BÀI 4:
Kỹ thuật quản lý session
Phát triển website với JSP

Mục tiêu bài học

- Các kiểu kỹ thuật quản lý session trong ứng dụng web
- Giới thiệu về JSP
- Các thành phần cú pháp của JSP

Kỹ thuật quản lý session

- Các ứng dụng Web sử dụng giao thức HTTP, là giao thức có đặc điểm là stateless và connectionless.
- Stateless: Server không duy trì và lưu giữ trạng thái của các máy khách sau mỗi lần request.

Kỹ thuật quản lý session

- Các ứng dụng Web không thể nhận diện được liệu các request có được gửi bởi cùng một máy khách, hay bởi các máy khác nhau.
- Để khắc phục nhược điểm này, ta cần cài đặt các kỹ thuật quản lý session.
- Quản lý session (Session management) là quá trình theo dõi và ghi nhận các hoạt động của máy khách qua nhiều lần request khác nhau.
- Các kỹ thuật quản lý session thông dụng:
  - Hidden form fields
  - Cookies
  - The `httpsession` API
  - URL rewritting

Quản lý Sessions sử dụng Hidden Form Fields

- Hidden field:
  - Là một thành phần của form tương tự như một text field trong một HTML form.
  - Không được hiển thị lên trên trang Web.
  - Được tạo bằng cách thiết lập giá trị cho thuộc tính `TYPE` của thẻ `<INPUT>` là hidden, như sau:

```
<INPUT TYPE="hidden" NAME="txtHiddenBox"   VALUE="Value1"/>
```

  - Dùng để lưu trữ session ID của một session.
  - Cho phép lưu trữ dữ liệu dưới dạng cặp name-value.
- Session ID là một chuỗi hỗn hợp bao gồm các chữ và số, được dùng để xác định một máy khách/người dùng là duy nhất.

Quản lý Sessions sử dụng Hidden Form Fields

- Dưới đây là đoạn code HTML cho phép người dung nhập tên, sau đó sẽ submit dữ liệu lên `HiddenServlet` servlet khi người dùng click lên `Login` button:

```
<FORM ACTION = "HiddenServlet" METHOD = “POST”>
Username: <INPUT TYPE = TEXT NAME = "user”align=CENTER><BR>
<INPUT TYPE = SUBMIT VALUE = "Login" align=CENTER>
FORM>
```

`HiddenServlet` servlet sẽ nhận dữ liệu từ form gửi lên

Quản lý Sessions sử dụng Hidden Form Fields

- Đoạn code dưới đây minh họa việc nhận user name và sinh ra form HTML với một trường ẩn (hidden form field) có tên là user:

