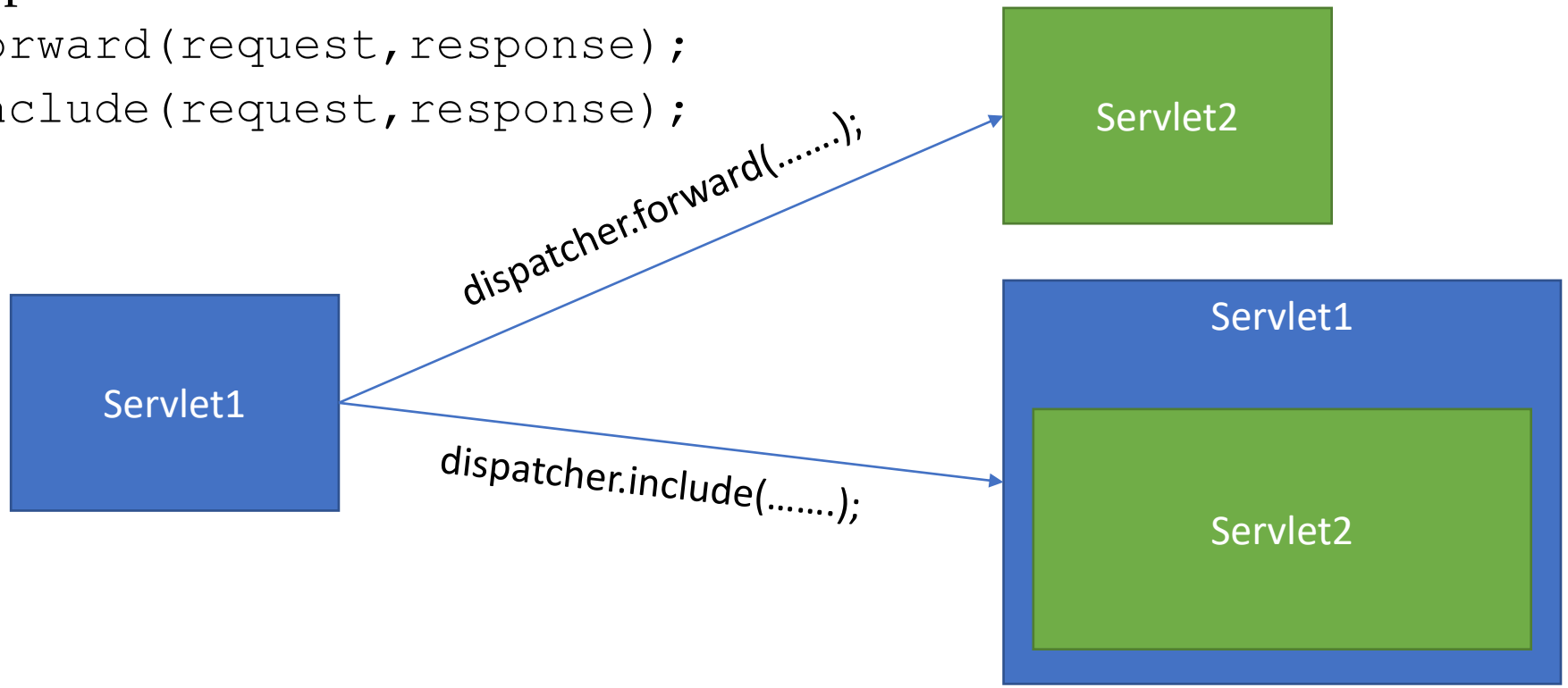
Trao đổi dữ liệu giữa các servlet

- Chuyển hướng từ Servlet đến một trang nội dung mới.
  - Sử dụng sendDirect(URL) của lớp HttpServletResponse.
  - Trang nội dung mới được thay thế hoàn toàn trang nội dung cũ,
  - Thông tin từ servlet1 được gởi đến servlet2 thông qua URL/URI



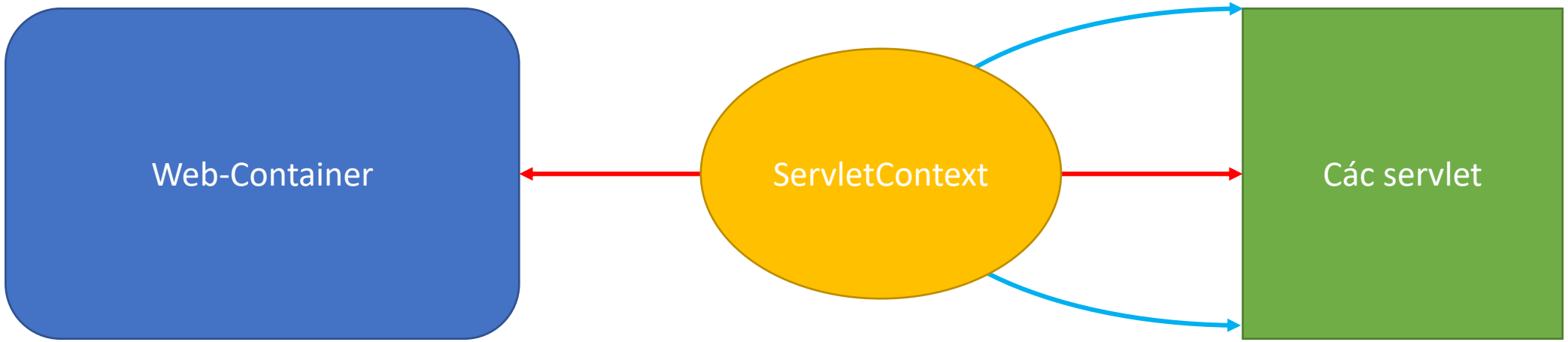
Trao đổi dữ liệu giữa các servlet

- Sử dụng RequestDispatcher .
  - Nhận đối tượng RequestDispatcher từ HttpServletRequest :
    - `RequestDispatcher dispatcher = request.getRequestDispatcher("URL");`
  - Gởi Request và Response đến Servlet khác :
    - `dispatcher.forward(request,response);`
    - `dispatcher.include(request,response);`



