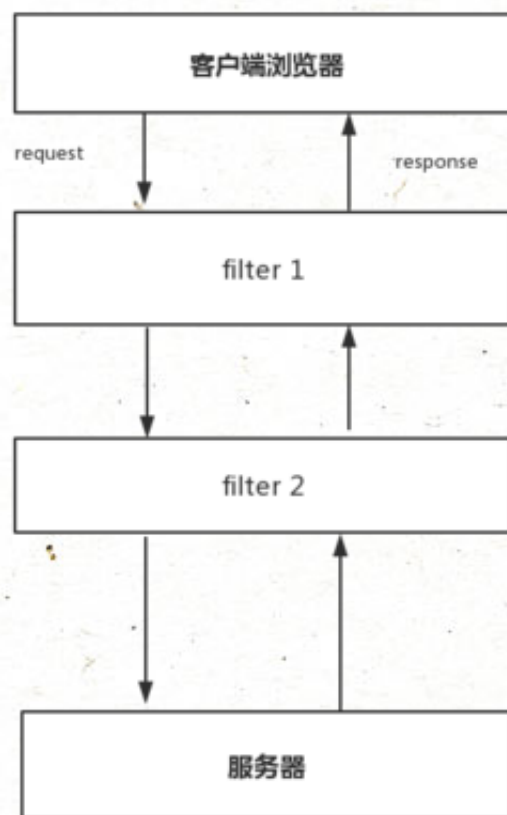


<http://www.jianshu.com/p/ad0bd3a14045>

一、filter简介

filter是Servlet规范里的一个高级特性，只用于对request、response的进行修改。

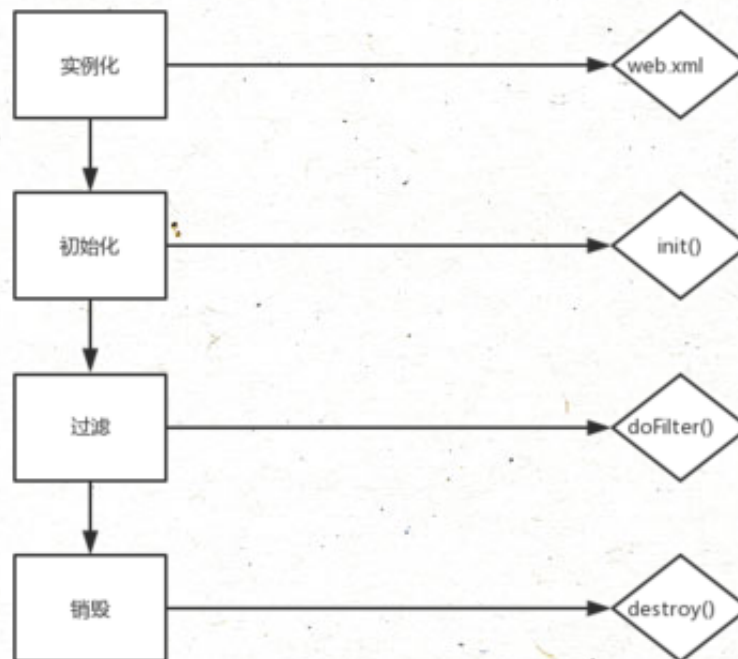
filter提出了FilterChain的概念，客户端请求request在抵达servlet之前都会经过filterChain里的所有filter，如图所示：



filterchain工作原理，（用processOn画的）

二、filter的生命周期

在web.xml中配置filter，当启动服务器时会实例化，并且会初始化，当有网络请求时会进行过滤操作，当服务器关闭时，会进行销毁，全过程如下图所示：



filter生命周期

三、编写第一个filter

filter类需实现filter接口，需复写里面的三个方法，其中init(),在初始化时调用；doFilter()方法每次都会调用，在这个方法中一定要执行chain.doFilter(),否则request不会交给后面的filter或者servler;ondestroy()在关闭服务器时调用。

```
public class FirstFilter implements Filter {
    @Override
    public void destroy() {
        System.out.println("destroy---FirstFilter");
    }
    @Override
    public void doFilter(ServletRequest request,
        ServletResponse response, FilterChain chain) throws IOException,
        ServletException {
        System.out.println("start----doFilter--FirstFilter");
        chain.doFilter(request, response);
        System.out.println("end-----doFilter--FirstFilter");
    }
    @Override
    public void init(FilterConfig filterConfig) throws ServletException
```

```
{
    System.out.println("init----FirstFilter");
}
```