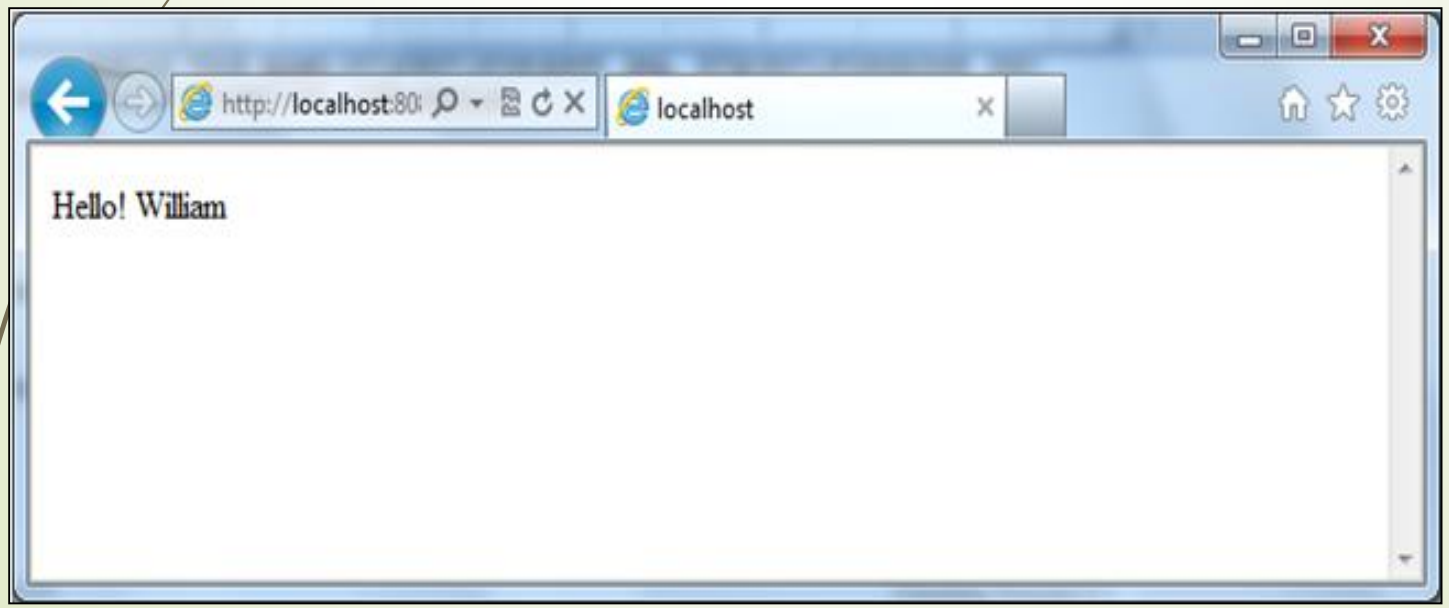
```
 String username = request.getParameter("user");
  out.println("Hello! click Submit to proceed");
  out.println("<Form action='SecondServlet'>");
/* Add a hidden field. */
  out.println("<input type='hidden' name='user` value="
+ username+">");
out.println("<input type='Submit`
value='Submit'></form>");
```

Quản lý Sessions sử dụng Hidden Form Fields

- Trong `SecondServlet` servlet, servlet này sẽ nhận dữ liệu được gửi lên từ form như sau:

```
String uname = request.getParameter("user");
 out.println("Hello! "+uname);
```

- Hình bên dưới minh họa kết quả:



Quản lý Sessions sử dụng Hidden Form Fields

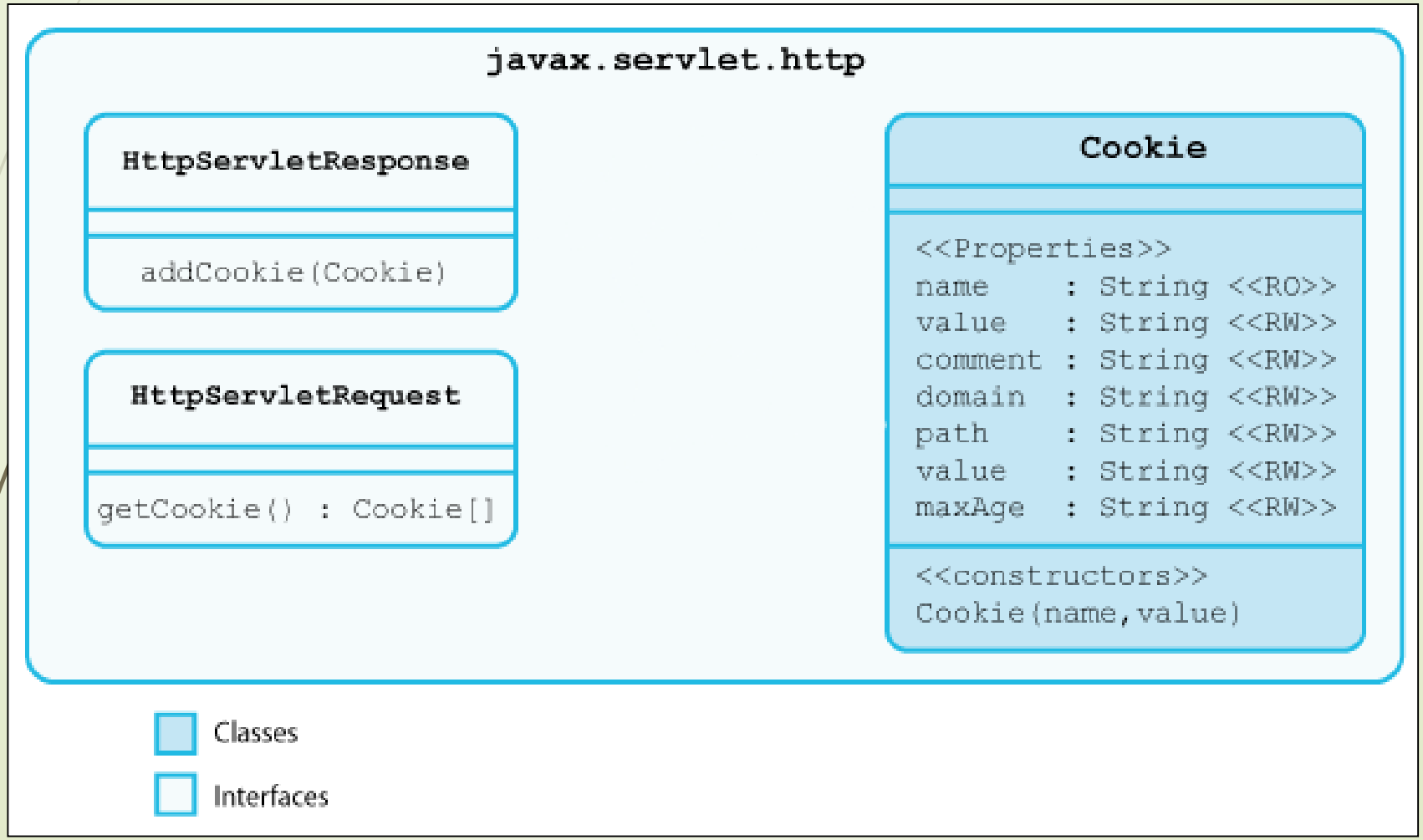
- Kỹ thuật quản lý session sử dụng trường form ẩn có 2 hạn chế sau:
  - Giá trị của trường ẩn sẽ bị mất khi người dùng click lên button back của trình duyệt Web.
  - Dữ liệu của trường ẩn chỉ là text nên sẽ dễ dàng bị đánh cắp.

Quản lý Sessions sử dụng Cookies

- Cookies:
  - Là các file văn bản có kích thước nhỏ được lưu trữ tại máy khách. Cookies do web server tạo ra để định vị và lưu trữ thông tin về máy khách.
  - Cookie lưu trữ giá trị dưới dạng các cặp name-value.
  - Được tạo ra bởi server và được gửi về máy khách cùng với HTTP response headers.
  - Được lưu trữ tại ổ cứng của máy khách và sau đó sẽ được gửi lên server.
  - Được tạo ra bằng cách sử dụng lớp `Cookie` thuộc gói `javax.servlet.http`.

Quản lý Sessions sử dụng Cookies

- Hình bên dưới mình họa lớp `Cookie` nằm trong gói `javax.servlet.http`.



Quản lý Sessions sử dụng Cookies

- Một số phương thức phổ biến của lớp `Cookie`:
  - `String getName()`: Trả về tên cookie
  - `void setValue(String value)`: Thiết lập giá trị cho cookie
  - `String getValue()`: Lấy về giá trị của cookie
  - `setMaxAge(int age)`: Thiết lập thời gian hết hạn của cookie
  - `int getMaxAge()`: Trả về thời gian hết hạn của cookie.

