**Servlet Filter**

A **Filter** is an object that is invoked at the pre-processing and post-processing of a request.

It is mainly used to perform filtering tasks such as conversion, logging, compression, encryption and decryption, input validation etc.

The Servlet filter is plug-gable, i.e. its entry is defined in the web.xml file, if we remove the entry of filter from the web.xml file, filter will be removed automatically and we don't need to change the Servlet

**Usage of Filter**

* Recording all detail about incoming requests
* Logs the IP addresses of the computers from which the requests originate
* Conversion
* Data compression
* Encryption and decryption
* Input validation etc.

**Advantage of Filter**

* Filter is plug-gable.
* One filter don't have dependency onto another resource.
* Less Maintenance

**Filter API**

Like Servlet filter have its own API. There are three interfaces in Servlet API to implement filter concept in a java web application. The javax.servlet package contains the three interfaces of Filter API.

* Filter
* FilterChain
* FilterConfig

**1) Filter interface**

For creating any filter, we must implement the Filter interface. Filter interface have three life cycle methods.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void init(FilterConfig config) | init() method is invoked only once. It is used to initialize the filter. |
| public void doFilter(HttpServletRequest request,HttpServletResponse response, FilterChain chain) | doFilter() method is invoked every time when user request to any resource, to which the filter is mapped.It is used to perform filtering tasks. |
| public void destroy() | This is invoked only once when filter is taken out of the service. |

**2) FilterChain interface**

whenever more than one filter are used in Servlet then it called as Filter Chain. The object of FilterChain is responsible to invoke the next filter or resource in the chain.This object is passed in the doFilter method of Filter interface. The FilterChain interface contains only one method:

* **public void doFilter(HttpServletRequest request, HttpServletResponse response):** it passes the control to the next filter or resource.
* **How to apply a Filter to Servlet**
* We can define filter same as Servlet. Let's see the elements of filter and filter-mapping.
* **Syntax**
* **<web-app>**
* **<filter>**
* **<filter-name>**...**</filter-name>**
* **<filter-class>**...**</filter-class>**
* **</filter>**
* **<filter-mapping>**
* **<filter-name>**...**</filter-name>**
* **<url-pattern>**...**</url-pattern>**
* **</filter-mapping>**
* **</web-app>**

**FilterConfig**

An object of FilterConfig is created by the web container. This object can be used to get the configuration information from the web.xml file.

**Methods of FilterConfig interface**

There are following 4 methods in the FilterConfig interface.

* **public void init(FilterConfig config):** init() method is invoked only once it is used to initialize the filter.
* **public String getInitParameter(String parameterName):** Returns the parameter value for the specified parameter name.
* **public java.util.Enumeration getInitParameterNames():** Returns an enumeration containing all the parameter names.
* **public ServletContext getServletContext():** Returns the ServletContext object.

**Example of FilterConfig**

In this example, if you change the param-value to no, request will be forwarded to the servlet otherwise filter will create the response with the message: this page is underprocessing. Let's see the simple example of FilterConfig. Here, we have created 4 files:

* index.html
* MyFilter.java
* HelloServlet.java
* web.xml

**Servlet Filter Chaining**

whenever more than one filter are used in Servlet then it called as **Filter Chain**. The object of FilterChain is responsible to invoke the next filter or resource in the chain.This object is passed in the doFilter method of Filter interface. The FilterChain interface contains only one method:

* **public void doFilter(HttpServletRequest request, HttpServletResponse response):** it passes the control to the next filter or resource.