

过滤器是一个实现了 `javax.servlet.Filter` 接口的 Java 类。过滤器在Web服务器启动的时候就加载了，因为要拦截Web资源，所以必须在所有Web资源启动之前就加载过滤器。

1	public void doFilter (ServletRequest, ServletResponse, FilterChain) 该方法完成实际的过滤操作，当客户端请求方法与过滤器设置匹配的URL时，Servlet容器将先调用过滤器的doFilter方法。FilterChain用户访问后续过滤器。
2	public void init(FilterConfig filterConfig) web 应用程序启动时，web 服务器将创建Filter 的实例对象，并调用其init方法，读取web.xml配置，完成对象的初始化功能，从而为后续的用户请求作好拦截的准备工作（filter对象只会创建一次，init方法也只会执行一次）。开发人员通过init方法的参数，可获得代表当前filter配置信息的FilterConfig对象。
3	public void destroy() Servlet容器在销毁过滤器实例前调用该方法，在该方法中释放Servlet过滤器占用的资源。

doFilter中FilterChain.doFilter(Request, Response) 未执行时 servlet不执行，未放行。该方法后先执行servlet，再执行该方法后的语句（响应阶段？）。

可用ServletRequest的该方法实现跳转：

```
request .getRequestDispatcher ( "users.jsp").forward(  
request , response );
```

web.xml 文件配置

web.xml 的加载顺序是：context-param -> listener -
> filter -> servlet

```
<filter>  
    <filter-name>LogFilter</filter-name>
```

```

    <filter-
class>com.runoob.test.LogFilter</filter-class>
    <init-param>    /*指定初始化参数
        <param-name>Site</param-name>
        <param-value>菜鸟教程</param-value>
    </init-param>
</filter>
<filter-mapping>    /*配置拦截
    <filter-name>LogFilter</filter-name>
    <url-pattern>/*</url-pattern>    /*设置 filter 所拦
截的请求路径(过滤器关联的URL样式)
</filter-mapping>

```

使用 FilterConfig 对象获取参数：

```

public void  init(FilterConfig config) throws
ServletException {
    // 获取初始化参数
    String site = config.getInitParameter("Site");
    // 输出初始化参数
    System.out.println("网站名称: " + site);
}

```

若适用于所有的 Servlet，指定 /* 。也可以指定一个特定的 Servlet 路径。过滤器的应用顺序和配置的顺序相同。

- <servlet-name>指定过滤器所拦截的Servlet名称。
- <dispatcher>指定过滤器所拦截的资源被 Servlet 容器调用的方式，可以是REQUEST, INCLUDE, FORWARD和ERROR之一，默认REQUEST。用户可以设置多个<dispatcher>子元素用来指定 Filter 对资源的多种调用方式进行拦截。

- <dispatcher>子元素可以设置的值及其意义
 - REQUEST: 直接访问目标资源时执行过滤器。当用户直接访问页面时, Web容器将会调用过滤器。包括: 在地址栏中直接访问、表单提交、超链接、重定向, 只要在地址栏中可以看到目标资源的路径, 就是REQUEST;
 - INCLUDE: 包含访问执行过滤器。如果目标资源是通过Request Dispatcher的include()方法, <jsp:include>访问时, 那么该过滤器将被调用。
 - FORWARD: 转发访问执行过滤器。如果目标资源是通过Request Dispatcher的forward()方法, <jsp:forward>访问时, 那么该过滤器将被调用。
 - ERROR: 如果目标资源是通过声明式异常处理机制调用时, 当目标资源在web.xml中配置为<error-page>中时, 那么该过滤器将被调用。