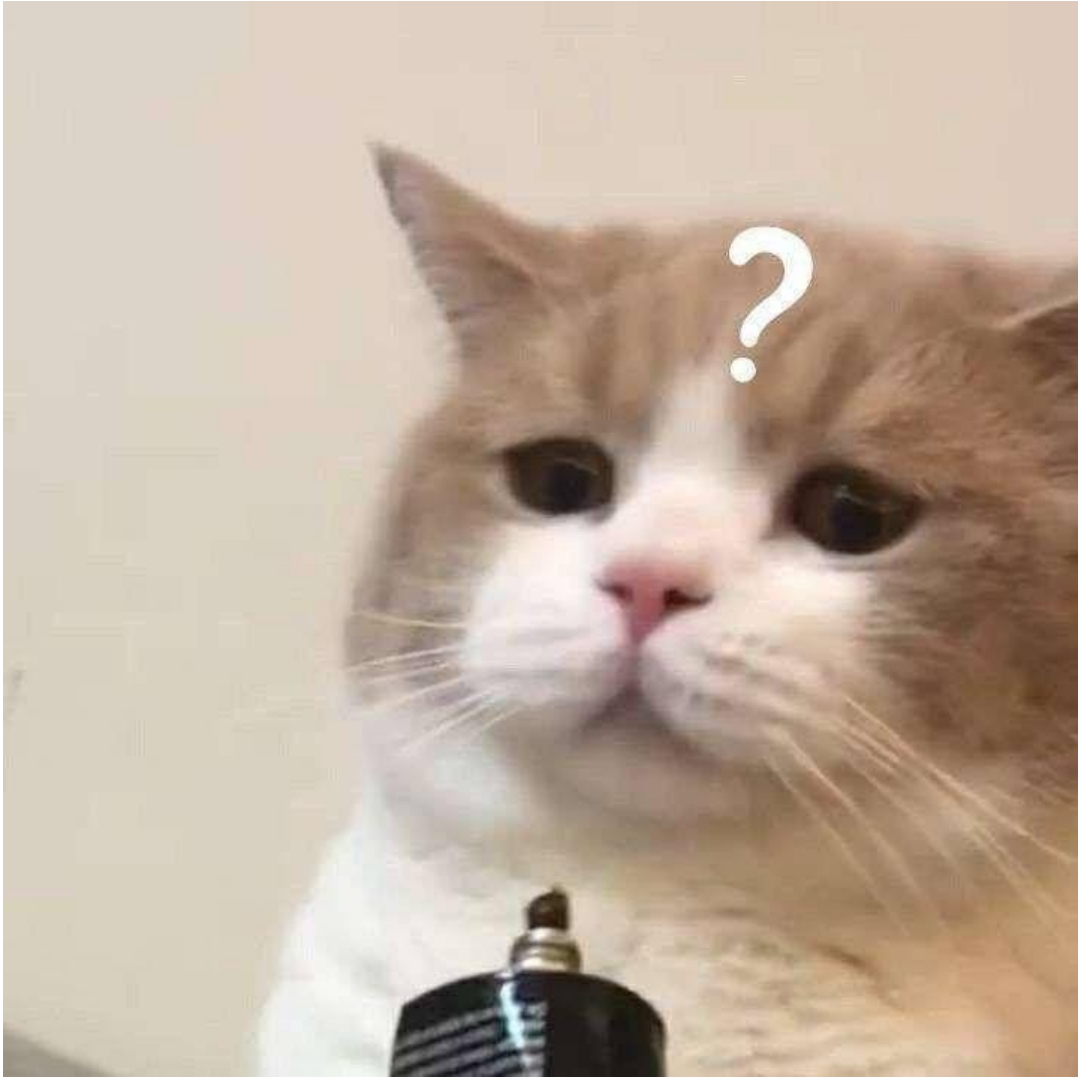
链滴

WebFilter-SpringBoot 过滤器注解实例讲解

作者：adlered
原文链接：https://ld246.com/article/1552525751616
来源网站：链滴



简介

Filter也称之为过滤器，它是Servlet技术中最让人激动的技术，Web开发人员通过Filter技术，对We服务器管理的所有Web资源：例如JSP，Servlet，静态图片文件或静态HTML文件等进行拦截，从而现一些特殊的功能，例如实现URL级别的权限访问控制，过滤敏感词汇，压缩响应信息等一些高级功。

可以在web.xml中配置过滤器的生效URL地址，它会过滤指定的URL地址经过的数据”/*”代表过滤有的URL地址请求数据。

这样做的好处有什么呢，可以直接配置过滤所有的请求编码都设置为UTF-8，在response的时候也可将响应编码也设置为UTF-8，这样就省去了写相同代码的步骤，优化代码。

Filter链

在一个Web应用中，可以开发编写多个Filter，这些Filter组合起来称之为一个Filter链。

Web服务器根据Filter在web.xml文件中的注册顺序，决定先调用哪个Filter，当第一个Filter的dofilte方法被调用时，web服务器会创建一个代表filter链的FilterChain对象传递给该方法，在doFilter()方中，开发人员如果调用FilterChain对象的doFilter()方法，则Web服务器会检查FilterChain对象中是还有Filter，如果有，则调用第二个Filter，如果没有，则调用目标资源。

