

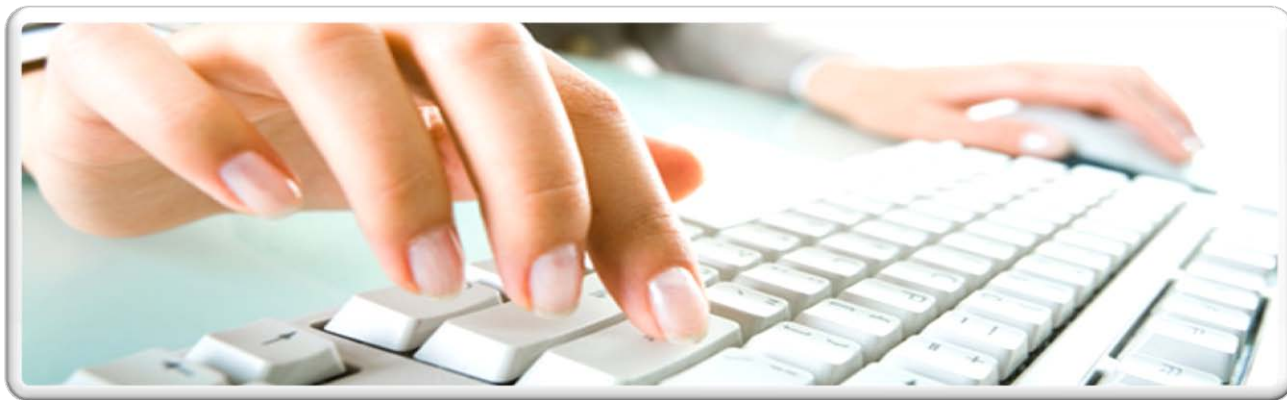
# Java web程序开发

过滤器

# 本章内容

- 过滤器定义与实现
- 过滤器链
- 过滤器应用
  - ▣ 中文乱码处理
  - ▣ 验证用户登录
  - ▣ 访问地址过滤
  - ▣ 网站logo过滤
- 扩展与复习：适配器模式，AOP与拦截器

