**BẢNG ĐÁNH GIÁ ĐIỂM THUYẾT TRÌNH**

**MÔN LẬP TRÌNH JAVA NÂNG CAO**

**LỚP CTK44-PM**

|  |  |
| --- | --- |
| **Thông tin đề tài thuyết trình** | **Thông tin sinh viên thực hiện đánh giá** |
| Tên đề tài: Tìm hiểu về Scope, Listener và Filter trong Java Web | MSSV của sinh viên đánh giá: 2012353 |
| Điểm tổng kết: Ngày trình bày: 14/04/2023 | Họ tên sinh viên & ký xác nhận: Đoàn Cao Nhật Hạ |
| ***Thông tin sinh viên thuyết trình*** |
| MSSV: 2011400  Họ tên: Nguyễn Tuấn Kiệt |

1. **ĐÁNH GIÁ THUYẾT TRÌNH**

**Thang điểm: Kém 0-5; Trung bình: 5-7; Khá 7-8; Tốt 8-9; Xuất sắc: 9-10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng chấm điểm nhóm thuyết trình** | | | | | | |
| Các chủ đề chính có được xác định? (A) | Slide thuyết trình có thiết kế tốt (B) | Giao tiếp với người nghe bằng ánh mắt (C) | Giao tiếp với người nghe bằng giọng nói (D) | Khả năng trình bày nội dung (E) | Thảo luận & Hỏi đáp (F) | Điểm trung bình = (A+B+C+D+E+F)/6 |
| **9** | **9** | **8,5** | **9** | **9** | **8,5** | **8,9** |

1. **BÀI TẬP NẮM KIẾN THỨC**

Dựa vào nội dung vừa trình bày, trả lời các câu hỏi sau:

1. Trình bày tóm tắt nội dung anh/ chị đã nắm được về chủ đề vừa trình bày?

* Scope: là định nghĩa về thời gian mà dữ liệu của bạn có tồn tại cho người dùng trong ứng dụng web.
* Application scope hay context scope :

+ bắt đầu từ khi ứng dụng web được chạy cho tới khi shutdown. Các tham số, giá trị (Parameters/attributes) bên trong application scope sẵn dùng với tất cả các request và session.

+ được định nghĩa bởi javax.servlet.ServletContext interface.

+ Trong một servlet, bạn có theer lấy các đối tượng application bằng cách gọi getServletContext() bên trong servlets code một cách trực tiếp hoặc gián tiếp quagetServletConfig().getServletContext().

+ Web container cung cấp một đối tượngServletContext cho mỗi ứng dụng web.

* Session scope:

+ Bắt đầu khi một client thành lập kết nối với ứng dụng web cho tới khi hết thời gian timeout hoặc browser bị đóng.

+ Các giá trị bên trong session scope có thể được truy cập từ nhiều request khác nhau của cùng 1 client

+ Một tính năng đáng chú ý của trình duyệt web (browser) là  session được chia sẻ giữa các tab khác nhau. Bạn có thể gửi request từ 1 tab khác mà không cần login lại. Ví dụ bạn login gmail ở 1 tab, sau đó bạn sang tab khác vẫn có thể tải mail, gửi mail mà không cần phải login lại. Do đó, hãy luôn mở một cửa sổ trình duyệt mới khi bạn muốn thực hiện các giao dịch bảo mật.

+ Session object sẵn dùng trong JSP pages giống như một object ẩn

+ Trong Servlet, bạn có thể lấy đối tượng object bằng cách gọi request.getSession().

* Request scope:

+ Bắt đầu ngay khi một HTTP request được gửi tới server và kết thúc khi server trả về một HTTP response

+ Các tham số/thuộc tính trong một Request scope có thể được truy cập từ các servlet hoặc jsp cùng phục vụ trong 1 request. Ví dụ bạn gọi 1 servlet/jsp sau đó các servlet/jsp này lại gọi các servlet/jsp khác rồi mới trả về reponse.

