1 Filtering is a standard feature of

Ans : all Servlet 2.5–compliant containers.

2 when does filter introduce int servelet technology ?

Ans : Servlet 2.3

3 Some popular uses for filters ?

Ans: include authentication, auditing, compression, encryption, and on-the-fly format transformation, to name but a few.

4 which Servlet further enhances container support of filters by providing filtering for dispatched requests?

Ans : Servlet 2.5

5 what is filter ?

Ans : Filters can intercept request header information before it reaches the resource in the processing pipeline and can therefore be used to create customized authentication schemes.

6 what can filter do?

a) intercept request header information before it reaches the resource
b) data transformation.
c) preempt the serving of a particular resource

7 filter can intercept http request header information ?true or false

8 filter can useful in data transformation, explain with an example

Ans : For example, a filter can first detect whether a user agent (browser) supports compressed data
streams. If the browser can handle the compression, the filter can then compress the response from a resource on the fly

9 filter can preerpt the reveing of the particuler resource ?

Ans : . One example could be a time-sensitive filter that blocks access to certain resources (such as an Internet proxy server) outside certain set hours.

10 what type of resource does a server serve?Exoain with an graph (see 401)

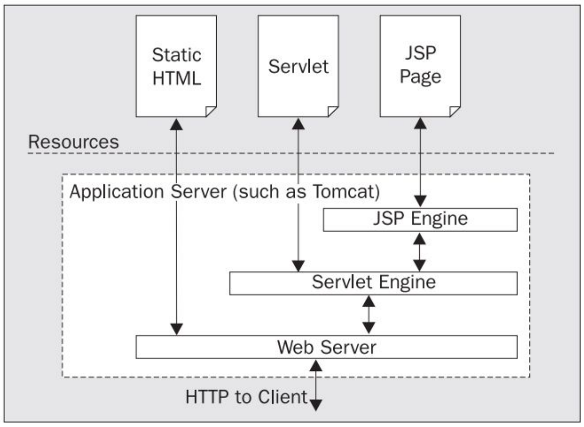

Figure 10-1. *A client can request various types of resources from a web server. The web server routes the request and returns the appropriate resource to the client.*

11 filter are called by the application server both prior to and subrequest resources procession to the server explain with graph?

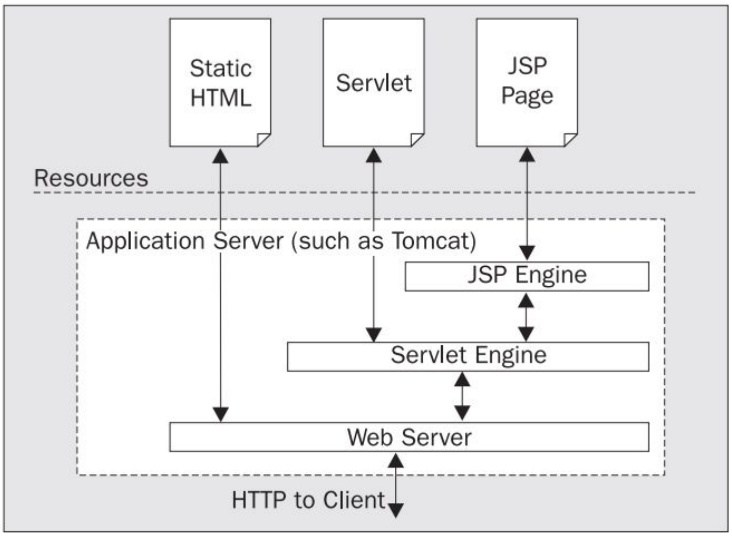

Figure 10-1. *A client can request various types of resources from a web server. The web server routes the request and returns the appropriate resource to the client.*

12 filter interface has _ main methods explain each methods (404)

a) public void init(FilterConfig config)throws ServletException:

b) public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws IOException, ServletException

c) public void destroy():

13 define filter definition and filter mapping (404)

- Filter definition: Tells the container the textual name associated with the filter
- Filter mapping: Tells the container which resources the filter will be applied to

14 draw a diagram of filter life cycle?