Quản lý Sessions sử dụng Cookies

- Để tạo và gửi cookie về máy khách, cần thực hiện những tác vụ sau:
  - Tạo một đối tượng của lớp Cookie như sau:

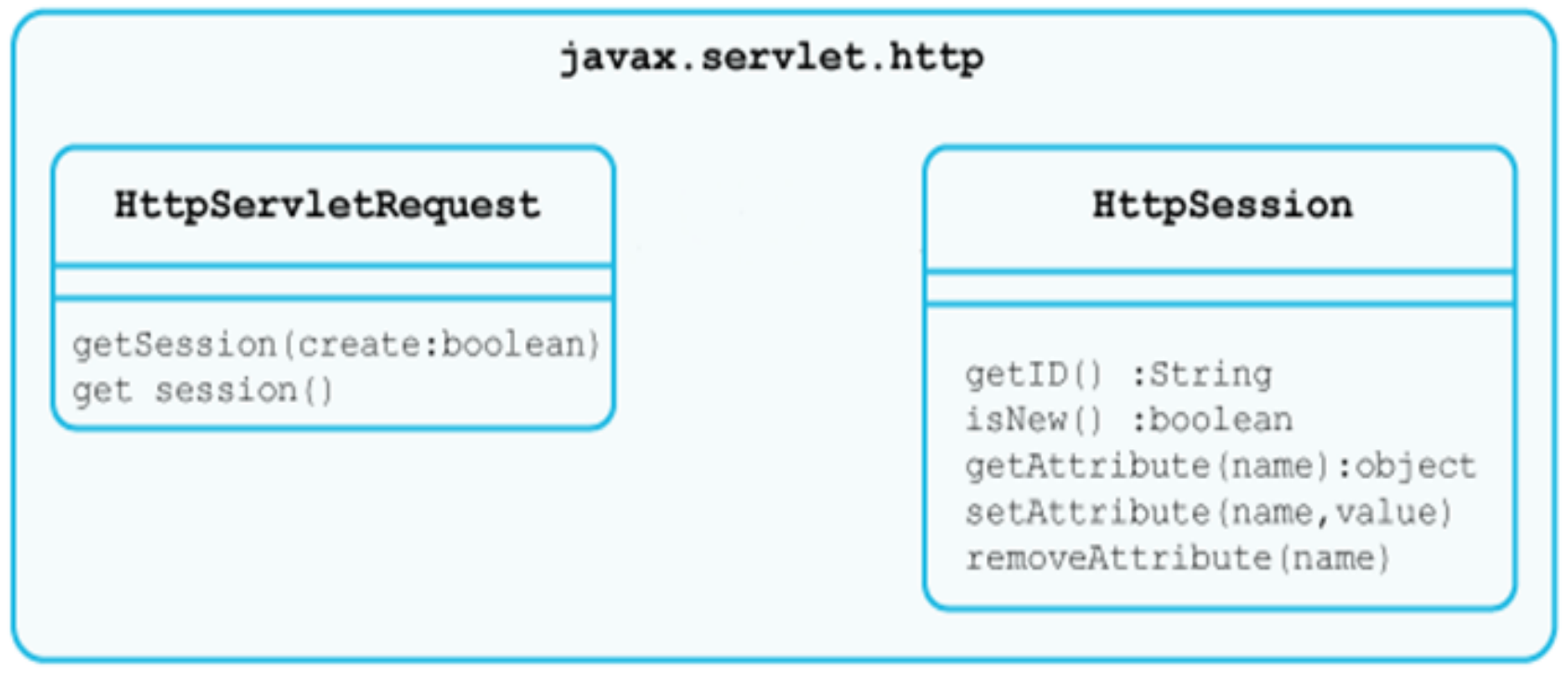
```
Cookie cookieObj = new Cookie("AppCookie",userId);
```

  - Sau đó gửi cookie cho máy khách như sau:

```
response.addCookie(cookieObj);
```

Quản lý Sessions sử dụng HttpSession interface

- `HttpSession` interface:
  - Có thể được dùng để cài đặt cơ chế quản lý session.
  - Thuộc về gói `javax.servlet.http`.
- Hình bên dưới minh họa về interface `HttpSession`.



