Java Servlets - Filters

Presented by

Sagar
Java Consultant

Servlet Filters

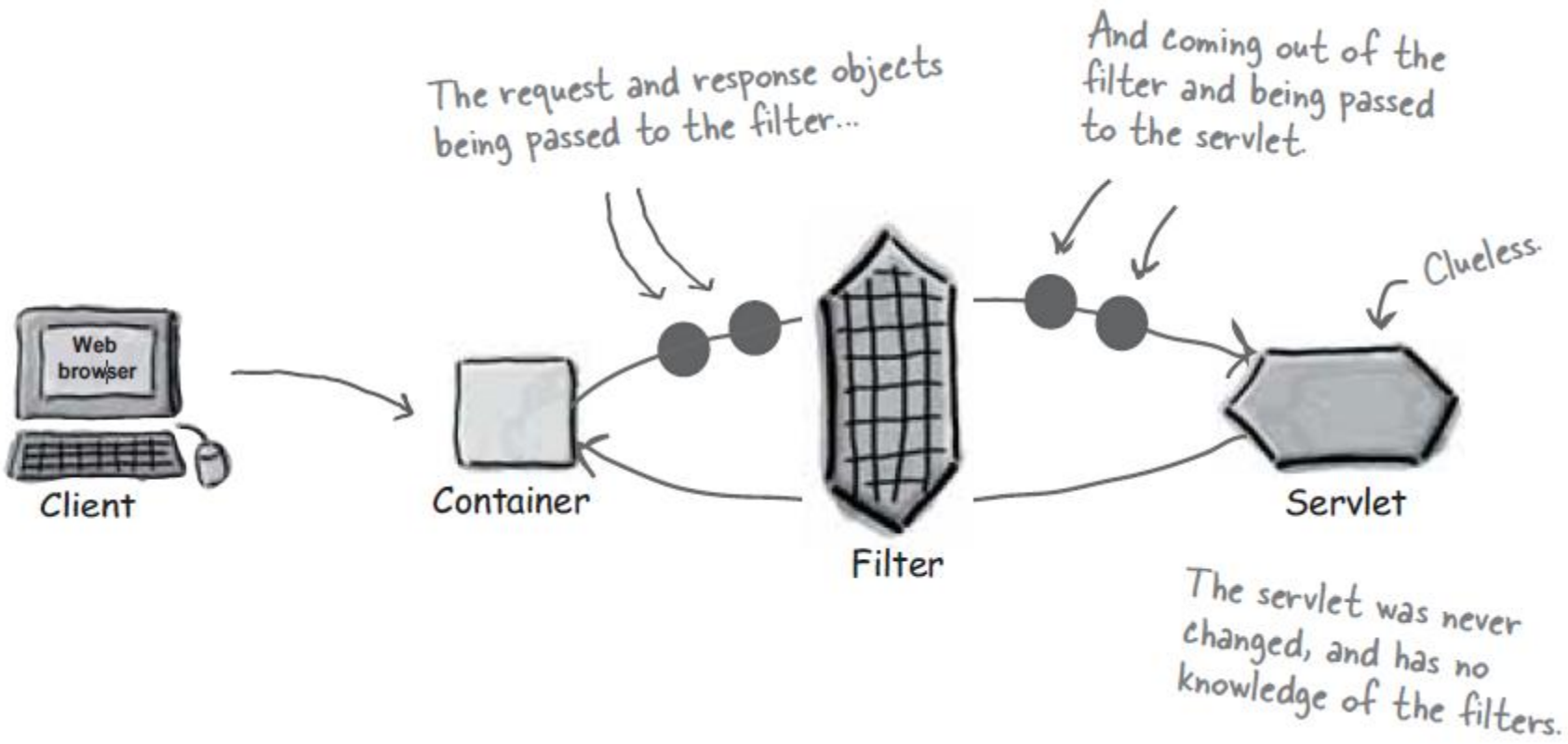
Filters let us to intercept the request.

Dynamic components, usually associated with servlets.