# 案例说明问题？

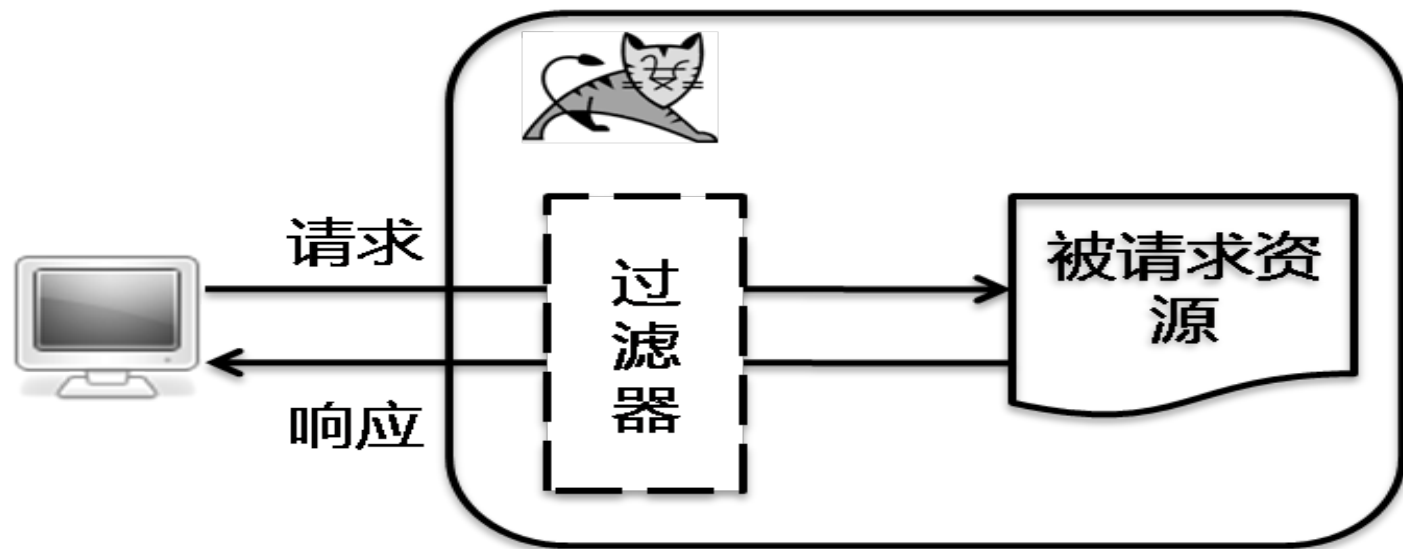


# 过滤器

## 过滤器

过滤器是向WEB应用程序在  
请求前和响应后添加新功能的组件

# 过滤器



# 过滤器实现

```
public class FirstFilter implements Filter {  
    public void init(FilterConfig arg0) throws ServletException {  
    }  
    public void destroy() {  
    }  
    public void doFilter(ServletRequest request, ServletResponse response,  
                        FilterChain chain) throws IOException, ServletException {  
    }  
}
```

