

Filters

Why Filters ?

1. Provides re-usability.

Meaning --- They provide the ability to encapsulate recurring tasks(=cross cutting concerns) in reusable units.

They provide clear cut separation between B.L & cross cutting concerns.

2. Can dynamically intercept req/resp to dyn or static content

What is Filter?

Dynamic web component just like servlet or

JSP. Resides within web-appln.(WC)

Filter life-cycle managed by WC

It performs filtering tasks on either the request to a resource (a servlet,JSP or static content), or on the response from a resource, or both.

It can dynamically intercepts requests and responses .

Usage of Filters

1. Authentication Filters

2. Logging Filters

3. Image conversion Filters

4. Data compression Filters

5. Encryption Filters

6. Session Check filter

How to create Filter Component?

1. Create Java class imple. javax.servlet.Filter i/f

2. Implements 3 life-cycle methods

2.1 public void init(FilterConfig filterConfig)
throws ServletException

Above called by WC --- only once during filter creation & initialization.(@appln start up time)

2.2 void doFilter(ServletRequest request,ServletResponse response,FilterChain chain) throws IOException, ServletException

Invoked by WC -- per every rq & resp processing time.

Here u can do pre-processing of req, then invoke chain.doFilter -- to invoke next component of filter chain --- finally it invokes service method of JSP/Servlet --- on its ret path visits filter chain in opposite manner & finally renders response to clnt browser.

2.3 public void destroy() ---invoked by WC at the end of filter life cycle.

Triggers --- server shut down/re-deploy/un deploy

How to deploy a Filter component ?

1. Annotation -- @WebFilter (class level annotation)

OR

2. XML tags (in web.xml)

```
<filter>
  <filter-name>abc</...>
  <filter-class>filters.AuthenticatioFilter</...>
  <init-param>
    <param-name>nm1</...>
    <param-value>val1</...>
  </...>
</filter>
<filter-mapping>
  <filter-name>abc</...>
  <url-pattern>/*</...>
</filter-mapping>
```

Detailed Description ---

Filters typically do not themselves create responses, but instead provide universal functions that can be "attached" to any type of servlet or JSP page.

Filters are important for a number of reasons. First, they provide the ability to encapsulate recurring tasks in reusable units. Organized developers are constantly on the lookout for ways to modularize their code. Modular code is more manageable and documentable, is easier to debug, and if done well, can be reused in another setting.

Second, filters can be used to transform the response from a servlet or a JSP page. A common task for the web application is to format data sent back to the client. Increasingly the clients require formats (for example, WML) other than just HTML. To accommodate these clients, there is usually a strong component of transformation or filtering in a fully featured web application. Many servlet and JSP containers have introduced proprietary filter mechanisms, resulting in a gain for the developer that deploys on that container, but reducing the reusability of such code. With the introduction of filters as part of the Java Servlet specification, developers now have the opportunity to write reusable transformation components that are portable across containers.