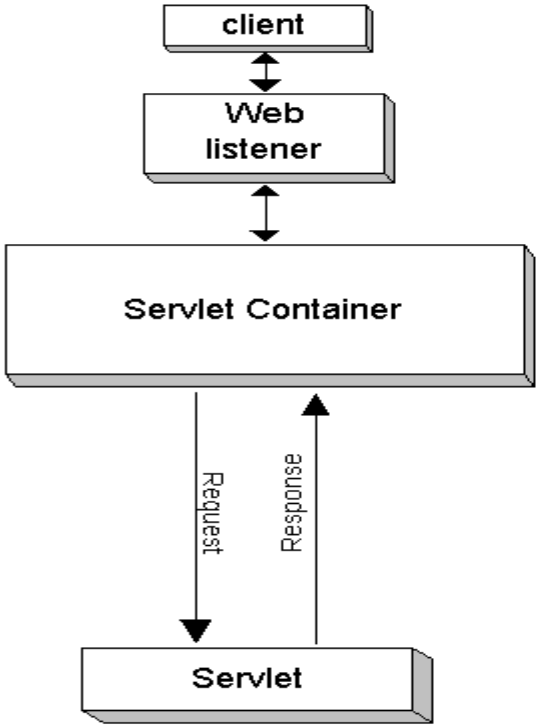
They intercepts the requests before sending to a servlet.

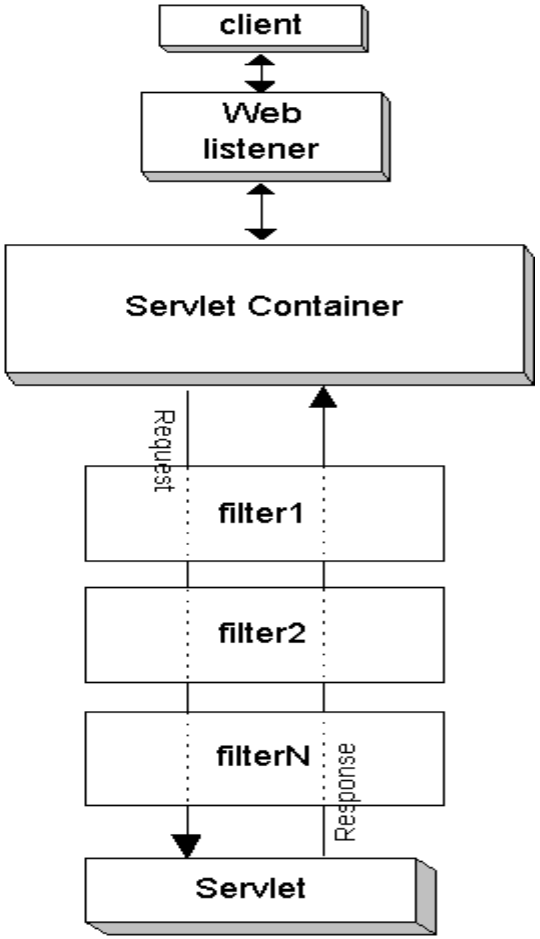
The container decides when to invoke the filters , based on the declarations In the Deployment Descriptor.

Servlets Filters



Servlets Filters





**Example

Filter life cycle

The Container manages their lifecycle

Just like servlets, filters have a lifecycle. Like servlets, they have **init()** and **destroy()** methods. Similar to a servlet's **doGet()/doPost()** method, filters have a **doFilter()** method.

Example - Filter

```
public class HelloWorldFilter implements Filter{

   public void init(FilterConfig fltrCnfg) throws ServletException {}

   public void doFilter(ServletRequest req, ServletResponse resp,
         FilterChain chain) throws IOException, ServletException {

      PrintWriter out=resp.getWriter();
   //Before Servlet
      out.print("I Filtered the request...!");
   // Navigates to servlet
      chain.doFilter(req, resp);//sends request to next resource, i.e, servlet
   //After servlet
      out.print("That's the power of filters..!");
   }
```

Example - Servlet

```
public class HelloWorldServlet 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 the world of Servlet Filters..!<br>");

    }

}
```

** Absolutely no clue about the filter

Deployment Descriptor

```
<filter>
    <filter-name>HelloWorldFilter</filter-name>
    <filter-class>HelloWorldFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>HelloWorldFilter</filter-name>
  <url-pattern>/HelloWorldServlet</url-pattern>
</filter-mapping>
```