Filter详解

Filter 过滤器的使用步骤：

1. 编写一个类去实现 Filter 接口
2. 实现过滤方法 doFilter()
3. 到 web.xml 中去配置 Filter 的拦截路径

代码示例：

```
public class AdminFilter implements Filter {
    /**
     * doFilter 方法，专门用于拦截请求。可以做权限检查
     */
    @Override
    public void doFilter(ServletRequest servletRequest, ServletResponse
servletResponse, FilterChain
            filterChain) throws IOException, ServletException {
        HttpServletRequest httpServletRequest = (HttpServletRequest)
servletRequest;
        HttpSession session = httpServletRequest.getSession();
        Object user = session.getAttribute("user");
       // 如果等于 null，说明还没有登录
        if (user == null) {

servletRequest.getRequestDispatcher("/login.jsp").forward(servletRequest,servlet
Response);
            return;
        } else {
            // 让程序继续往下访问用户的目标资源，非常重要，如果没有这个就算有权限也无法访问
            filterChain.doFilter(servletRequest,servletResponse);
        }
    }
}
```

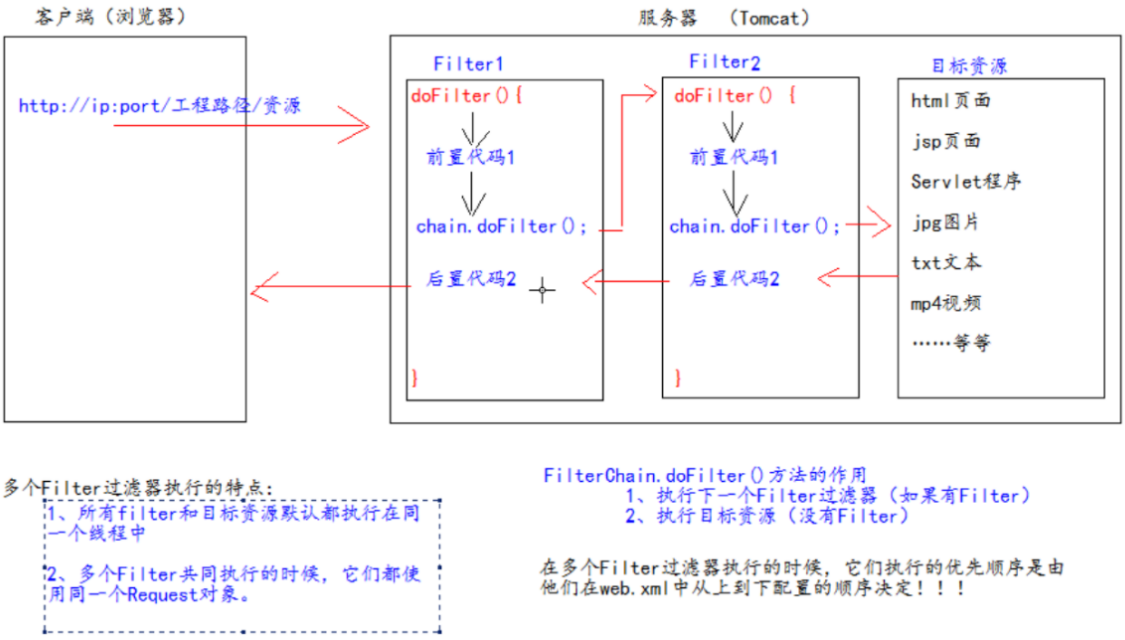
XML配置示例：

```
<!--filter 标签用于配置一个 Filter 过滤器-->
<!--filter-name 是给 filter 起一个别名-->
<!--filter-class是配置 filter 的全类名-->
<!--filter-mapping 配置 Filter 过滤器的拦截路径-->
<!--filter-name 表示当前的拦截路径给哪个 filter 使用-->
<!--url-pattern 配置拦截路径
    / 表示请求地址为：http://ip:port/工程路径/ 映射到 IDEA 的 web 目录
    /admin/* 表示请求地址为：http://ip:port/工程路径/admin/*
-->

<filter>
    <filter-name>CharacterEncodingFilter</filter-name>
    <filter-class>com.srx.servlet.filter.CharacterEncodingFilter</filter-class>
</filter>
```

```
<filter-mapping>
    <filter-name>CharacterEncodingFilter</filter-name>
    <url-pattern>/servlet/*</url-pattern>
</filter-mapping>
```

FilterChain 就是过滤器链（多个过滤器如何一起工作）



Filter 的拦截路径

```
--精确匹配
/target.jsp 以上配置的路径，表示请求地址必须为：http://ip:port/工程路径/target.jsp

--目录匹配
/admin/* 以上配置的路径，表示请求地址必须为：http://ip:port/工程路径/admin/*

--后缀名匹配
*.html 以上配置的路径，表示请求地址必须以.html 结尾才会拦截到

*.do 以上配置的路径，表示请求地址必须以.do 结尾才会拦截到

*.action 以上配置的路径，表示请求地址必须以.action 结尾才会拦截到
```

Filter 过滤器它只关心请求的地址是否匹配，不关心请求的资源是否存在！！

Filer的生命周期

Filter的生命周期可分为创建、执行、销毁三个阶段。

1.创建阶段：Web服务器启动的时候会创建Filter实例对象，并调用init()方法，完成对象的初始化。

2.执行阶段：当客户端请求目标资源时，服务器会筛选出符合映射条件的Filter，并按照Filter在web.xml中配置的顺序依次执行doFilter() 方法。

3.销毁阶段：服务器关闭时，Web服务器调用destroy()方法销毁Filter对象。