# 过滤器生命周期

实例化

Servlet 容器创建过滤器的实例

初始化

该容器调用 `init()` 方法

过滤

如果符合过滤规则，则容器调用 `doFilter` 方法

销毁

销毁实例之前调用 `destroy()` 方法

不可用

销毁实例并标记为垃圾收集

# 定义映射

<filter>

<filter-name>FilterTimer</filter-name>

<filter-class>com.wangwang. FirstFilter </filter-class>

</filter>

<filter-mapping>

<filter-name>FilterTimer</filter-name>

<url-pattern>/\*</url-pattern>

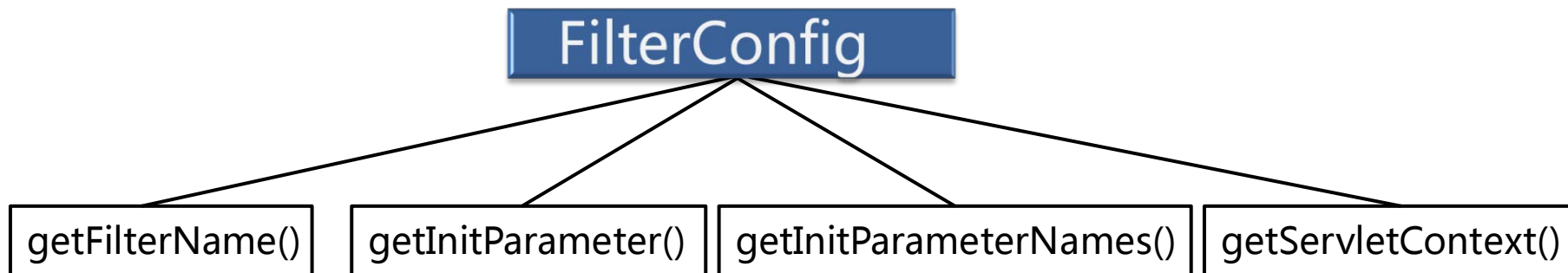
</filter-mapping>



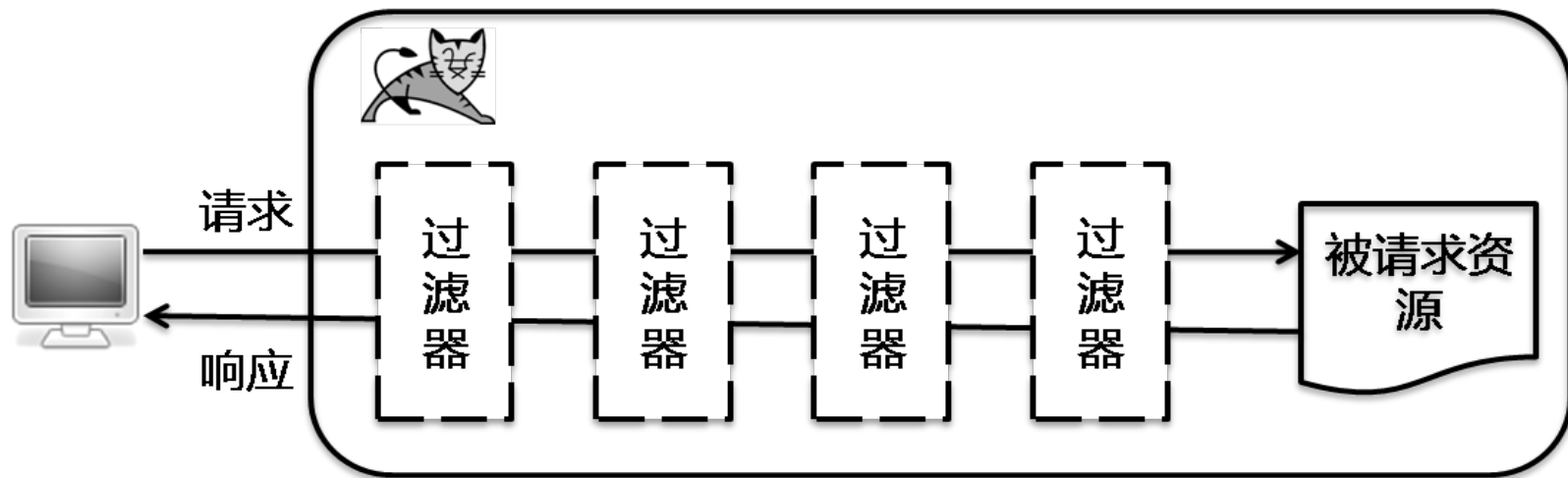
# 案例一：乱码处理过滤器



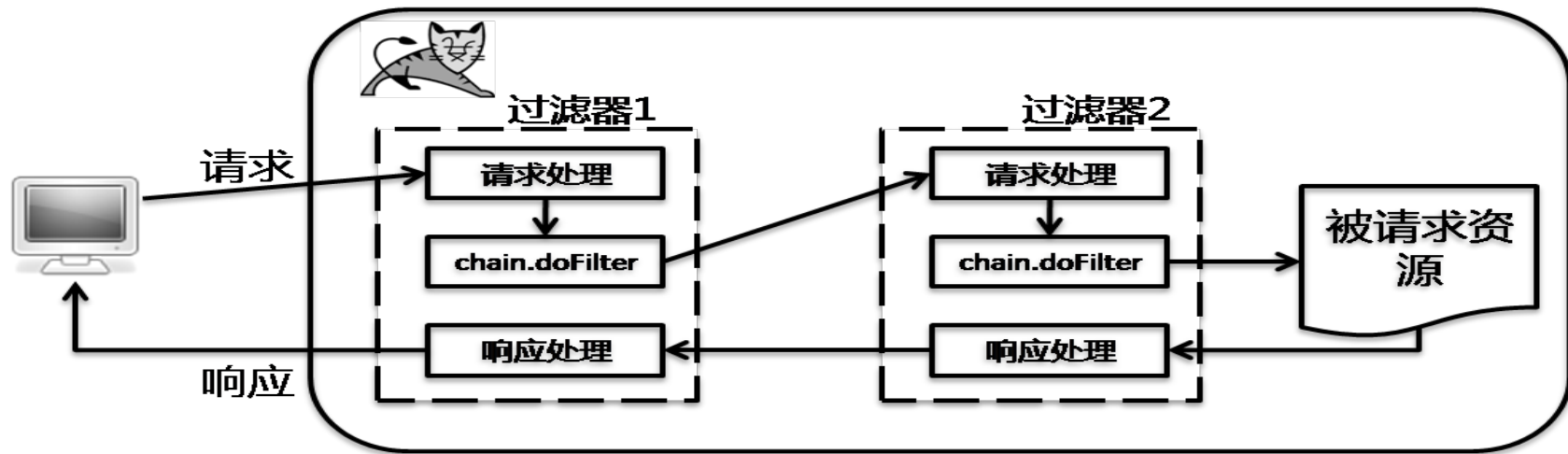
# FilterConfig



# 过滤器链



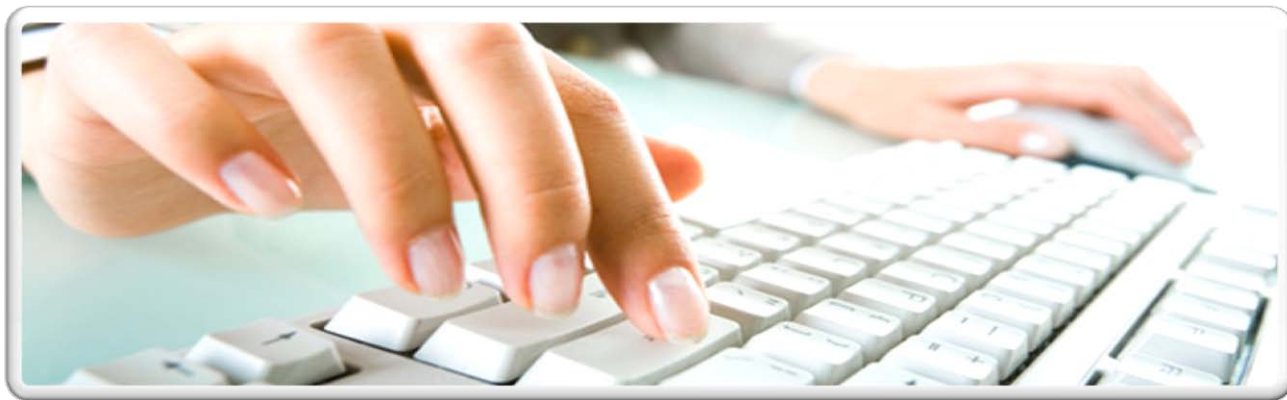
