Servlet Filter
Why do we need Filter?

Jakarta Servlet & JSP

Miss Xing

Why do we need Filter?

- Suppose you want to implement the following task:
- A resource is accessible only when the user session is valid.
  - Use session attribute
- If lots of servlets and JSPs need the same feature
  - Check in every servlet and JSP?
  - Maintenance, DRY
- If later on, we want to change the attribute name
  - Require changes in every place

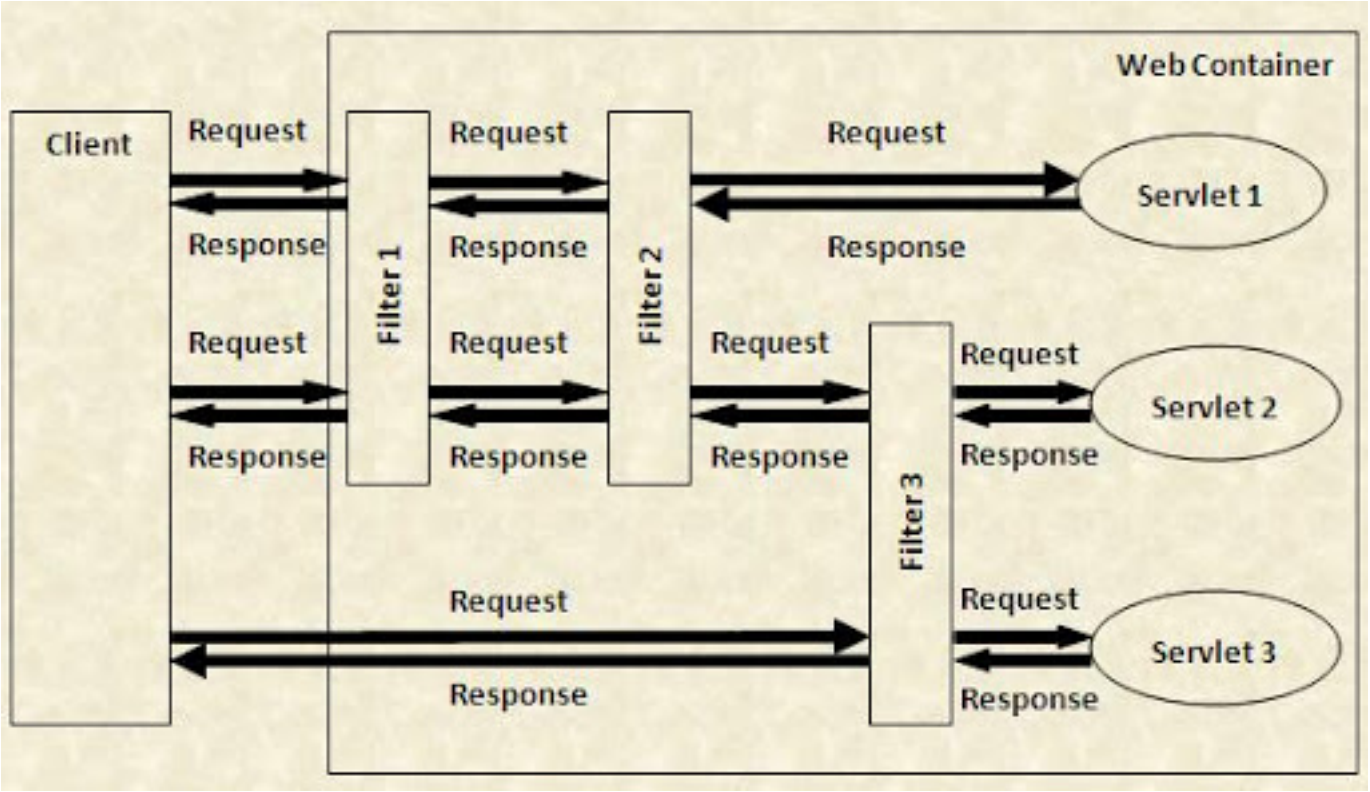
What is a Servlet Filter?

