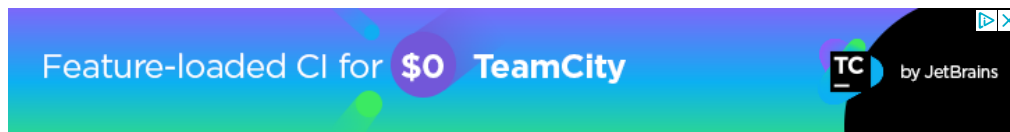


Blog sobre Java EE

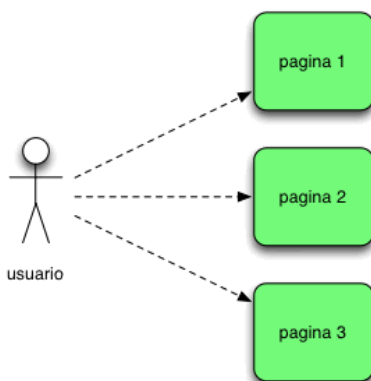
Estás aquí: [Inicio](#)/[Java EE](#)/[Java Web](#)/Servlet 3.0 y Java Filter

Servlet 3.0 y Java Filter

5 marzo, 2014 por [Cecilio Álvarez Caules](#) — 6 comentarios

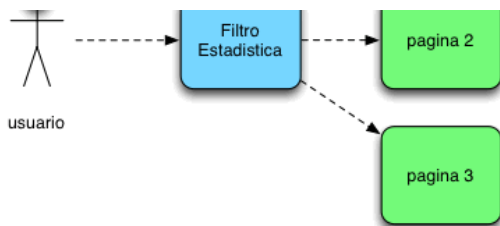


Cuando empezamos a trabajar un Java EE se nos presentan muchos conceptos nuevos ,a veces demasiados pero uno de los más clásicos es el concepto de java filter o filtro java .Practicamente todos los frameworks de capa de presentación los usan de una forma o de otra para aportar nuevas funcionalidad extendiendo de las ya disponibles . Ahora bien ¿como funcionan? y sobre todo para que sirven. Vamos a partir de la idea que tenemos una aplicación web con tres páginas.



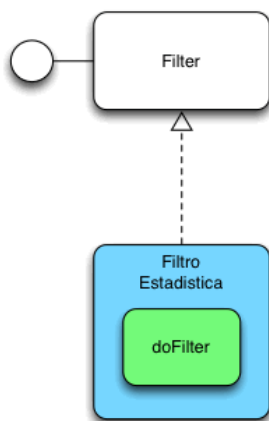
Java Filter

Un filtro de java se encarga de añadir una nueva funcionalidad a la aplicación colocandose entre el usuario y las páginas .Por ejemplo en nuestro caso vamos a implementar un sencillo filtro que construye unas estadísticas sobre cuantas veces se ha visitado cada página



Java Filter (Construcción)

Definir un filtro en Java es bastante sencillo sobre todo a partir de Servlet 3.0 que nos podemos apoyar en el sistema de anotaciones. Un Filtro implementa la interface Filter de JEE que aporte el método doFilter.



Vamos a ver el código de este filtro :

```

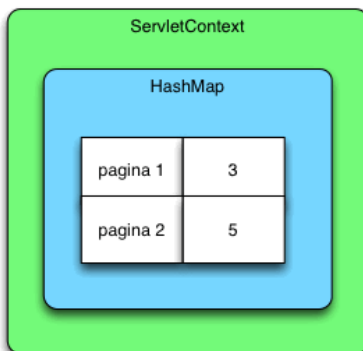
1  </pre>
2  package com.arquitecturajava;
3
4  import java.io.IOException;
5  import java.util.HashMap;
6
7  import javax.servlet.Filter;
8  import javax.servlet.FilterChain;
9  import javax.servlet.FilterConfig;
10 import javax.servlet.ServletContext;
11 import javax.servlet.ServletException;
12 import javax.servlet.ServletRequest;
13 import javax.servlet.ServletResponse;
14 import javax.servlet.annotation.WebFilter;
15 import javax.servlet.http.HttpServletRequest;
16 @WebFilter(filterName="/FiltroEstadistica",urlPatterns="/*" )
17 public class FiltroEstadistica implements Filter {
18
19     public void destroy() {
20         // TODO Auto-generated method stub
21     }
22     @SuppressWarnings("unchecked")
23     public void doFilter(ServletRequest request, ServletResponse response, Fil
24         // TODO Auto-generated method stub
25         // place your code here
26
  
