

<http://blog.csdn.net/ggibenben1314/article/details/45341855>

在struts2中用过filter过滤器，在springmvc中还有拦截器，它们都能过滤请求，但是到底有什么区别呢？

一、定义

拦截器：是在面向切面编程的就是在你的service或者一个方法，前调用一个方法，或者在方法后调用一个方法比如动态代理就是拦截器的简单实现，在你调用方法前打印出字符串（或者做其它业务逻辑的操作），也可以在你调用方法后打印出字符串，甚至在你抛出异常的时候做业务逻辑的操作。

过滤器：是在javaweb中，你传入的request、response提前过滤掉一些信息，或者提前设置一些参数，然后再传入servlet或者struts的action进行业务逻辑，比如过滤掉非法url（不是login.do的地址请求，如果用户没有登陆都过滤掉），或者在传入servlet或者struts的action前统一设置字符集，或者去除掉一些非法字符。

二、xml文件配置

1.filter

该过滤器的方法是创建一个类XXXFilter实现此接口，并在该类中的doFilter方法中声明过滤规则，然后在配置文件web.xml中声明他所过滤的路径

```
1. <filter>
2.   <filter-name>XXXFilter</filter-name>
3.   <filter-class>
4.     com.web.util.XXXFilter
5.   </filter-class>
6. </filter>
7.
8. <filter-mapping>
9.   <filter-name>XXXFilter</filter-name>
10.  <url-pattern>*.action</url-pattern>
11. </filter-mapping>
```

2.Interceptor

它也要实现HandlerInterceptor 接口，这里只介绍方法注解配置

```
1. <!-- 拦截器 -->
2. <mvc:interceptors>
3.     <!-- 多个拦截器,顺序执行 -->
4.     <mvc:interceptor>
5.         <mvc:mapping path="/entryOrJsonController/*" /><!-- 如果不配置
或/*,将拦截所有的Controller -->
6.         <bean class="com.wy.interceptor.CommonInterceptor"></bean>
7.     </mvc:interceptor>
8. </mvc:interceptors>
```

三、具体区别

	filter	Interceptor
多个的执行顺序	根据filter mapping配置的先后顺序	按照配置的顺序，是可以通过order顺序
规范	在Servlet规范中定义的，是Servlet容器支持的	Spring容器内的，Spring框架支持的
使用范围	只能用于Web程序中	既可以用于Web程序，也可以用于Application、Sw程序中。
深度	Filter在只在Servlet前后起作用	拦截器能够深入到方法前后、异常抛出后等

四、总结

两者的本质区别：拦截器是基于**Java**的反射机制的，而过滤器是基于函数回调。从灵活性上说拦截器功能更强大些，Filter能做的事情，他都能做，而且可以在请求前，请求后执行，比较灵活。Filter主要是针对URL地址做一个编码的事情、过滤掉没用的参数、安全校验（比较泛的，比如登录不登录之类），太细的话，还是建议用interceptor。不过还是根据不同情况选择合适的。