

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

目录视图

摘要视图

订阅

从创业到再就业，浅述对程序员职业生涯的看法

征文 | 你会为 AI 转型么？

赠书：7月大咖新书机器学习/Android/python

Servlet3.0学习总结(二)——使用注解标注过滤器(Filter)

标签： `servlet3.0` `filter`

2016-04-22 21:36

223人阅读

评论(0)

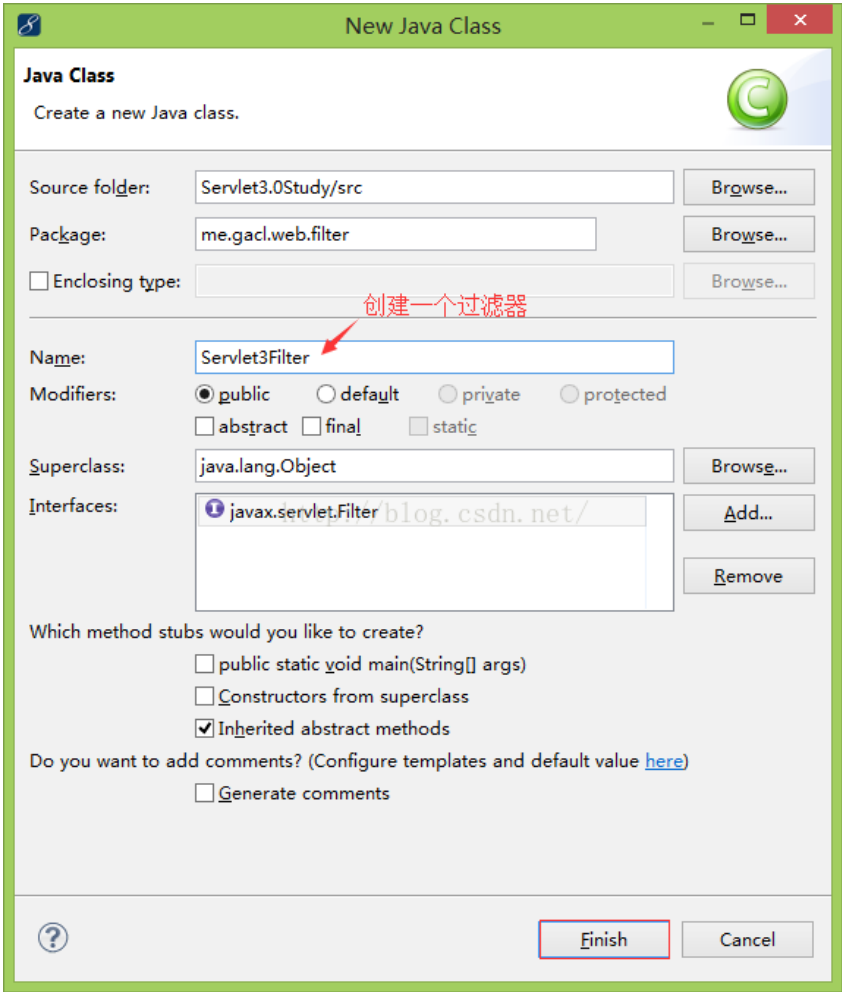
收藏

举报

分类： `web后台 (12)`

Servlet3.0提供@WebFilter注解将一个实现了javax.servlet.Filter接口的类定义为过滤器，这样我们在web应用中使用过滤器时，也不再需要在web.xml文件中配置过滤器的相关描述信息了。