Quản lý Sessions sử dụng HttpSession interface

- `HttpServletRequest` interface:
  - Cung cấp các phương thức để cho phép quản lý session:
    - `HttpSession getSession(create:boolean)`:
    - `getSession()`
- `HttpSession` interface:
  - Chứa các phương thức được sử dụng để truy cập và thao tác với dữ liệu liên quan đến session:
    - `String getId()`: Lấy về session id
    - `boolean isNew()`: Kiểm tra xem session có phải mới tạo hay không
    - `Object getAttribute(String name)`: Lấy về giá trị của một attribute trong session theo tên
    - `Enumeration<String> getAttributeNames()`: Trả về danh sách các tên thuộc tính trong session
    - `void setAtrribute(String name, Object value)`: Tạo một cặp key-value vào session
    - `void removeAtribute(String name)`: Xóa một attribute trong session theo name
    - `void invalidate()`: Hủy session

Quản lý Sessions sử dụng HttpSession interface

- Đoạn code bên dưới minh họa việc tạo và quản lý session sử dụng interface `HttpSession`:

```
String userId = request.getParameter("txtUserID");
HttpSession session = request.getSession();
PrintWriter out = response.getWriter();
session.setAttribute("sID", userId);
out.println((String)session.getAttribute("sID"));
```

Các thành phần của trang JSP

- JSP là một công nghệ cho phép tạo ra các trang web động một cách dễ dàng hơn rất nhiều so với Servlet.
- JSP là công nghệ ra đời sau Servlet, bản chất của JSP là Servlet. Hai công nghệ JSP-Servlet được sử dụng đồng thời, để bổ trợ cho nhau.
- Một trang Web được phát triển bằng công nghệ JSP sẽ có phần mở rộng là .jsp.
- Công nghệ JSP bao gồm những thành phần cú pháp sau:
  - JSP comments
  - JSP directives
  - JSP declarations
  - JSP scriptlets
  - JSP expressions
  - JSP actions
  - JSP implicit objects

Các thành phần của trang JSP

- JSP comments:
  - Được sử dụng để giải thích ý nghĩa của các đoạn mã JSP trong một trang JSP.
  - Không được thêm vào HTTP response.
  - Có thể được thêm vào một trang JSP bằng cách sử dụng các cú pháp sau:
    - `<%-- comments --%>`
    - `<% /** this is a comment ... **/ %>`
    - `<!-- comments ... -->`
- JSP directives:
  - Cung cấp các thông tin toàn cục mô tả về trang JSP.
  - Có 3 loại sau:
    - page directive
    - taglib directive
    - include directive

Các thành phần của trang JSP

- page directive:
  - Được dùng để định nghĩa các thuộc tính nhằm thông báo cho Web container về các thiết lập chung của trang JSP.
  - Page directive có các thuộc tính sau:
    - `language`: Định nghĩa về ngôn ngữ kịch bản (scripting language) để viết nên trang JSP.
    - `extends`: Khai báo lớp cha mà servlet được sinh ra từ trang JSP sẽ kế thừa.
    - `import`: import các packages, classes, hoặc interfaces trong trang JSP.
    - `session`: Khai báo trang JSP có sử dụng session hay không.
    - `buffer`: Khai báo kích thước của bộ nhớ đệm (buffer).
    - `autoFlush`: Nếu khai báo là true, bộ nhớ đệm sẽ được tự động flushed.
    - `isThreadSafe`: Khai báo liệu trang JSP có phải là thread-safe hay không.

Các thành phần của trang JSP

- `errorPage`: Thiết lập URL của trang JSP dùng để xử lý các ngoại lệ.
- `isErrorPage`: Thiết lập xem liệu trang JSP hiện tại có phải là một trang xử lý lỗi hay không.
- `isELIgnored`: Thiết lập liệu trang JSP hiện tại có bỏ qua các EL expressions hay không, nếu thuộc tính này được khai báo là `true`.
- `info`: Cung cấp thông tin mô tả về trang JSP.
- `pageEncoding`: Thiết lập page encoding được sử dụng bởi trang JSP để gửi response cho trình duyệt.
- `contentType`: Định nghĩa kiểu MIME type cho response.