原文链接：WebFilter-SpringBoot 过滤器注解实例讲解

Filter匹配

精确匹配：

/路径/资源名

如：/index.html 、/hello/index.jsp 、 /client/LoginServlet

精确匹配只要在请求地址完全一样时才会调用Filter

路径匹配：

/路径名/*

如：/hello/* 、/*

路径匹配只要是所请求的资源是在设置的路径下就会调用Filter

如：/hello/* 只要访问 项目根目录/hello/ 下的任意资源就会调用Filter

如: /* 只要访问 项目根目录下的资源就会调用Filter

后缀匹配：

*.后缀名

如：*.jsp 、 *.html

后缀匹配只要访问的资源路径是以指定后缀结尾就会调用Filter

Filter的生命周期

Filter的创建好销毁都由Web服务器负责，Web应用程序启动时，Web服务器将创建Filter的实例化象，并调用其init()方法，完成对象的初始化功能，从而为后续的用户请求做好拦截的准备工作，Filte对象只会创建一次，init()方法也只会执行一次。通过init()方法的参数，可获得代表当前Filter配置信的FilterConfig的对象。

Web容器调用destroy()方法销毁Filter，destroy()方法在Filter的生命周期中仅执行一次；在destroy()法中，可以释放过滤器使用的资源。

FilterConfig接口

用户在配置Filter时，可以使用<init-param>为Filter配置一些初始化参数，当Web容器实例化Filter象，调用其init()方法时，会把封装了Filter初始化参数的filterConfig对象传递进来。因此开发人员在写Filter时，通过filterConfig对象的方法，就可获得：

String getFilterName()

得到Filter的名称。

String getInitParameter(String name)

返回在部署描述中指定名称的初始化参数的值。如果不存在返回null。

Enumeration getInitParameterNames()

返回过滤器的所有初始化参数的名字的枚举集合。

public ServletContext getServletContext()

返回Servlet上下文对象的引用。

使用

Filter的声明方式有两种：

第一种是使用注解的方式声明

第二种是在web.xml中使用标签的方式进行声明

注解Filter

使用@WebFilter(filterName = "过滤器名字", urlPatterns = "过滤地址", description = "描述")

举个栗子

```
@Controller
@WebFilter(filterName = "TestFilter", urlPatterns = "/*.do")
public class TestFilter implements Filter {

    public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws
ervletException, IOException {
        System.out.println("开始过滤2");
        chain.doFilter(req, resp);
    }

    public void init(FilterConfig config) throws ServletException {
        System.out.println("创建filter对象2");
    }

    public void destroy() {
        System.out.println("filter已经被销毁2");
    }
}

@Controller
@WebFilter(filterName = "TestFilter2", urlPatterns = "/*.do")
public class TestFilter2 implements Filter {

    public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws
ervletException, IOException {
        System.out.println("开始过滤1");
        chain.doFilter(req, resp);
    }

    public void init(FilterConfig config) throws ServletException {
        System.out.println("创建filter对象1");
    }
```

```
    public void destroy() {
        System.out.println("filter已经被销毁1");
    }
}
```

通过注解的方式配置Filter过滤器

如果在SpringBoot里最好加上一个@Controller标签，Filter的执行顺序可以使用@Order(i)决定Filte的**执行顺序**，i的值越大执行顺序越靠前。

在web.xml里配置的Filter是由**上下顺序**决定的。

web.xml配置Filter

再举个栗子

```
<!-- Filter文件配置 -->
<filter>
    <filter-name>TestFilter</filter-name>
    <filter-class>cn.zlys.service.TestFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>TestFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- Filter的执行顺序和配置的顺序有关 -->
<filter>
    <filter-name>TestFilter2</filter-name>
    <filter-class>cn.zlys.service.TestFilter2</filter-class>
</filter>
<filter-mapping>
    <filter-name>TestFilter2</filter-name>
    <url-pattern>/*.do</url-pattern>
</filter-mapping>
```

在web.xml中配置的好处是在项目发布后可以不更改项目源码就可以更改Filter执行的URL地址或者是Fi ter的class路径。并且通过web.xml配置Filter还可以选择过滤的方式，比如只过滤请求或者过滤ERRO 。

使用dispatcher标签决定过滤的方式，一共有REQUEST、ERROR、FORWARD、INCLUDE，分别是求、错误、转发、包含，包含是JSP里的include标签。

在Filter中也有init-param标签，可以在项目初始化时获取数据，比如说在init-param标签里配置编格式为UTF-8格式，这样的话就可以使项目的编码动态的更改，如果想要在项目发布后更改项目的编格式，直接更改web.xml中init-param里的数据就可以了。

总结

Filter的发明真的是便利的我们的生活，可以省略不必要的代码，如果是请求的编码确认可以不用写多遍，因为如果Serlvet过多时，在每个Servlet都要写设置编码格式，如果想要更改编码格式的时候需要每个 Servlet都重写一遍，但是有了Filter就可以只写一遍，就实现了在每个Servlet里都更改了请编码格式和响应编码格式。

Filter不仅可以通过url-pattern来拦截指定url地址匹配的资源，而且还可以通过servlet-name来指定截哪个指定的servlet(专门为某个servlet服务，servlet-name对应Servlet相关配置)

原文链接：WebFilter-SpringBoot 过滤器注解实例讲解