

filter

一、Filter简介与快速入门

1.1 简介

filter是对客户端访问资源的过滤，符合条件放行，不符合条件不放行，并且可以对目标资源访问前后进行逻辑处理。

1.2 快速入门

1. 编写一个过滤器的类实现Filter接口
2. 实现接口中尚未实现的方法(着重实现doFilter方法)
3. 在web.xml中进行配置(主要是配置要对哪些资源进行过滤)

```
15 @Override
16 //Filter创建的时候执行init方法
17 public void init(FilterConfig filterConfig) throws ServletException {
18     //1. 获得web.xml中filter 的名称<filter-name>QuickFilter1</filter-name>
19     System.out.println(filterConfig.getFilterName()); //过滤器的名字
20     //2. 获得当前filter的初始化参数
21     System.out.println(filterConfig.getInitParameter("aaa")); //init-param
22     //3. 获得ServletContext
23     filterConfig.getServletContext();
24
25     System.out.println("init ....");
26 }
27
28 @Override
29 //doFilter是Filter的核心过滤的方法
30 /*
31  * request: 内部封装是客户端http请求的内容
32  * response: 代表是响应
33  * FilterChain: 过滤器链对象
34  */
35 public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
36     throws IOException, ServletException {
37
38     System.out.println("quick1 running....");
39     //放行请求
40     chain.doFilter(request, response); 放行
41 }
42
43 @Override
44 //Filter对象销毁的时候执行destroy方法
45 public void destroy() {
46     System.out.println("destroy...");
47 }
48
49 }
```

我们着重重写doFilter方法，别的基本不管。

web.xml中的配置如下：

```

<filter>
  <filter-name>QuickFilter1</filter-name>
  <filter-class>com.itheima.web.filter.QuickFilter1</filter-class>
</filter>
<filter-mapping>
  <filter-name>QuickFilter1</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

```

和servlet的配置类似

二、Filter的API与生命周期

初始化方法：代表filter对象初始化方法，filter创建时执行，也就是**服务器启动时**就创建Filter

销毁方法：**服务器关闭时执行**，此时filter销毁

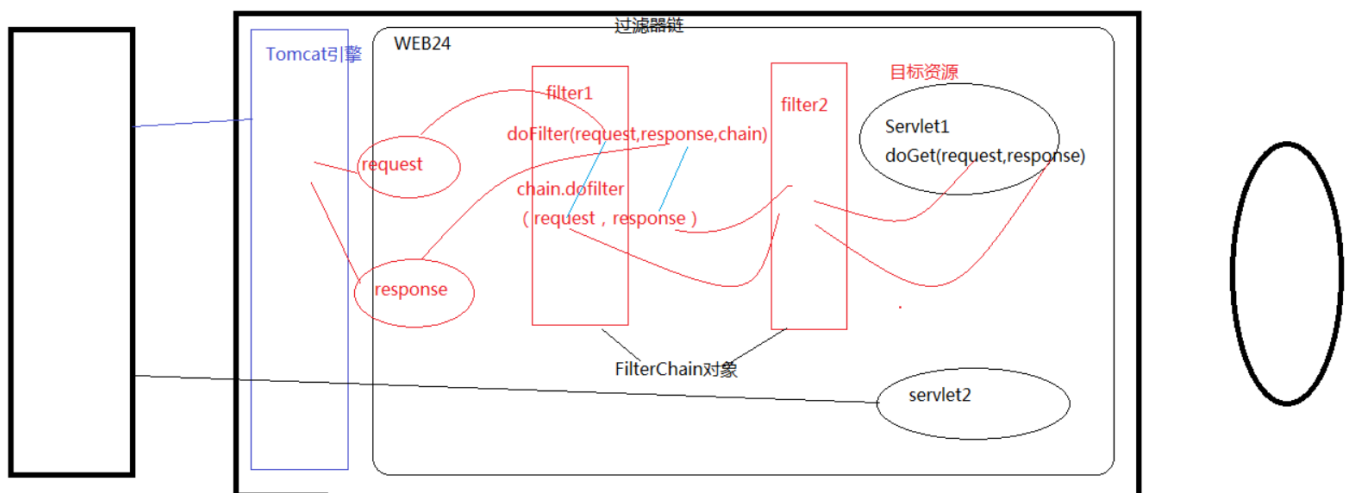
doFilter方法：代表filter执行过滤的核心方法，如果**某资源在已经被配置到这个filter进行过滤**的话，那么**每次访问这个资源都会执行doFilter方法**其中的参数：

ServletRequest/ServletResponse：每次在执行doFilter方法时 web容器负责创建一个request和一个response对象作为doFilter的参数传递进来。该request对象和response对象就是在访问目标资源的service方法时的request和response。

FilterChain：过滤器链对象，通过该对象的doFilter方法可以放行该请求

上面已经说明了Filter的生命周期。

三、Filter配置



如果某个过滤器的<filter-mapping>标签在前，那么该filter首先拦截。

<url-pattern>配置：

1. **完全匹配** /servlet1
2. **目录匹配** /aaa/bbb/* ----最多的
/user/：访问前台的资源进入此过滤器
/admin/：访问后台的资源时执行此过滤器
3. **扩展名匹配** *.abc *.jsp

注意：url-pattern可以使用servlet-name替代，也可以混用

dispatcher标签设置:

REQUEST: 默认值, 代表直接访问某个资源时执行filter

FORWARD: 转发时才执行filter

INCLUDE: 包含资源时执行filter

ERROR: 发生错误时 进行跳转是执行filter