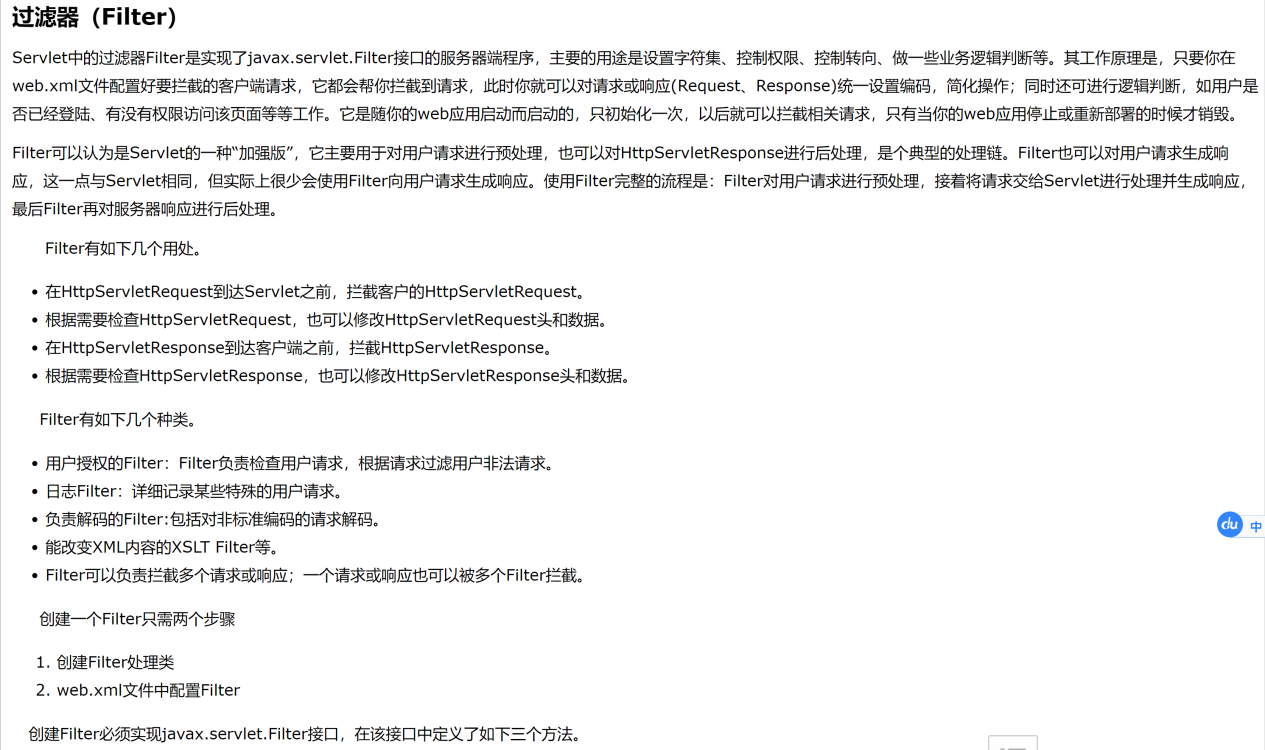
# Filter

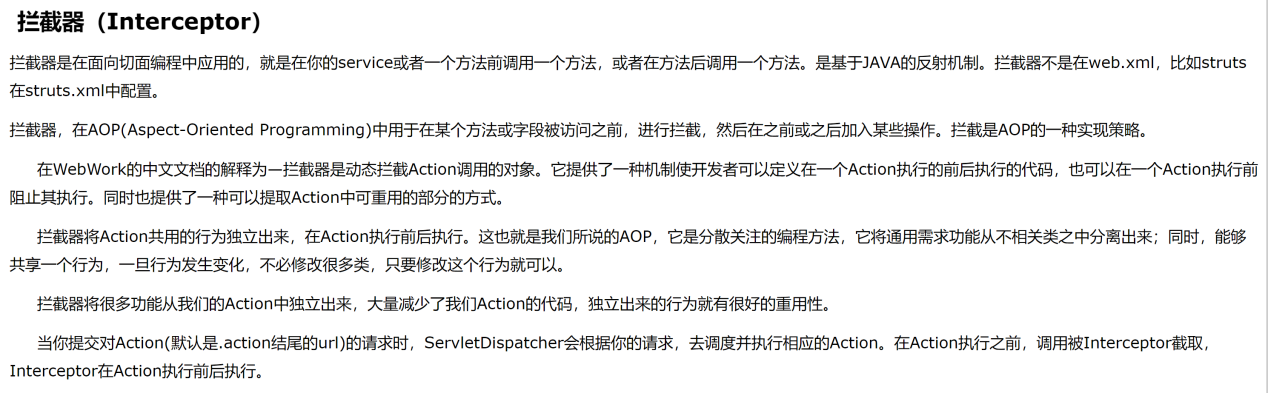
过滤器

Filter依赖于Servlet



# Interceptor

拦截器，基于java反射机制，属于面向切面编程的一种运用。动态代理就是拦截器的简单实现。





# Listener

监听器

# 区别

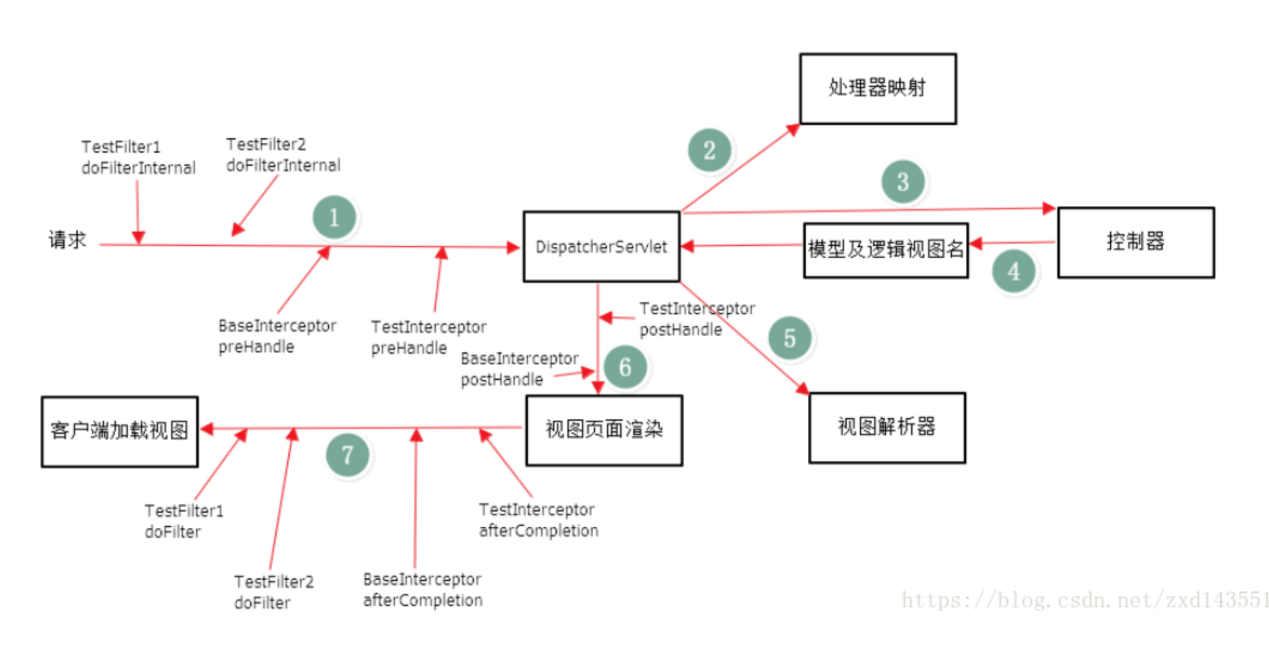


两者的本质区别：拦截器（Interceptor）是基于Java的反射机制，而过滤器（Filter）是基于函数回调。从灵活性上说拦截器功能更强大些，Filter能做的事情，都能做，而且可以在请求前，请求后执行，比较灵活。Filter主要是针对URL地址做一个编码的事情、过滤掉没用的参数、安全校验（比较泛的，比如登录不登录之类），太细的话，还是建议用interceptor。不过还是根据不同情况选择合适的。

# 执行顺序

context-param -> listener -> filter -> servlet

过滤前-拦截前-Action处理-拦截后-过滤后



# 生命周期

## Listener

它也是随web应用的启动而启动，只初始化一次，随web应用的停止而销毁

## Filter

(1)、启动服务器时加载过滤器的实例，并调用init()方法来初始化实例；

(2)、每一次请求时都只调用方法doFilter()进行处理；

(3)、停止服务器时调用destroy()方法，销毁实例。

## Intercepter

## Servlet

一般集成HttpServlet，servlet只会在第一次请求的时候被加载和实例化。servlet一旦被加载，一般不会从容器中删除，直至应用服务器关闭或重新启动

初始化：init()

调用：doGet(),doPost()每次访问都会执行

销毁：destroy()。

# Struts2和Springmvc区别