```

```

37 }else {
38
39 // actualizamos claves e incrementamos
40 urls=(HashMap<String,Integer>)contexto.getAttribute("estadistica");
41
42 if (urls.get(peticion.getRequestURL().toString())==null) {
43
44 urls.put(peticion.getRequestURL().toString(), 1);
45 }else {
46 urls.put(peticion.getRequestURL().toString(), urls.get(peticion.getRequest
47 )
48 }
49
50 chain.doFilter(request, response);
51 }
52
53 /**
54 * @see Filter#init(FilterConfig)
55 */
56 public void init(FilterConfig fConfig) throws ServletException {
57 // TODO Auto-generated method stub
58 }
59
60 }
61 <pre>

```

El filtro es relativamente sencillo .Se encarga de almacenar un HashMap o diccionario en el contexto de de aplicación (ámbito de aplicación). Este mapa contendrá el nombre de la página y las veces que hemos accedido a ella .



De esta forma será luego muy sencillo construir un JSP básico que accede al contexto (ServletContext) y se encargue de imprimir por pantalla la lista de páginas ya acceso que ha guardado . Vamos a ver su código a través del uso de scriptlet de JSP que aunque no es la mejor práctica sirve para clarificar .

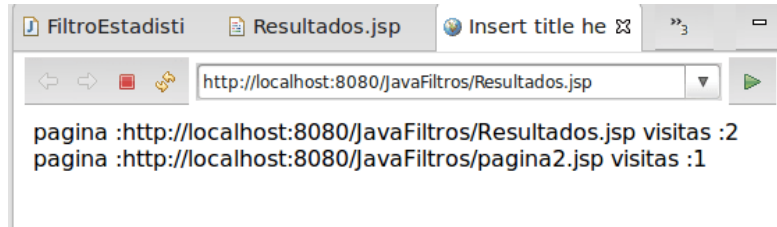
```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@page import="java.util.HashMap" %>
4
5 <%@page import="java.util.Set" %>
6
7 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.
8 <html>
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11 <title>Insert title here</title>
12 </head>
13 <body>
14 <%

```

```
25 | %>  
26 | </body>  
27 | </html>
```

Realizada esta operación el JSP nos mostrará lo siguiente :



El código : [JavaFiltros](#)



PDF



0

COMPART

Archivada en: [Java Web](#)

Etiquetada con: [JavaeeCore](#)

Comentarios



Fred dice

12 enero, 2016 en 19:15

Muchas gracias por el aporte, muy claro y facil de probar.

Saludos.

[Responder](#)



Cecilio Álvarez Caules dice

15 enero, 2016 en 9:46

gracias 😊

[Responder](#)

Tengo una duda de como generar mensajes desde un servlet, ya que estoy leyendo el libro “EJB 3.1 Cookbook” de Richard M. Reese, de la editorial Packt Publishing capítulo 3 Message-Driven Beans.

Para ser exacto a la hora de declarar el objeto Queue

```
@Inject
@JMSConnectionFactory("jms/RemoteConnectionFactory")
private JMSContext context;
@Resource(lookup="TestQ")
private Queue queue;

***(en el libro el objeto QueueConnectionFactory lo declara distinto
@Resource(mappedName="jms/SomeMessageFactory")
private QueueConnectionFactory queueConnectionFactory;
)
```

Esto lo cheque por el libro Beginning Java EE 7 de Antonio gonalves en la parte de producir mensajes dentro del contenedor.