Jakarta Servlet & JSP

Miss Xing

Servlet Filter

- A filter is a reusable piece of code that can transform the content of HTTP requests, responses, and header information.



Servlet Filters can perform many functions

- Authentication and Authorization
- Logging and auditing – Tracking users of a web application
- Image conversion - Scaling maps, and so on
- Data Compression – Making downloads smaller
- Localization – Targeting the request and response to a particular locale
- Encryption, tokenizing, mime-type chaining, caching, etc

Servlet Filter Interface & Life cycle

Jakarta Servlet & JSP

Miss Xing

Servlet Filter Interface

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

    void doFilter(ServletRequest var1, ServletResponse var2,
FilterChain var3) throws IOException, ServletException;

    default void destroy() {
    }
}
```

- init(): called only once, **FilterConfig** is used by container to provide init parameters and servlet context object to the Filter.
- doFilter(): invoked every time by container when apply filter to a resource.
- destroy(): called only once

First Servlet Filter

Jakarta Servlet & JSP

Miss Xing

Configuration - XML

```
<filter>
      <filter-name>RequestLoggingFilter</filter-name> <!-- mandatory -->
      <filter-class>miss.xing.filter.RequestLoggingFilter</filter-class> <!-- mandatory -->
      <init-param> <!-- optional -->
         <param-name>initParamKey</param-name>
         <param-value>initParamValue</param-value>
      </init-param>
   </filter>

   <filter-mapping>
      <filter-name>RequestLoggingFilter</filter-name> <!-- mandatory -->
      <url-pattern>/*</url-pattern> <!-- either url-pattern or servlet-name is mandatory -->
<!--       <servlet-name>LoginServlet</servlet-name>-->
      <dispatcher>REQUEST</dispatcher>
   </filter-mapping>
```

DispatcherType

Jakarta Servlet & JSP

Miss Xing

DispatcherType

- A DispatcherType can be associated with a Java Servlet Filter to limit its scope.

Type	Description
REQUEST	Only when the request comes directly from the client
ASYNC	Only when the asynchronous request comes from the client
FORWARD	Only when the request has been forwarded to a component
INCLUDE	Only when the request is being processed by a component that has been included
ERROR	Only when the request is being processed with the error page mechanism

```
public enum DispatcherType {
    FORWARD,
    INCLUDE,
    REQUEST,
    ASYNC,
    ERROR;

    private DispatcherType() {
    }
}
```

URL Pattern

Jakarta Servlet & JSP

Miss Xing

URL Patterns

- The `url-pattern` element of a `servlet-mapping` or a `filter-mapping` associates a filter or servlet with a set of URLs.
- When a request arrives, the container uses a simple procedure for matching the URL in the request with a `url-pattern` in the `web.xml` file or annotations.
- A URL pattern may contain a subset of US-ASCII characters. Other values must be escaped.

URL Pattern Syntax

- Every character in a pattern must match the corresponding character in the URL path exactly.
- `/*` matches any sequence of characters from that point forward including path `"/"`.
- The pattern `*.extension` matches any file name ending with `extension`.
- No other wildcards are supported, and an asterisk at any other position in the pattern is not a wildcard.

URL Pattern Syntax Examples

- `<url-pattern>/status/*</url-pattern>`

http://example.com/examples/status/synopsis	Matches
http://example.com/examples/status/complete?date=today	Matches
http://example.com/examples/status	Matches
http://example.com/examples/server/status	Does not match

- `<url-pattern>*.map</url-pattern>`

http://example.com/examples/US/Oregon/Portland.map	Matches
http://example.com/examples/US/Washington/Seattle.map	Matches
http://example.com/examples/Paris.France.map	Matches
http://example.com/examples/US/Oregon/Portland.MAP	Does not match, the extension is uppercase
http://example.com/examples/interface/description/mail.mapi	Does not match, the extension is mapi rather than map

Jakarta Filter Order

Jakarta Servlet & JSP

Miss Xing

Servlet Filter Order:

- The order the container uses in building the chain of filters to be applied for a particular request URI is as follows:
  - First, the `<url-pattern>` matching filter mappings in the same order that these elements appear in the deployment descriptor.

