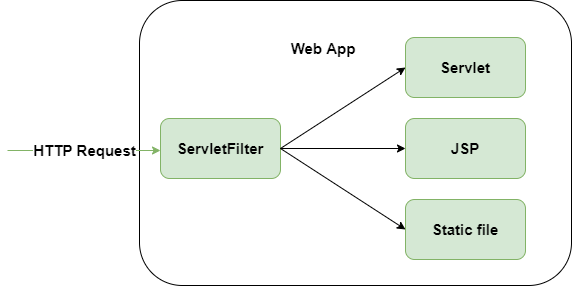
**Servlet – Filter**

A filter is an object that is used throughout the pre-and post-processing stages of a request. Filters are mostly used for filtering tasks such as server-side logging, authentication, and authorization, input validation, and so on.



The servlet is pluggable, which means that the entry is specified in the web.xml file. If the entry is deleted from the web.xml file, the filter is immediately deactivated. It must implement javax.servlet in order to generate a filter. Because servlets are constructed in the highly portable Java language and adhere to a common framework, they are very portable. As a result, it enables the creation of advanced server extensions in server and operating systems that are independent of one other.

**Needs of Servlet Filters:**

* Logging on the server.
* Request parameter is logged to log files.
* Authentication and authorization on the server.
* Compressing and Decompressing
* Encryption and decryption are two different things.
* Validation on the server.

**Benefits of Servlet Filters:**

* It can be plugged in.
* The filter is not reliant on a third-party resource.
* It requires little maintenance.

**Servlet Filter Methods:**

The filter interface consists of 3 life cycle methods.

| **Methods** | **Description** |
| --- | --- |
| public void init(FilterConfig config) throws ServletException | This invokes the web container to indicate to a filter that is being placed into service. It takes one parameter, i.e. FilterConfig type or FilterConfig object. |
| public void doFilters(ServletRequest request,ServletResponse response,FileterChain chain) throws ServletException,IOException | The doFilter() function is called whenever a user requests a resource that is mapped to the filter. It’s used for filtering purposes. |
| public void destroy() | When the filter is removed from the service, this function is only called once. |

**Example**

In this example, we’ll make a webpage **index.html** with a link that says “click here.” Click the link to invoke the servlet “**ServletFilter.java**” But before this Servlet is executed, the filter “**MyFilter.java**” associated with it will be executed which is specified in the deployment descriptor.

**index.html**

* HTML

|  |
| --- |
| <**html**>      <**head**>          <**title**>HttpSession Event Listeners</**title**>      </**head**>      <**body**>          <**a** href="ServletFilter">click here</**a**>      </**body**>  </**html**> |

**ServletFilter.java**

* Java

|  |
| --- |
| **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.\*;  **public** **class** ServletFilter **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request,                        HttpServletResponse response)  **throws** ServletException, IOException      {          response.setContentType("text/html");          PrintWriter out = response.getWriter();          out.print(              "<br>welcome to servlet filter example<br>");      }  } |

**MyFilter.java**

* Java

|  |
| --- |
| **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** javax.servlet.\*;  **public** **class** MyFilter **implements** Filter {      FilterConfig config;  **public** **void** init(FilterConfig config)  **throws** ServletException      {  **this**.config = config;      }  **public** **void** doFilter(ServletRequest req,                           ServletResponse resp,                           FilterChain chain)  **throws** IOException, ServletException      {          PrintWriter out = resp.getWriter();          String s = config.getInitParameter("construction");  **if** (s.equals("yes")) {              out.print("This page is under construction");          }  **else** {              // sends request to next resource              chain.doFilter(req, resp);          }      }  **public** **void** destroy() {}  } |

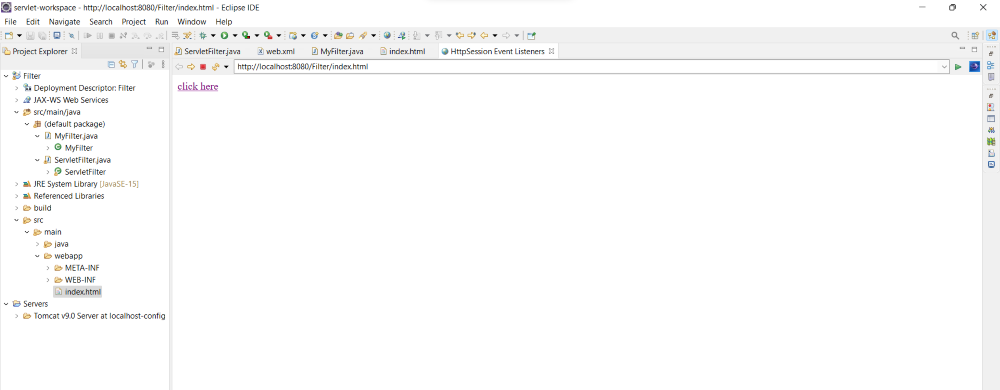
**web.xml**

* XML

|  |
| --- |
| <?**xml** version="1.0" encoding="UTF-8"?>  <**web-app**      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"      xmlns="<http://xmlns.jcp.org/xml/ns/javaee>"   xsi:schemaLocation="[http://xmlns.jcp.org/xml/ns/javaee](http://xmlns.jcp.org/xml/ns/javaee%C2%A0)  <http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd>"   id="WebApp\_ID" version="4.0">      <**servlet**>          <**servlet-name**>ServletFilter</**servlet-name**>          <**servlet-class**>ServletFilter</**servlet-class**>      </**servlet**>      <**servlet-mapping**>          <**servlet-name**>ServletFilter</**servlet-name**>          <**url-pattern**>/ServletFilter</**url-pattern**>      </**servlet-mapping**>      <**filter**>          <**filter-name**>f1</**filter-name**>          <**filter-class**>MyFilter</**filter-class**>          <**init-param**>              <**param-name**>construction</**param-name**>              <**param-value**>no</**param-value**>          </**init-param**>      </**filter**>      <**filter-mapping**>          <**filter-name**>f1</**filter-name**>          <**url-pattern**>/servlet1</**url-pattern**>      </**filter-mapping**>  </**web-app**> |

**Output:**

Click on “Click Here”



After Clicking the following link you will get this message