- The taglib directive:
  - Được dùng để import một thư viện thẻ vào trang hiện tại.
  - Được chèn vào trang JSP bằng đoạn mã sau:

    ```
    <%@ taglib uri="tag_lib_URI" prefix="prefix" %>
    ```

  - Tablig directive có những thuộc tính sau:
    - Uri: Khai báo vị trí của file tld chứa cấu hình của custom tag.
    - Prefix: Định nghĩa một prefix string dùng để truy cập đến custom tag.

Các thành phần của trang JSP

- The include directive:
  - Được dùng để include nội dung của một file khác vào trang hiện tại.
  - Cú pháp như sau:

```
<%@ include file = "URLname" %>
```

- JSP declarations:
  - Cho phép khai báo các biến hoặc các hàm trong một trang JSP.
  - Những thành phần được khai báo ở trong JSP declaration đều sẽ là toàn cục đối với servlet được tạo ra từ trang JSP.
  - Cú pháp: `<%! and %>`

Các thành phần của trang JSP

- Ví dụ sử dụng JSP declarations để khai báo biến và methods:

```
<%!
int i=5;
int add()
{
i=i+5;
return i;
}
%>
```

Các thành phần của trang JSP

- JSP expressions:
  - Dùng để chèn trực tiếp các giá trị vào response output.
  - Được đánh giá khi người dùng đưa ra một HTTP request.
  - Cú pháp:

```
<%= expression%>
```

- Đoạn code sau minh họa việc sử dụng JSP expressions để đánh giá giá trị của một biểu thức:

```
<h1>The product of 5 and 2 is: <%= (2 * 5) %></h1>
```

Các thành phần của trang JSP

- JSP scriptlets:
  - Là thành phần linh hoạt nhất của JSP, cho phép chèn các đoạn mã Java vào trang web.
  - Cú pháp: <% Java code %>
  - Scriptet được thực thi vào request time.
  - Ví dụ:
  - <%
    int x = 5;
    out.println("x = " + x);
    %>

Các thành phần của trang JSP

- Đoạn code minh hoạt việc sử dụng JSP scriptlets để chèn mã Java vào trang JSP:

```
<% int i=10;
        if(i>0)
            {
            out.println("i is a positive
   number");
            }
        else
            {
            out.println("i is a negative
   number");
            }
%>
```

Các thành phần của trang JSP

- JSP actions:
  - Là các thẻ có sẵn của JSP, dùng để thực thi các tác vụ phổ biến trong các trang JSP, VD như include nội dung của các file khác, hoặc chuyển tiếp trang.
  - Cú pháp:

    ```
    <jsp:actionname attribute="">
    ```
- Danh sách các loại JSP action tags:
  - `<jsp:useBean>`
  - `<jsp:getProperty>`
  - `<jsp:setProperty>`
  - `<jsp:forward>`
  - `<jsp:include>`
  - `<jsp:param>`
  - `<jsp:plugin>`