```
<filter-mapping>
   <filter-name>AuthenFilter</filter-name>
   <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
   <filter-name>LogFilter</filter-name>
   <url-pattern>/*</url-pattern>
</filter-mapping>
```

  - Next, the `<servlet-name>` matching filter mappings in the same order that these elements appear in the deployment descriptor.

Servlet Filter Order (Cont.):

- If a filter mapping contains both <servlet-name> and <url-pattern>:

```
<filter-mapping>
   <filter-name>Multiple Mappings Filter</filter-name>
   <url-pattern>/foo/*</url-pattern>
   <servlet-name>Servlet1</servlet-name>
   <servlet-name>Servlet2</servlet-name>
   <url-pattern>/bar/*</url-pattern>
</filter-mapping>
```

```
<filter-mapping>
   <filter-name>Multipe Mappings Filter</filter-name>
   <url-pattern>/foo/*</url-pattern>
</filter-mapping>

<filter-mapping>
   <filter-name>Multipe Mappings Filter</filter-name>
   <servlet-name>Servlet1</servlet-name>
</filter-mapping>

<filter-mapping>
   <filter-name>Multipe Mappings Filter</filter-name>
   <servlet-name>Servlet2</servlet-name>
</filter-mapping>

<filter-mapping>
   <filter-name>Multipe Mappings Filter</filter-name>
   <url-pattern>/bar/*</url-pattern>
</filter-mapping>
```

@WebFilter

Jakarta Servlet & JSP

Miss Xing

@WebFilter

Name	Type	Required	Description
filterName	*String*	Optional	Name of the filter.
value or **urlPatterns**	*String[]*	Optional	Specify one or more URL patterns to which the filter applies. Either of attribute can be used, but not both.
dispatcherTypes	*DispatcherType[]*	Optional	Specify types of dispatcher to which the filter applies. Default is `javax.servlet.DispatcherType.REQUEST`
servletNames	*String[]*	Optional	Specify names of servlets to which the filter applies.
displayName	*String*	Optional	Display name of the filter.
description	*String*	Optional	Description of the filter.
asyncSupported	*boolean*	Optional	Specify whether the filter supports asynchronous operation mode. Default is false.
initParams	*WebInitParam[]*	Optional	Specify one or more initialization parameters of the filter. Each parameter is specified by `@WebInitParam` annotation type.
smallIcon	*String*	Optional	Specify name of the small icon of the filter.
largeIcon	*String*	Optional	Specify name of the large icon of the filter.

Jakarta Filter Order with @WebFilter

Jakarta Servlet & JSP

Miss Xing

Servlet Filters' Order when use Annotation

- Cannot not define the filter execution order using `@WebFilter` annotation.
- Take advantage of `web.xml`

```
@WebFilter(filterName="filter1")
public class Filter1 implements Filter {}

@WebFilter(filterName="filter2")
public class Filter2 implements Filter {}
```

```
<filter-mapping>
   <filter-name>filter1</filter-name>
   <url-pattern>/url1/*</url-pattern>
</filter-mapping>
<filter-mapping>
   <filter-name>filter2</filter-name>
   <url-pattern>/url2/*</url-pattern>
</filter-mapping>
```

References

- https://jakarta.ee/specifications/servlet/5.0/jakarta-servlet-spec-5.0.html#what-is-a-filter
- https://www.journaldev.com/1933/java-servlet-filter-example-tutorial#servlet-filter
- https://docs.oracle.com/cd/E19798-01/821-1841/bnagf/index.html
- https://www.codejava.net/java-ee/servlet/webfilter-annotation-examples
- https://stackoverflow.com/questions/6560969/how-to-define-servlet-filter-order-of-execution-using-annotations-in-war
- https://docs.roguewave.com/hydraexpress/3.5.0/html/rwsfservletug/4-3.html