ServletContext

- ServletContext cung cấp phương thức cho phép tương tác giữa Web-container và servlet.
- ServletContext dùng để nhận thông tin cấu hình trong web.xml.
- ServletContext có thể quản lý các thuộc tính trong web.xml.
- ServletContext cung cấp khả năng truyền thông giữ các servlet trong web container.



ServletContext

- ServletContext cung cấp phương thức cho phép tương tác giữa Web-container và servlet.
  - Thêm JSP
    - Context.addJspFile()
  - Tạo mới , thêm Servlet
    - context.createServlet();
    - context.addServlet();
  - Tạo mới, thêm Filter
    - context.addFilter();
    - context.createFilter();
  - Tạo mới, thêm Listener
    - context.createListener();
    - context.addListener();

ServletContext

- ServletContext dùng để nhận thông tin cấu hình trong web.xml.
  - context.setInitParameter();
  - context.getInitParameter();
  - context.getInitParameterNames()
- ServletContext có thể quản lý các thuộc tính trong web.xml.
  - context.getAttributeNames();
  - context.getAttribute()
  - context.setAttribute();
- ServletContext cung cấp khả năng truyền thông giữ các servlet trong web container
  - Servlet gởi object bằng cách tạo mới attribute
  - Servlet nhận object bằng cách lấy một attribute.

Lập trình sự kiện trong servlet

- Là quá trình đáp ứng một thay đổi trạng thái của các đối tượng trong servlet container
  - ServletRequestEvent
  - ServletContextEvent
  - ServletRequestAttributeEvent
  - ServletContextAttributeEvent
  - HttpSessionEvent
  - HttpSessionBindingEvent
- Tương ứng với nhóm sự kiện là các listener
- Các listener cần được đăng ký với Web-container thông qua web.xml.

Lập trình sự kiện trong servlet

- Lắng nghe sự kiện bằng listener
  - J2EE cung cấp các listener dưới dạng các interface
  - Lập trình viên phải lập trình lớp listener cài đặt các interface tương ứng
  - Lập trình các hàm sự kiện

```
public class MyContextListener implements ServletContextListener {
	@Override
	public void contextDestroyed(ServletContextEvent arg0) {
		System.out.println("ServletContext da bi huy");
	}

	 //Run this before web application is started
	@Override
	public void contextInitialized(ServletContextEvent arg0) {
		System.out.println("ServletContext da duoc khoi dong");
	}
}
```

Lập trình sự kiện trong servlet

- Đăng ký với Web-container thông qua web.xml.

```
<listener>
	<listener-class>
		MyContextListener
	</listener-class>
</listener>
```

Lập trình Filter

- Filter là một đối tượng phần mềm sẽ được thực thi trước và sau khi xử lý yêu cầu từ Web-client.
- Filter dung để :
  - Ghi nhận tất cả các yêu cầu được gởi đến
  - Lưu giữ địa chỉ IP của nơi gởi yêu cầu
  - Hoán đổi, chuyển hướng
  - Nén và giải nén , mã hóa và giải mã dữ liệu nhận
  - Kiểm tra xác thực dữ liệu đầu vào.
- Thiết kế lớp MyFilter thừa kế từ lớp Filter
- Đăng ký MyFilter và xếp đặt vị trí filter đối với Servlet trong web.xml

Lập trình Filter

- Thiết kế lớp MyFilter thừa kế từ lớp Filter

```
public class MyFilter implements Filter{
public void init(FilterConfig arg0) throws ServletException {}

public void doFilter(ServletRequest req, ServletResponse resp,FilterChain chain)
        throws IOException, ServletException {
        out.print("Thực hiện trước khi xử lý yêu cầu");
        chain.doFilter(req, resp);
        out.print("Thực hiện sau khi xử lý yêu cầu");
}
public void destroy() {}
}
```

Lập trình Filter

- Đăng ký MyFilter và xếp đặt vị trí filter đối với Servlet trong web.xml

```
<filter>
	<filter-name>
		Filter1
	</filter-name>
	<filter-class>
		MyFilter
	</filter-class>
</filter>
<filter-mapping>
	<filter-name>Filter1
	</filter-name>
	<url-pattern>
		/MyServlet1
	</url-pattern>
</filter-mapping>
```

Lập trình Session

- Quản lý phiên làm việc (Session tracking) là phương pháp quản lý trạng thái cá nhân trong một hệ thống được phát triển bằng ứng dụng WEB.
- Các phương pháp quản lý phiên làm việc :
  - Cookies
  - Hidden Form Field
  - URL Rewriting
  - HttpSession
- Trong nội dung này :
  - Cookies với servlet
  - Session với servlet

Truy cập cookies :

- Lấy cookies :
  - Cookie [] cookies = request.getCookies();
  - Truy cập cookie :
    - cookies.getName();
    - cookies.getValue();
- Thêm cookies để gởi đi
  - Cookie cookie = new Cookie(“name”, “value”);
  - cookie.setMaxAge(3600);
  - response.addCookie(cookie);
- Xóa cookie : làm quá hạn thời gian tồn tại