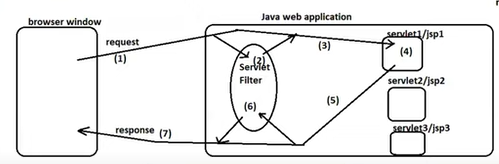
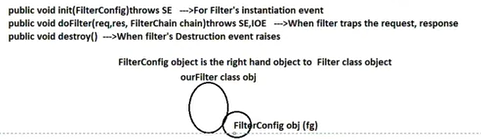
* Servlet Filters are part of servlet api. Servlet Filters are special web comps of java web application that are capable trapping either all request-responses or multiple requests-responses to apply common, additional and optional pre-request processing and post-response generation logic without disturbing other web comps of web application.



* Servlet filter or filters can be mapped 1 or more web comps of web application.
* Every Filter must be cfg and linked with other web comps(s) either using <filter>,<filter-mapping> tags(xml driven cfgs) or using @webFilter annotation(annotation driven cfgs).
* Every Filter must implement javax.servlet.Filter(I) or Jakarta.servlet.Filter(I) directly or indirectly.

The java8 interface contains 2 default methods and 1 abstract method as life cycle methods(Filter).



* 3 Types of Filters

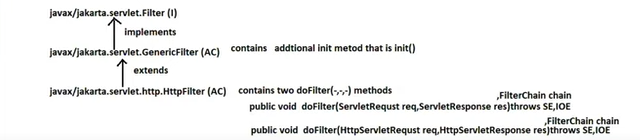
a. request Filters.

b. response Filters.

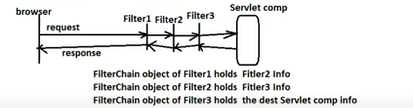
c. request-response filters.

2. Servlet Filter API:

===============



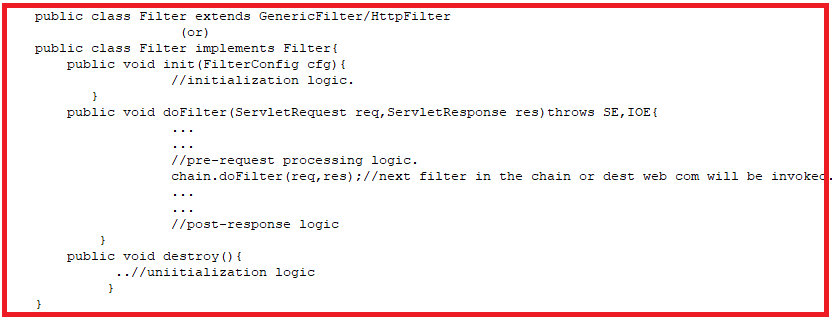
* Filter chain object holds Dest web comps info if the filter is last filter in the chain otherwise Filterchain object of each filter comp holds next filter info/details.



* If multiple filter are trapping the request in chain.. in which order they are trapping request .. exactly in reverse order they trap the response.

3.Sample Filter Code

===============



In three ways, we can create filters.

a. using GenericFilter

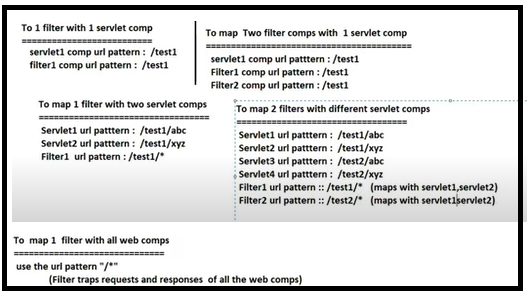
b. using HttpFilter

c. Filter interface.

The best approach is using HttpFilter.

4. Mapping Between Servlet and Filter

===========================



* The servlet container creates our filter class object either during server startup during the deployments of web application(no <load-on-startup>) is required.
* The servlet container destroys our filter class object when the web application stopped/undeployed or reloaded or server crashed/stopped.

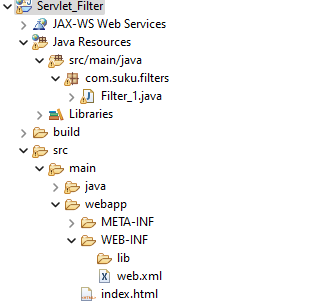
5.Examples:

=========

1.Developing servletFilter that allows request to all the web comps of webapplication between 9am to 5pm.

Step1: keep any web application ready.

Step2: add Filter comp to web application having url pattern”/\*” as show below.



Index.html

========

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h2 align='center'>Filter worked</h2>

</body>

</html>

Filter1.java

========

**package** com.suku.filters;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.time.LocalDateTime;

**import** jakarta.servlet.FilterChain;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebFilter;

**import** jakarta.servlet.http.HttpFilter;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

@WebFilter("/\*")

**public** **class** Filter\_1 **extends** HttpFilter {

**public** **void** doFilter(HttpServletRequest req,HttpServletResponse res,FilterChain fc) **throws** IOException, ServletException {

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

LocalDateTime ld=LocalDateTime.*now*();

**int** hour=ld.getHour();

**if**(hour>=9 && hour<=17) {

fc.doFilter(req, res);

}

**else**

{

pw.write("<h2> Web site operated after 9AM and before 5PM.</h2>");

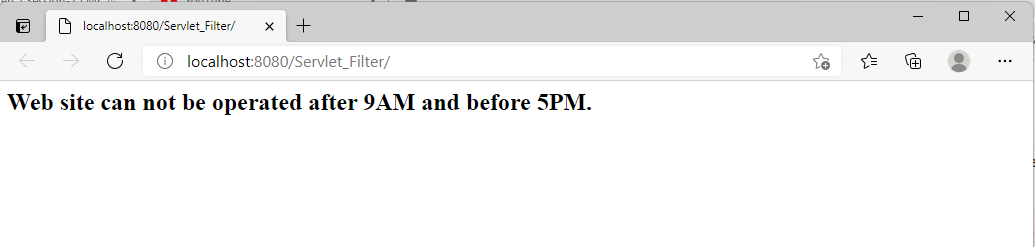
}

}

}

Step3: Run The Application.