配置filter:

```
<filter>
    <filter-name>FirstFilter</filter-name>
    <filter-class>com.forezp.filter.FirstFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>FirstFilter</filter-name>
    <url-pattern>/index.jsp</url-pattern>
    <url-pattern>*.do</url-pattern>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

其中，url_pattern可以配置多个，也可以用通配符，当访问满足路径匹配，并且符合dispatcher时，request会被filter拦截进行处理，处理完后的response再次被filter拦截，可以进行处理。

其中dispatcher 默认REQUEST，四种不同的dispatcher:

- REQUEST:请求时有效
- FORWARD:当某servlet通过forward到该servlet才有效
- INCLUDE: jsp通过< jsp: incluser/> 请求servlet有效
- ERROR: < %@page errorPage="" % >有效

四、防盗链

filter的特性使它可以处理特殊的工作，例如防盗链，字符编码的处理，日志记录，数据加密，过滤一些黑词等等。

例如：防盗链图片，当其他网站请求本网站图片资源时显示错误的图片，只有本应用先生的图片才显示正确的图片，代码如下：

```
public class ImageFilter implements Filter{
    public void init(FilterConfig config) throws Exception{}
}

public void doFilter(ServletRequest req,ServletResponse
res,FilterChain chain)throws Exception{
    HttpServletRequest request=(HttpServletRequest )req;
    HttpServletResponse response=(HttpServletResponse)res;
    String referer=request.getHeader("referer");
```

```

        if(referer==null||!referer.contains(request.getServerName())){
request.getRequestDispatcher("/error.png").forwar(request,response);
        }else{
            chain.doFilter(request,response);
        }
    }
    public void destroy(){}
}

```

在web.xml中配置:

```

<filter>
    <filter-name>imageFilter</filter-name>
    <filter-class>com.forezp.ImageFilter </filter-class>
</filter>
<filter-mapping>
    <filter-name>imageFilter</filter-name>
    <url-pattern> /images/* </url-pattern>
</filter-mapping>

```

当访问images下的所有图片会经过该filter,根据访问头信息,如果说本站点的访问则显示正确图片,否则先生错误图片。

五、字符编码

直接上代码:

```

public class CharsetFilter implements Filter{
    private String characterEncoding;
    private String enabled;
    public void init(FilterConfig config) throws Exception(){
        characterEncoding=config.getInitParameter("characterEncoding");
        enabled=config.getInitParameter("enabled").equals("true");
    }
    public void doFilter(ServletRequest req,ServletResponse
res,FilterChain chain)throws Exception{
        if(enabled|| characterEncoding!=null){
            req.setCharacterEncoding(characterEncoding);
            res.setCharacterEncoding(characterEncoding);
        }
        chain.doFilter(req,res);
    }
    public void destroy(){
        characterEncoding=null;
    }
}

```

```
}
```

在web.xml中配置：

```
<filter>
    <filter-name>CharsetFilter</filter-name>
    <!-- spring有专门处理字符编码的CharacterEncodingFilter,
        也可以用这个处理编码, 不需要自己写Filter处理 -->
    <filter-class>com.forezp.CharsetFilter </filter-class>
    <init-param>
        <param-name>characterEncoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
        <param-name>enabled</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>CharsetFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

其中页面编码方式也必须一致，希望全部用utf-8，另外需要配置Tomcat的 / config/server.xml编码：

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" URIEncoding="UTF-8"/>
```

另外，还有比较常见的日志记录filter、异常捕捉filter、权限校验、内容替换filter等等。

filter有很大的弹性机制，功能强大，而且跟servlet、jsp没耦合。filter是现在面向切面编程aop的一种思想体现，它能够胜任很多工作。

2.5的filter需要在web.xml中配置，执行顺序按照配置顺序，另外3.0可以用注解的方式配置filter，此时没有配置的顺序。