Các thành phần của trang JSP

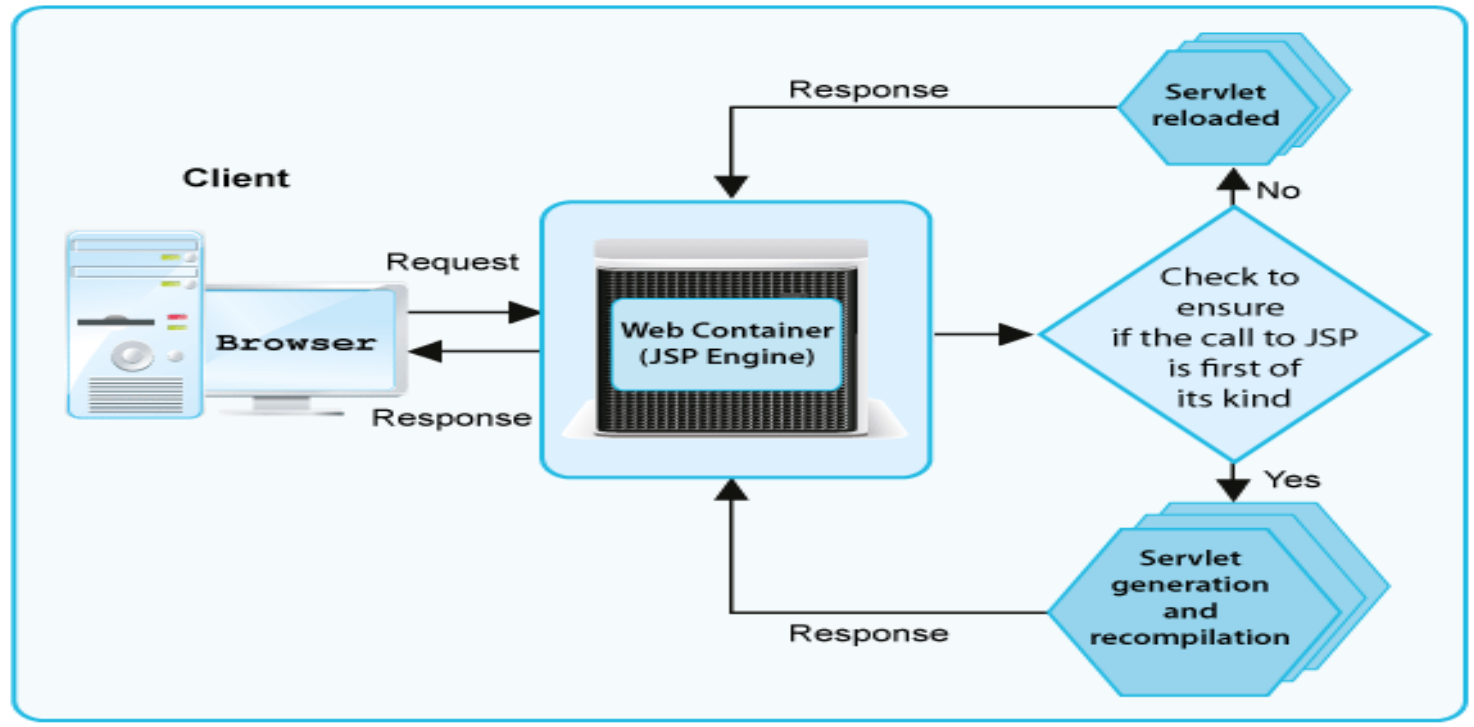
- JSP implicit objects:
  - Là các đối tượng ngầm định đã được định nghĩa sẵn, được cung cấp bởi container và có thể được chèn vào JSP expressions và scriptlets.
  - Được mapped với các classes và interfaces của Servlet API.
- Ý nghĩa của các JSP implicit objects:
  - `request`: Thể hiện đối tượng `HttpServletRequest` được kết hợp cùng với request.
  - `response`: Thể hiện đối tượng `HttpServletResponse` được kết hợp cùng với response được gửi lại cho trình duyệt.
  - `out`: Thể hiện đối tượng `JspWriter` được kết hợp với response.
  - `session`: Thể hiện đối tượng `HttpSession` dùng để cho phép lưu trữ và thao tác với session của người dùng.
  - `application`: Thể hiện đối tượng ServletContext được sử dụng cho ứng dụng Web.

Các thành phần của trang JSP

- `config`: Thể hiện đối tượng `ServletConfig` được kết hợp với servlet được sinh ra từ trang JSP.
- `page`: Thể hiện cho instance hiện tại của trang JSP hiện tại.
- `pageContext`: Thể hiện page context của trang JSP.
- `exception`: Thể hiện ngoại lệ `Throwable` trong một trang JSP.

Vòng đời của trang JSP

- Bất cứ khi nào trình duyệt gửi yêu cầu một trang JSP nào đó lên server, căn cứ vào URL được gửi, server sẽ gửi request đó cho JSP engine.
- JSP engine là một phần của Web container làm nhiệm vụ biên dịch trang JSP trở thành servlet.
- Hình bên dưới minh họa các sự kiện xảy ra sau khi máy khách gửi yêu cầu một trang JSP.