Con esto ya no tenia error en declarar QueueConnectionFactory en el servidor Wildfly, pero en Queue me sigue tirando error

```
*****
Deployment failed: TestServletJMSWildFly.war: {
  "outcome" : "failed",
  "failure-description" : {"JBAS014771: Services with
missing/unavailable dependencies" :
["jboss.naming.context.java.module.TestServletJMSWildFly.TestServletJMSWildFly.env.\"com.servlet.TextServlet\".queue
is missing
[jboss.naming.context.java.module.TestServletJMSWildFly.TestServletJMSWildFly.env.TestQ]"]},
  "rolled-back" : true
}
*****
```

Tambien lo probe en el servidor Jboss EAP 6.4/6.3

Ojala y me puedas solventar esta duda, de como generar mensajes desde un servlet?

Responder

Trackbacks

Extendiendo Java EL - Arquitectura Java dice:

17 diciembre, 2014 a las 11:51

[...] Otros artículos relacionados: [Servlet 3.0](#) . [ServletContext](#) . [ServletFilter](#) [...]

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cookies

[JAVA SE](#)[SPRING](#)[JAVA EE](#)[JAVASCRIPT](#)[FRAMEWORKS JS](#)[ARQUITECTURA](#)[MIS LIBROS](#)[MIS CURSOS](#)

10 octubre, 2017 a las 10:10

[...] artículos relacionados : ServletContext, Java Filter , [...]

[Responder](#)

Spring Security (I) configuracion - Arquitectura Java dice:

20 marzo, 2014 a las 12:40

[...] param) . El siguiente paso es declarar un listener que nos inicialice el framework y por último un filtro que protega toda la aplicación de accesos no permitidos y delegue en Spring Framework todas las [...]

[Responder](#)

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Correo electrónico *

Web

PUBLICAR COMENTARIO

Este sitio usa Akismet para reducir el spam. [Aprende cómo se procesan los datos de tus comentarios.](#)

Mis Cursos de Java Gratuitos

Java Herencia



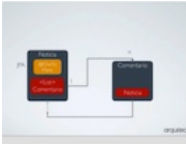
Java JDBC



Servlets

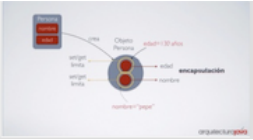


Introduccion JPA

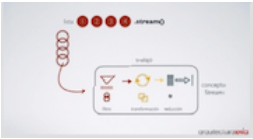


Mis Cursos de Java

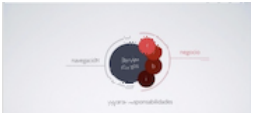
Programación Orientada a Objeto en Java



Java APIS Core



Java Web



arquitectura java con spring



POPULAR

- El patrón de inyección de dependencia y su utilidad
- Enterprise Java Beans y su funcionamiento
- Framework vs Libreria dos conceptos importantes
- Los Frameworks y su lado oscuro
- DRY vs DAMP dos principios importantes
- ¿Qué es un Java Maven Artifact ?
- Utilizando un JPA Stream con JPA 2.2
- JPA @Basic , optimizando los fechings
- JSON Viewer Awesome y chrome
- Utilizando un JSON Generator en Java

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

- ¿Qué es Spring Boot?
- Static Method vs instance method y su uso correcto
- Java Constructores this() y super()
- Usando Java Session en aplicaciones web
- Java Iterator vs ForEach
- Ejemplo de Java Singleton (Patrones y ClassLoaders)
- Introducción a Servicios REST
- Los Frameworks y su lado oscuro
- Java Override y encapsulación
- Usando el patron factory
- ¿Cuales son las certificaciones Java?
- Comparando java == vs equals

[Servlet 3.0 y Java Filter](#)

[¿Qué es un Microservicio?](#)

[Java new String y la creación de objetos](#)

[Mis Libros](#)

[Spring MVC Configuración \(I\)](#)

[El patrón de inyección de dependencia y su utilidad](#)

[¿ Que es REST ?](#)

[¿Qué es un Java Maven Artifact ?](#)

[Entiendo los métodos de Java equals y hashCode](#)