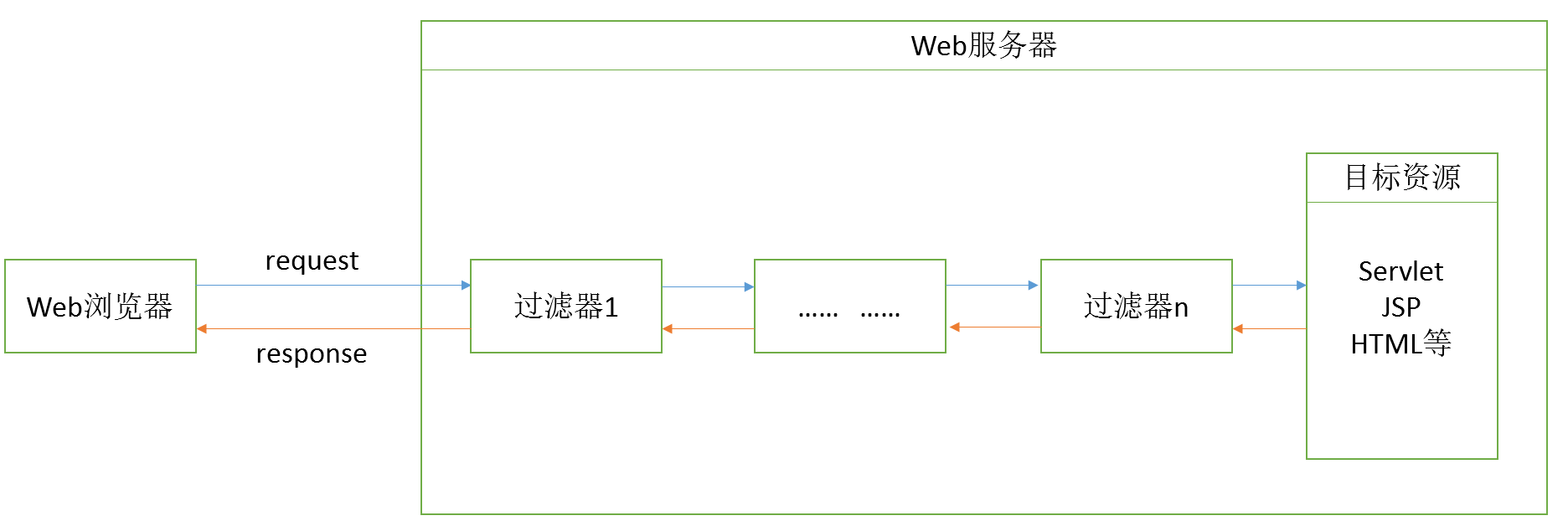
# 过滤器

## 过滤器的应用背景

### 本质

1. 过滤器是一个实现Filter接口的类，在Servlet之前就加载完成。过滤器不是一个普通的Servlet，不能处理请求也不能做出响应。主要用于对HttpServletRequest进行预处理，也可以对HttpServletResponse进行后处理。
2. 过滤器位于客户端和 web 应用程序之间，用于检查和修改两者之间流过的请求和响应。在请求到达 Servlet 或者 JSP 之前，过滤器截获请求。在响应送给客户端之前，过滤器截获响应。
3. 多个过滤器可以形成一个过滤器链，过滤器链中不同过滤器的**先后顺序**由部署文件 web.xml 中过滤器映射<filter-mapping>的顺序决定

## 过滤器的基本原理



在 Servlet 作为过滤器使用时，它可以对客户的请求进行处理。处理完成之后，它会交给下一个过滤器处理，这样，客户的请求在过滤器链中逐个处理，直到请求发送到响应的处理Servlet为止。

## 过滤器的简单开发

### 基本步骤

1. 新建一个类实现Filter接口。
2. 在web.xml中注册过滤器（包括基本信息和调用顺序）。
3. 可以用注解@WebFilter注册过滤器（注解和xml文件二选一）

### Filter接口

default public void init(FilterConfig filterConfig) throws ServletException {}

由 Web 容器调用，初始化此 Filter

default public void destroy() {}

由 web 容器调用，销毁此 Filter

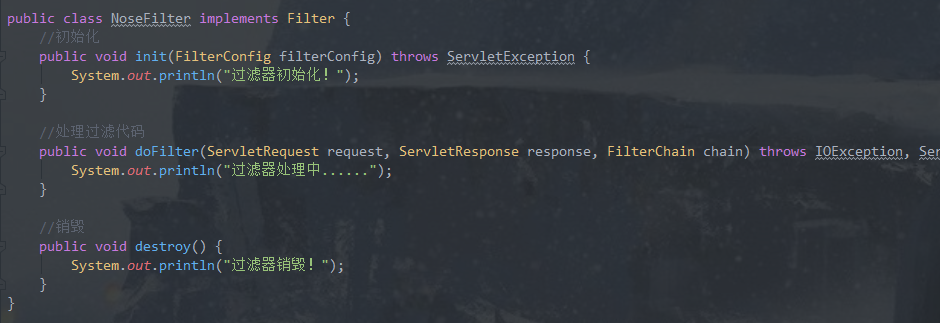
public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain)throws IOException, ServletException;

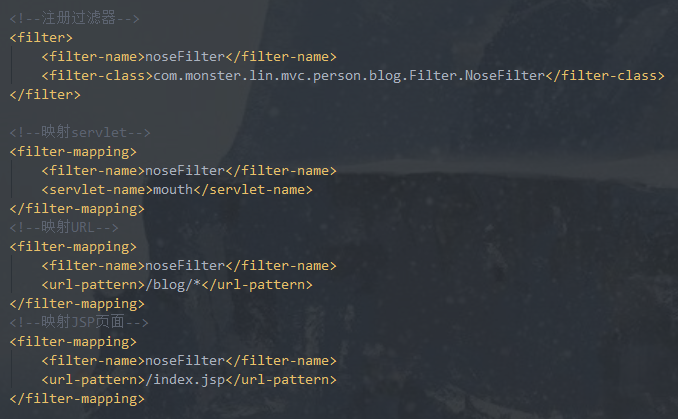
具体处理过滤的代码

### 过滤器实现

1. 过滤器类



1. web.xml注册过滤器



在实际应用中，要特别注意过滤器链的执行顺序问题，web 服务器根据过滤器在 web.xml 文件中的注册顺序，决定先调用哪个过滤器，当第一个过滤器的 doFilter 方法被调用时，web 服务器会创建一个代表过滤器链的 FilterChain 对象传递给该方法。在 doFilter 方法中，开发人员如果调用了 FilterChain 对象的 doFilter 方法，则 web 服务器会检查 FilterChain 对象中是否还有 filter，如果有，则调用第 2 个 filter，如果没有，则调用目标资源。

1. 注解@WebFilter注册过滤器



## Filter简单实例



1. 处理http请求时，将ServletRequest转换为HttpServletRequest。
2. 处理完成后，调用chain将请求和响应传回过滤链。