Vòng đời của trang JSP

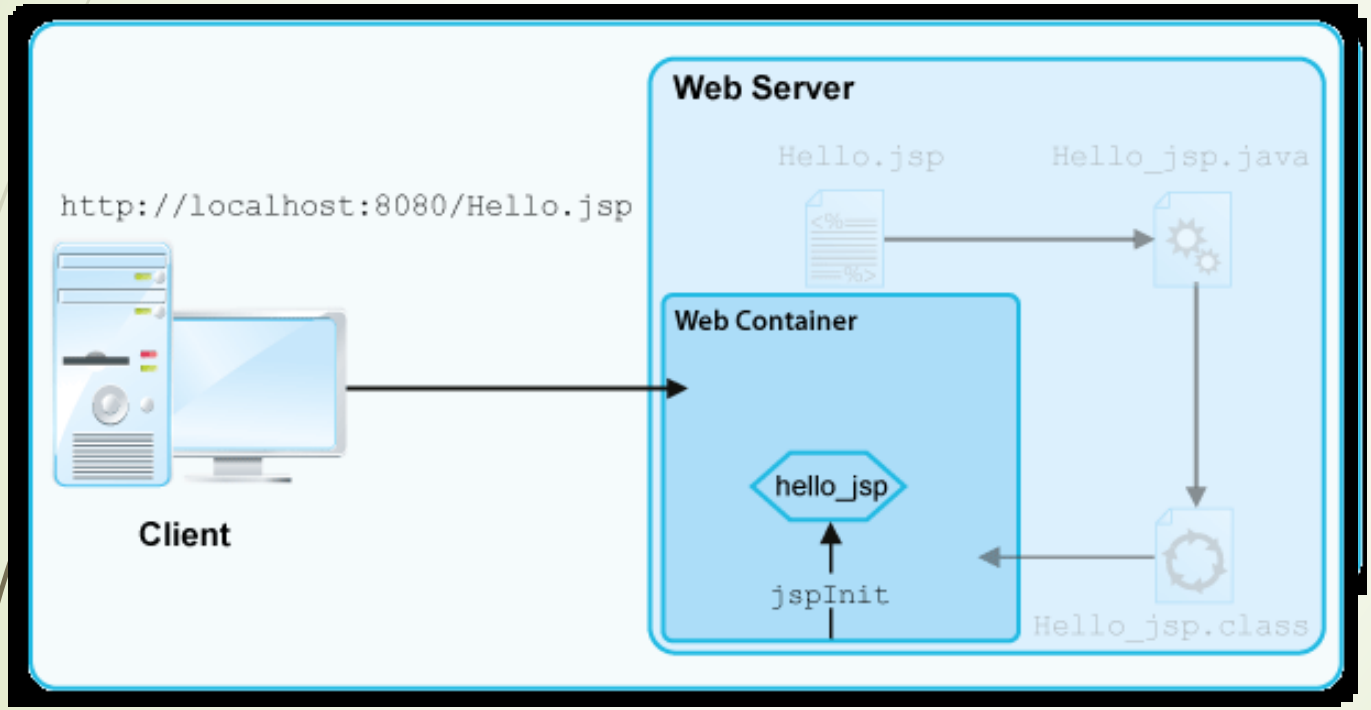
- Vòng đời của trang JSP được quản lý bằng các phương thức sau thuộc về `javax.servlet.jsp.JspPage` interface:
  - `jspInit()`
  - `_jspService()`
  - `jspDestroy()`

Quá trình xử lý của trang JSP

- Trang JSP cần được chuyển thành servlet trước khi nó có thể phục vụ các request của client.
- Quá trình chuyển đổi (conversion) của một trang JSP thành servlet sẽ được thực thi trong những giai đoạn sau:
  - Translation
  - Compilation
  - Servlet class loading
  - Servlet instance creation
  - Servlet initialization
  - Servicing client requests
  - Servlet destruction

Quá trình xử lý của trang JSP

- Hình bên dưới minh họa các giai đoạn diễn ra trong quá trình chuyển đổi trang JSP thành servlet.



Tổng kết

- Các kiểu kỹ thuật quản lý session trong ứng dụng web
- Giới thiệu về JSP
- Các thành phần cú pháp của JSP