下面我们来创建一个过滤器，体验一下使用@WebFilter注解标注过滤器，如下所示：



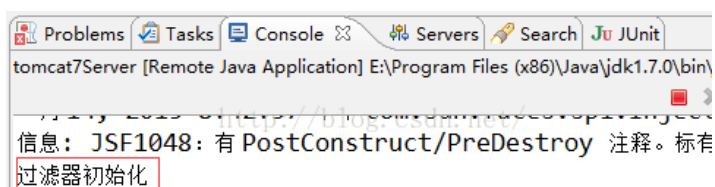
过滤器的代码如下：

```
[java]
01. <pre name="code" class="java"><span style="font-size:14px;">
02. </span>

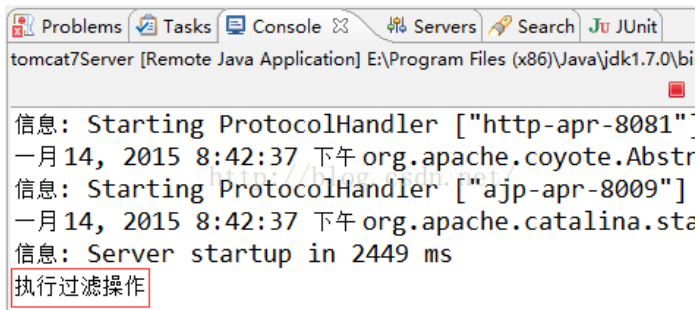
[java]
01. <span style="font-size:14px;">package me.gacl.web.filter;
02. import java.io.IOException;
03. import javax.servlet.Filter;
04. import javax.servlet.FilterChain;
05. import javax.servlet.FilterConfig;
06. import javax.servlet.ServletException;
07. import javax.servlet.ServletRequest;
08. import javax.servlet.ServletResponse;
09. import javax.servlet.annotation.WebFilter;
10.
11. /**
12.  * 使用注解标注过滤器
13.  * @WebFilter将一个实现了javax.servlet.Filter接口的类定义为过滤器
14.  * 属性filterName声明过滤器的名称,可选
15.  * 属性urlPatterns指定要过滤的URL模式,也可使用属性value来声明。(指定要过滤的URL模式是必选属性)
16.  */
17. @WebFilter(filterName="Servlet3Filter",urlPatterns="/*")
18. public class Servlet3Filter implements Filter {
19.
20.     @Override
21.     public void destroy() {
22.         System.out.println("过滤器销毁");
23.     }
24.
25.     @Override
26.     public void doFilter(ServletRequest request, ServletResponse response,
27.         FilterChain chain) throws IOException, ServletException {
28.         System.out.println("执行过滤操作");
29.         chain.doFilter(request, response);
30.     }
31.
32.     @Override
33.     public void init(FilterConfig config) throws ServletException {
34.         System.out.println("过滤器初始化");
35.     }
36. }</span>
```

过滤器中使用了@WebFilter(filterName="Servlet3Filter",urlPatterns="/*")进行配置,Filter的名字是Servlet3Filter,表示urlPatterns="/*"该Filter拦截所有的请求。

web服务器启动时就会初始化该Filter,如下图所示：



当有用户访问web应用中的资源时，该过滤器就会进行拦截，如下图所示：



可以看到，使用@WebFilter注解，我们可以将一个实现了javax.servlet.Filter接口的类定义为过滤器，用urlPatterns属性或者value属性指定要过滤的URL模式。

可以指定多种过滤模式@WebFilter(filterName="Servlet3Filter",urlPatterns={"/UserManagerServlet","/index.jsp"})

有了@WebFilter注解之后，我们的web.xml就无需任何配置了

```
[java]
01. <span style="font-size:14px;">
02. </span>

[java]
01. <span style="font-size:14px;"><?xml version="1.0" encoding="UTF-8"?>
02. <web-app version="3.0"
03.     xmlns="http://java.sun.com/xml/ns/javaee"
04.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
05.     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
06.         http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
07.     <display-name></display-name>
08.     <welcome-file-list>
09.         <welcome-file>index.jsp</welcome-file>
10.     </welcome-file-list>
11. </web-app></span>

[java]
01. <span style="font-size:14px;">
02. </span>
```

这样我们的web.xml文件就非常干净了。可以说，Servlet3.0规范的出现，大大减少了开发Servlet和Filter配置方面的工作

from: <http://www.cnblogs.com/xdp-gacl/p/4224837.html>