+ Request object có sẵn trong JSP page như là 1 object ẩn . Bạn có thể set value cho 1 thuộc tính trong request object từ servlet và lấy nó ra ở JSP (Phải trong cùng 1 request).

* JSP page scope:

+ Giới hạn bởi phạm vi và thời gian tồn tại của các thuộc tính trong cùng 1 page nơi mà nó được tạo.

+ Nó sẵn dùng trong một JSP page giống như một object ẩn.

* Listener:
* Là một phương thức hoặc hàm servlet đang chờ sự xuất hiện của một sự kiện.
* Là các lớp giám sát một loại sự kiện cụ thể và kích hoạt chức năng khi sự kiện đó xảy ra.
* Hoạt động để nghe các sự kiện trong các vùng chứa web như phiên được tạo, thuộc tính được chèn vào phiên hoặc vùng chứa được kích hoạt.
* Được tạo ra chủ yếu để lắng nghe các sự kiện (event) và phản hồi.
* Bộ chứa servlet tạo ra các sự kiện kích hoạt hành động của các lớp người nghe sự kiện.
* Listener trong Servlet: Context, Request, Attribute, Session,...
* Servlet API cung cấp các Listener interfaces sau:

+ javax.servlet.AsyncListener

+ javax.servlet.ServletContextListener

+ javax.servlet.ServletContextAttributeListener

+ javax.servlet.ServletRequestListener

+ javax.servlet.ServletRequestAttributeListener

+ javax.servlet.http.HttpSessionListener

+ javax.servlet.http.HttpSessionBindingListener

+ javax.servlet.http.HttpSessionAttributeListener

+ javax.servlet.http.HttpSessionActivationListener

* Cấu hình Listener (trong file web.xml)

<listener>

<listener-class>

listeners.AppContextListener

</listener-class>

</listener>

* Filter:
* Là một đối tượng được gọi trong quá trình tiền xử lý và hậu xử lý của một yêu cầu
* Nó dễ kết hợp vào hệ thống, được định nghĩa trong file web.xml. Khi cần có thể thêm hoặc xóa filter mà không ảnh hưởng đến servlet.
* Sử dụng Filter khi:

+ Ghi lại tất cả các request đến.

+ Ghi nhật ký địa chỉ IP của máy tính từ các resquests nguồn.

+ Chuyển đổi.

+ Nén dữ liệu.

+ Mã hóa và giải mã.

+ Xác thực đầu vào, v.v.

* Cấu hình Filter (trong file ưeb.xml)

<filter>

<filter-name>RequestLoggingFilter</filter-name> <!-- mandatory -->

<filter-class>com.journaldev.servlet.filters.RequestLoggingFilter</filter-class> <!-- mandatory -->

<init-param> <!-- optional -->

<param-name>test</param-name>

<param-value>testValue</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>RequestLoggingFilter</filter-name> <!-- mandatory -->

<url-pattern>/\*</url-pattern> <!-- either url-pattern or servlet-name is mandatory -->

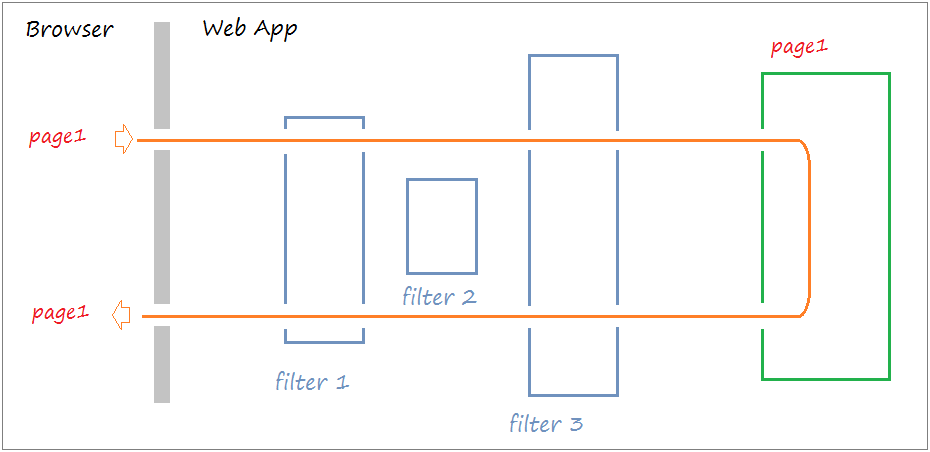
<servlet-name>LoginServlet</servlet-name>

<dispatcher>REQUEST</dispatcher>

</filter-mapping>

* Một số trường hợp sử dụng Filter

+ Trường hợp 1:



+ Trường hợp 2:

Chart

Description automatically generated

+ Trường hợp 3:

A picture containing chart

Description automatically generated

* Các phương thức Servlet Filter

+ public void doFilter (ServletRequest, ServletResponse, FilterChain): Phương thức này được gọi bởi container mỗi khi một cặp request/response được truyền thông qua chain.

+ public void init(FilterConfig filterConfig) : Phương thức này được triệu hồi bởi Web container để chỉ một Filter mà đang được đặt bên trong service

+ public void destroy(): Phương thức được gọi bởi Web container để chỉ một Filter bị hủy khỏi service

* **Authentication Filter/ FilterConfig:**

**Có thể thực hiện xác thực bằng Filter.**

**→ Thông qua ví dụ về xác thực khi đăng nhập**

1. Anh/ chị hãy đưa ra các nhận xét và góp ý cho sinh viên vừa trình bày?

* Bổ sung thêm phần Listener:

+ Ưu điểm:

+ Nhược điểm:

* Servlet API cung cấp các Listener interfaces:

+ javax.servlet.AsyncListener: đối tượng listener này được thông báo nếu có sự thay đổi trạng thái của một hoạt động không đồng bộ được khởi xướng bởi một ServletRequestobject. Các sự kiện không đồng bộ được kích hoạt do một ServletRequest là loại

+ javax.servlet.ServletContextListener: đối tượng listener này được thông báo khi một sự kiện vòng đời như khởi tạo của một servlet xảy ra. Các đối tượng sự kiện tương ứng là javax.servlet.ServletContextEvent.

+ javax.servlet.ServletContextAttributeListener: đối tượng người nghe này được thông báo khi có sự thay đổi trong các thuộc tính của các sự kiện, vòng đời của servlet. Các sự kiện tương ứng là javax.servlet.ContextAttributeEvent.

+ javax.servlet.ServletRequestListener: đối tượng người nghe này được thông báo nếu nhận được yêu cầu servlet hoặc gửi ra một yêu cầu từ các ứng dụng Web. Các sự kiện tương ứng là  javax.servlet.ServletRequestEvent.

+ javax.servlet.ServletRequestAttributeListener: đối tượng người nghe này được thông báo nếu có sự thay đổi trong các thuộc tính của các yêu cầu servlet. Các sự kiện tương ứng là javax.servlet.RequestAttributeEvent.

+ javax.servlet.http.HttpRequestListener: đối tượng người nghe này được thông báo khi có yêu cầu HTTP được nhận từ các thành phần khác hoặc khi yêu cầu HTTP được gửi đến các thành phần khác của ứng dụng.

+ javax.servlet.http.HttpSessionBindingListener: đối tượng người nghe này được thông báo khi một đối tượng nhất định bị ràng buộc hoặc khôngbịràngbuộc bởi một session  HTTP. Các sự kiện tương ứng là javax.servlet.http.HttpSessionBindingEvent.

+ javax.servlet.http.HttpSessionAttributeListener: đối tượng người nghe này được thông báo khi có sự thay đổi trong các thuộc tính của một session HTTP.

+ javax.servlet.http.HttpSessionActivationListener: session HTTP có thể được thụ động thay vì loại bỏ chúng khỏi bộ nhớ và một lần nữa kích hoạt khi có một yêu cầu cho các servlet. Khi kích hoạt phiên HTTP xảy ra, đối tượng người nghe này được thông báo.