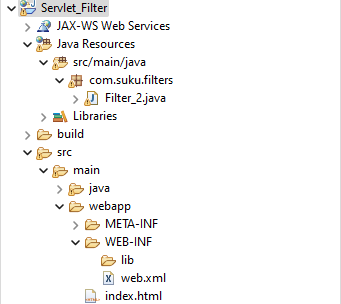
Output:- Now in computer time is 20:00



2.Developint ServletFilter comp that allows requests from certain browser s/w (chrome) only.

Step1: keep the web application ready.

Step2: Develop the servletFilter comp having url pattern “/\*” as below.



Index.html

========

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h2 align='center'>Filter worked</h2>

</body>

</html>

Filter2.java

========

**package** com.suku.filters;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** jakarta.servlet.FilterChain;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebFilter;

**import** jakarta.servlet.http.HttpFilter;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

@WebFilter("/\*")

**public** **class** Filter\_2 **extends** HttpFilter {

**public** **void** doFilter(HttpServletRequest req,HttpServletResponse res,FilterChain fc) **throws** IOException, ServletException {

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

String bname=req.getHeader("user-agent");

**if**(bname.contains("chrome")) {

fc.doFilter(req, res);

}

**else**

{

pw.write("<h2> Web site should work in only chrome browser</h2>");

**return**;

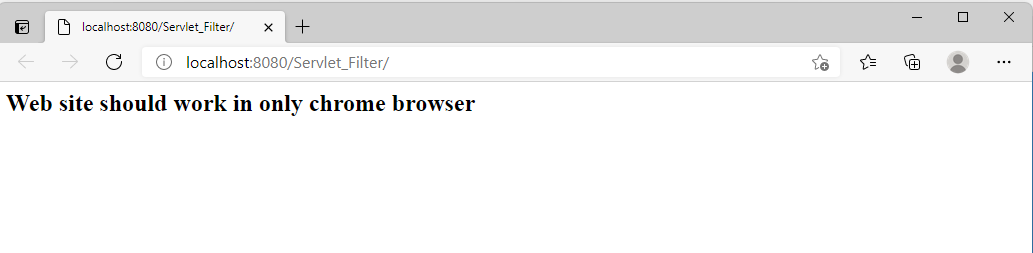
}

}

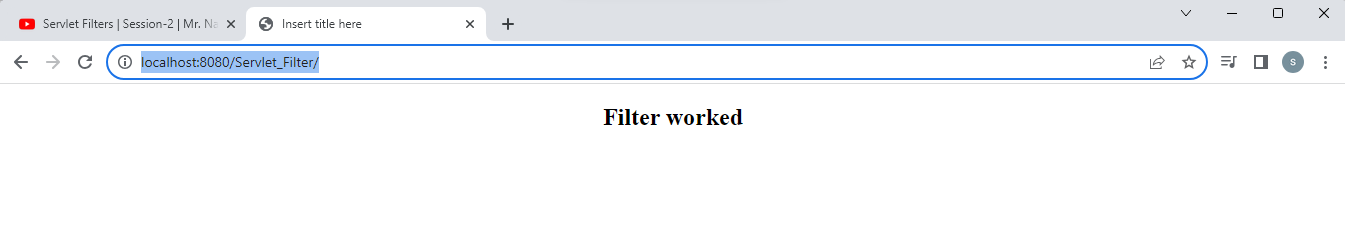
}

Step3: Test the application.

Output:- Opened in Edge browser.



Run2: Opened in chrome browser



Note:- if multiple filters are cfg on servlet comp using xml cfgs then in which order the filter comps are cfg in web.xml file the request will be trapped and the response will be trapped in the reverse order.

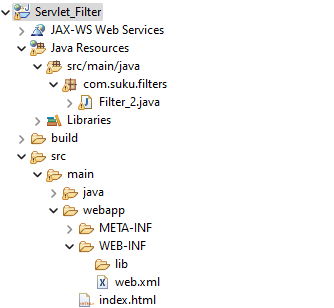
If the same multiple filters are cfg using annotations then based on the alphabetic order of filter class names the request trapping and respone trapping order will be decided.

Example :3 Develop Filter comp that evaluates the performance of each web comp by getting its request processing time.

[This filter comp must be developed as request-response filter]

Step1: Ready the any web application.

Step2: Develop the Filter comp.



Index.html

========

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h2 align='center'>Filter worked</h2>

</body>

</html>

Filter2.java

========

**package** com.suku.filters;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** jakarta.servlet.FilterChain;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebFilter;

**import** jakarta.servlet.http.HttpFilter;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

@WebFilter("/\*")

**public** **class** Filter\_2 **extends** HttpFilter {

**public** **void** doFilter(HttpServletRequest req,HttpServletResponse res,FilterChain fc) **throws** IOException, ServletException {

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

Long sTime,eTime;

sTime=System.*currentTimeMillis*();

fc.doFilter(req, res);

eTime=System.*currentTimeMillis*();

System.***out***.println("Execution Completed in:"+ (eTime-sTime));

pw.close();

}

}

Output:

