

Ejercicios de Filtros y Wrappers

Índice

1 Filtro de acceso restringido.....	2
2 Restringir el acceso al chat.....	2
3 Wrapper de ejemplo (*).....	2
4 Registro de accesos (*).....	3

1. Filtro de acceso restringido

Vamos a probar la aplicación `jsp-sesion11-filtro` en la que tenemos un filtro `RestringirAcceso` que prohíbe el acceso de los usuarios no registrados al contenido del directorio `restringido`. Se pide:

a) Desplegar la aplicación en Tomcat. Una vez instalada comprobamos que la aplicación se ha instalado correctamente. Tras esto intentamos acceder a uno de los recursos del directorio `restringido`:

```
http://localhost:8080/jsp-sesion11-filtro/restringido/index.html  
http://localhost:8080/jsp-sesion11-filtro/restringido/index.jsp
```

¿Qué ocurre? ¿Por qué?

b) Validarse como usuario y volver a intentar acceder al directorio `restringido`. ¿Ahora que ocurre? ¿Qué ventajas tiene implementar esta restricción de acceso mediante un filtro?

2. Restringir el acceso al chat

Recordemos que en sesiones anteriores realizamos un chat mediante servlets, en el que había que registrarse como usuario antes de entrar a hablar. Sin embargo, si se introduce directamente la dirección:

```
http://localhost:8080/jsp-sesion05-chat/chat/chatFrames.html
```

Podemos acceder directamente al chat sin estar registrados. Se pide:

a) Implementar un filtro en el chat que restrinja el acceso al chat si no se ha registrado un `nick` en la sesión.

b) ¿A qué recursos deberá afectar este filtro? Introducir en el descriptor de despliegue (`web.xml`) la configuración necesaria para que el filtro intercepte los intentos de acceso a los recursos restringidos.

3. Wrapper de ejemplo (*)

La aplicación `jsp-sesion11-wrapper` incorpora un filtro que utiliza un *wrapper* para analizar la respuesta generada. Toma esta respuesta del *wrapper* (asumimos que es contenido HTML), y la analiza utilizando la librería *htmlparser*, para extraer su título. Una vez tiene el título registra que se ha accedido a dicha página en el *log* y devuelve la respuesta al cliente. Se pide:

a) Desplegar la aplicación en Tomcat. Probar a acceder a varios ficheros HTML estáticos de los incluidos dentro de la aplicación. Comprobar que en el *log* se ha registrado el acceso indicando el título de las páginas.

b) Si tuviesemos recursos que no fuesen HTML (como por ejemplo imágenes) dentro de la ruta a la que afecta el filtro, ¿que ocurriría?. Intentar acceder a una imagen dentro de la aplicación y ver el error que se produce. ¿A qué se debe esto? ¿Como podríamos solucionarlo?

4. Registro de accesos (*)

Vamos a realizar una aplicación que contabilice el número de accesos a las páginas mediante un filtro. Esta aplicación tiene el nombre `jsp-sesion11-ranking`. Para ello tendremos una base de datos, con una tabla `PAGINAS` en la que figurará:

<code>ruta varchar(255)</code>	Ruta de la página visitada
<code>titulo varchar(100)</code>	Título de la página
<code>accesos integer</code>	Número de accesos realizados

El servlet `RankingPaginasServlet` nos genera un listado de las páginas ordenadas por número de visitas.

Lo primero que deberemos hacer será instalar la base de datos. El script para la creación de esta base de datos se encuentra en el directorio `db`.

Se pide:

a) Desarrollar el filtro `AccesoPaginaFilter` que actúe sobre todos los recursos estáticos, y que contabilice el número de visitas que se realiza a ellos almacenando esta información en la base de datos. Deberá cumplir las siguientes características:

- Cuando una página sea visitada por primera vez, se deberá registrar una nueva entrada en la BD, con la URL de la página visitada y el contador de visitas inicializado a 1. Por el momento no obtendremos el título de la página, ya que para esto se requiere utilizar un *wrapper*, así que en este campo introduciremos siempre el valor (*Título desconocido*).
- Cuando la página ya estuviese registrada en la BD, lo que haremos será incrementar el número de visitas en 1.

b) Actualizar el filtro `AccesoPaginaFilter` para que obtenga el título de la página a la que se ha accedido, utilizando para ello el *wrapper* genérico que se proporciona. Deberá cumplir las siguientes características:

- Sólo registrará los recursos cuyo contenido sea html (tipo de contenido `text/html`).
- Cuando una página sea visitada por primera vez, se deberá extraer su título de la etiqueta `<title>`, para lo cual deberá usarse un *wrapper*. En este caso insertaremos los datos de la página en la BD anotando una visita.
- Cuando la página ya estuviese registrada en la BD, lo que haremos será incrementar el número de visitas en 1.

c) Si la página HTML está almacenada en la caché del navegador, el acceso no se contabilizará correctamente. ¿Qué cabecera HTTP podríamos utilizar para solucionar este problema? ¿Donde podríamos establecer esta cabecera? Establecer las cabeceras necesarias para evitar el uso de la caché y comprobar el correcto funcionamiento de la aplicación.

Nota

Para asegurarnos de que funciona correctamente con los diferentes navegadores, podemos incluir una serie de cabeceras destinadas a evitar que la página se almacene en la caché:

```
Cache-control: no-cache  
Cache-control: no-store  
Pragma: no-cache  
Expires: 0
```