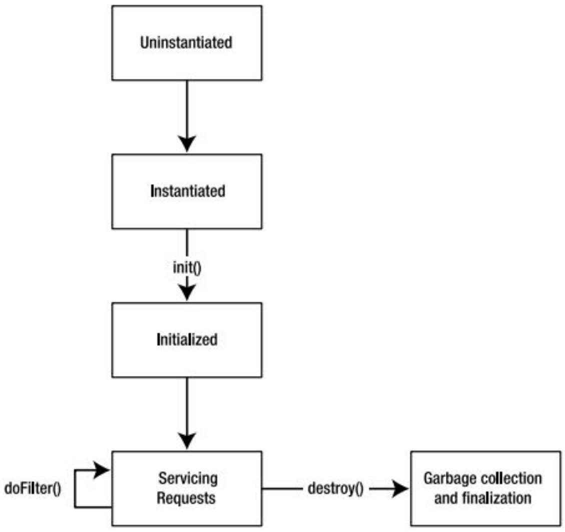

Figure 10-4. *The life cycle of a filter is similar to the life cycle of a servlet. The filter is created and initialized, it processes requests, and it is destroyed when no longer needed.*

15 write 3 methods of filter config interface and explain the functions?

Ans :

- **public String getFileName()**: You can use this method to obtain the textual name of the filter, as defined in the web.xml deployment descriptor.
- **public String getInitParameter(String paramName):** The getInitParameter() method obtains the string value of a specific initialization parameter by name. Returns null if not found.
- **public Enumeration getInitParameterNames(): T**his method obtains a java.util. Enumeration consisting of all the names of the initialization parameters for this instance. These parameters are specified in the web.xml deployment descriptor within the <filter> definitions. Returns an empty enumeration if no parameter is set.
- **public ServletContext getServletContext():** This method obtains the ServletContext that the filter is executing within. This context is typically specified in the server.xml file of the server.

16 write down filter definition like filter-name,class,param(407)

- **<filter-name>**: Textual name to associate with the filter. Used in filter mapping. This is a mandatory element.
- **<filter-class>:** The actual class that implements a filter. Should be a fully qualified class name with a package prefix. This is a mandatory element.
- **<init-param>:** Specifies the initial parameters to supply to this instance of the filter. Contains <param-name> and <param-value> subelements, specifying the name and value of the parameter, respectively. Note that <init-param> is an optional child element of <filter>, which can also appear multiple times—once for each initialization parameter for the filter.

17 define filter mapping with an example

```
<filter>
<filter-name>Audit Filter</filter-name>
<filter-class>filters.AuditFilter</filter-class>
</filter>

<filter-mapping>
<filter-name>Visual Audit Filter</filter-name>
<servlet-name>mylocate</servlet-name>
</filter-mapping>

<servlet>
<servlet-name>mylocate</servlet-name>
<servlet-class>FindProd</servlet-class>
</servlet>
```

18 define /*, /servlet/*,/jsp/*.jsp,/dept/accounting/*

/* >= Everything that is served by this web application, including static pages, servlets, and JSP pages

/servlet/* >= All servlets (assuming all servlets are mapped under the /servlet path)

/jsp/*.jsp >= All JSP pages located on the /jsp path

/dept/accounting/* >= All resources in the accounting department branch of the web application

19 draw a graph related to incoming and outgoing request to access

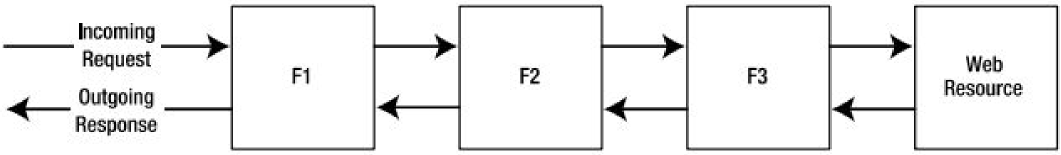

Figure 10-5. *An incoming request is processed by zero or more filters, and the response from the web resource can be processed by the same set of filters.*