

# Praktikum DBAE / WI

## Filter

Pascal Reuss, M.Sc.

Raum A08b Spl.  
Email: [reusspa@uni-hildesheim.de](mailto:reusspa@uni-hildesheim.de)

# Filter

Filterung der Eingaben des Benutzers (Request) und der Antworten des Servers (Response)



# Einsatz von Filtern

## Verschiedene Einsatzmöglichkeiten

- Daten komprimiert übertragen
- Request und Response loggen
- Benutzereingaben aus Formularen validieren
- Benutzer authentifizieren
- XSLT und Image-Conversion
- usw.

Trennung dieser Funktionalitäten von den JSPs/Servlets

# Implementierung

Zur Verfügung steht das Interface:

`javax.servlet.Filter`

Es enthält die Methoden:

`public void init (FilterConfig config)`

`public void doFilter (ServletRequest req, ServletResponse res, FilterChain chain)`

`public void destroy()`

Keine Basisimplementierung für Filter vorhanden (vgl. Servlet)

kann aber einfach implementiert werden (siehe Tutorium)

# Beispiel einer Basisimplementierung

```
public class BaseFilter implements Filter {  
    protected FilterConfig config;  
  
    @Override  
    public void destroy() {  
        config = null;  
    }  
  
    /**  
     * Tut nichts. Leitet nur die Anfrage und die Antwort weiter  
     */  
    @Override  
    public void doFilter(ServletRequest req, ServletResponse res,  
        FilterChain chain) throws IOException, ServletException {  
        chain.doFilter(req, res);  
    }  
  
    @Override  
    public void init(FilterConfig fConfig) throws ServletException {  
        config = fConfig;  
    }  
}
```

# Konfiguration eines Filters

Wird im “web library descriptor” vorgenommen (web.xml)

## Definition des Filters

- Name des Filters

- Klasse

- Init-Parameter (mit Name und Wert der Parameter)

## Filter Mapping

- URL Mapping

# Beispiel einer Konfiguration

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchemaInstance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/webapp_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/webapp_2_5.xsd" id="WebApp_ID"
version="2.5">

    <display-name>testfilter</display-name>

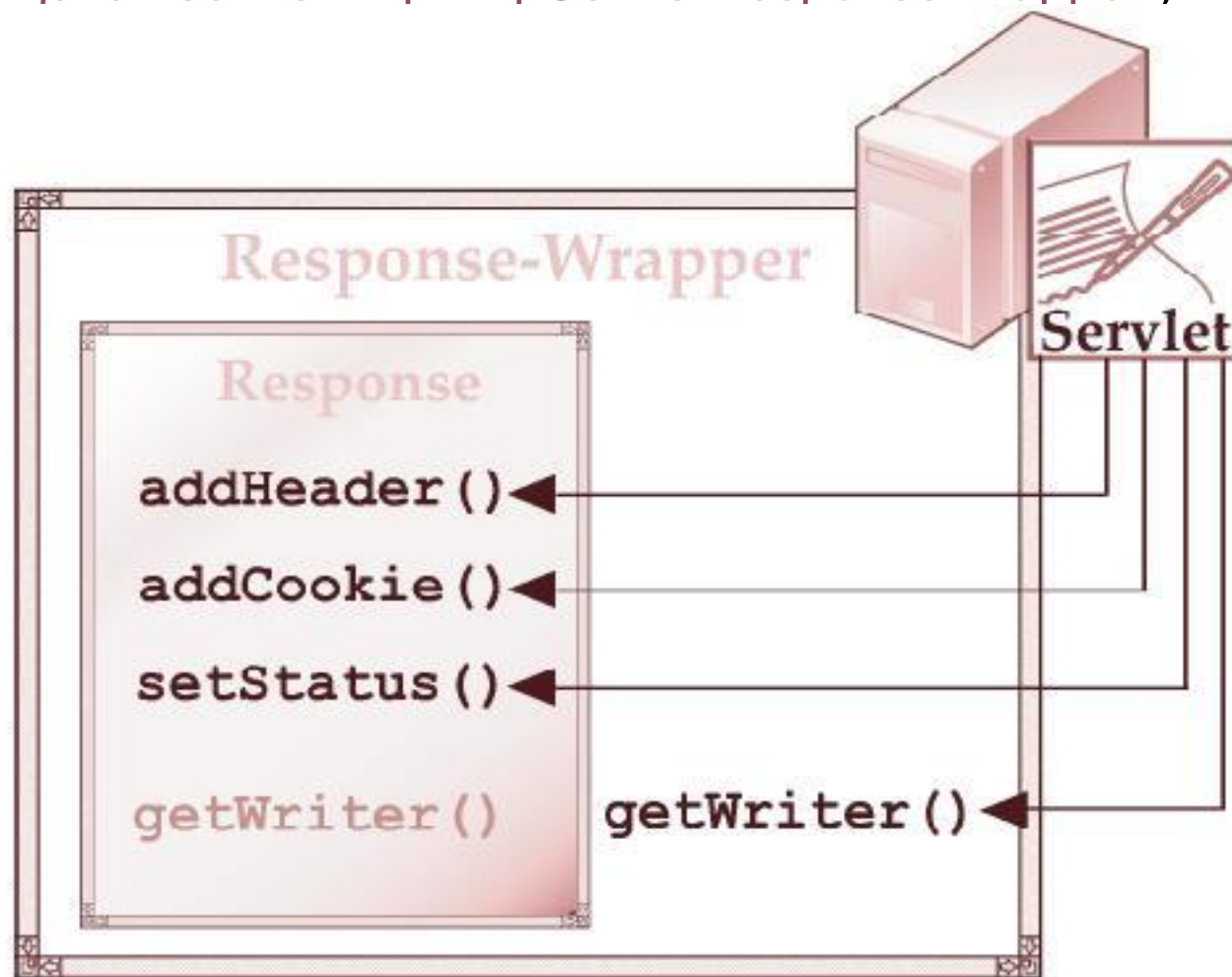
    <filter>
        <filter-name>ResponseFormatFilter</filter-name>
        <filter-class>filters.ResponseFormatFilter</filter-class>
    </filter>

    <filter-mapping>
        <filter-name>ResponseFormatFilter</filter-name>
        <url-pattern>/codebook.jsp</url-pattern>
    </filter-mapping>

</web-app>
```

# Manipulation von Request und Response

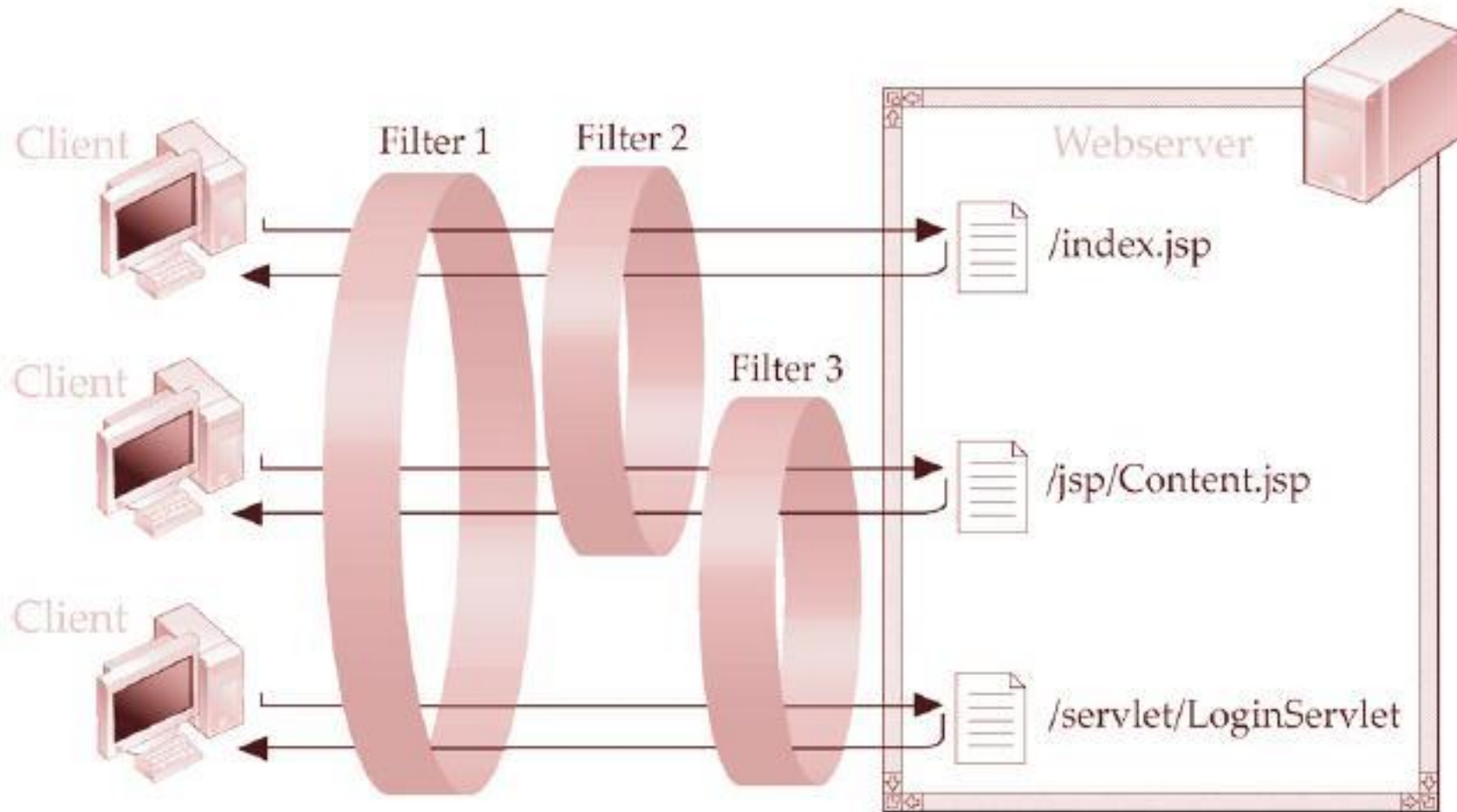
Wird in der Regel mit Hilfe von Wrapper-Klassen implementiert (z.B. Mit einer Unterklasse von `javax.servlet.http.HttpServletResponseWrapper` )





# Filterkette

Definition mehrerer Filter möglich → Reihenfolge der Filterung durch die einzelnen filter wird vorgegeben durch die Reihenfolge der Filter in Web.xml



# Referenzen

K. Samaschke, T. Stark  
Das J2EE Codebook