# 过滤器链的请求流程



# 适配器模式应用

- 大多数情况下只需要重写doFilter方法
- 使用适配器模式解决

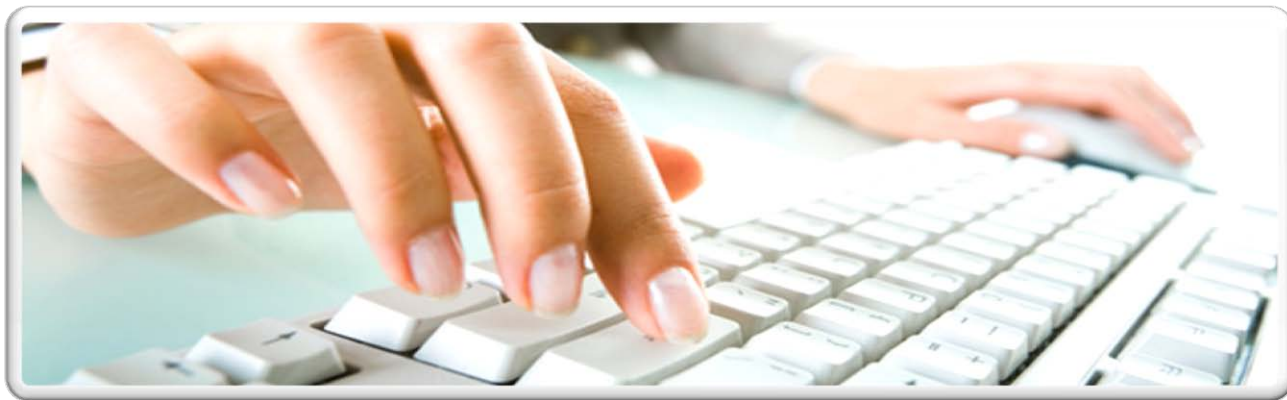
# 案例二：限制访问地址



# 案例三：权限管理过滤器



# 案例四：网站logo修改过滤器





# 与AOP与拦截器

- AOP-面向方面编程
- 拦截器

# 本章内容

- 过滤器定义与实现
- 过滤器链
- 过滤器应用
  - ▣ 中文乱码处理
  - ▣ 验证用户登录
  - ▣ 访问地址过滤
  - ▣ 网站logo过滤
- 扩展与复习：适配器模式，AOP与拦截器



# 练习时间

旺旺老师

西安云工厂—专业信息技术服务商

[www.xaygc.com](http://www.xaygc.com)

# page指令属性

方法名	定义
language	定义要使用的脚本语言
import	将包和类导入 Scriptlet
buffer	设置用来存储客户端请求的缓冲区的大小
errorPage	定义处理异常的 JSP 页面
isErrorPage	表示当前页面能否作为错误页面
contentType	指定页面格式和所使用的字符集