Tech Mahindra

Filters and Listeners

TLS

Rewards and Recognition

2012 ASTD BEST AWARD WINNER

Training TOP 125 Winner - 2013

Tata Institute of Social Sciences CLO Awards India Best Virtual Learning Award

Objectives

At the end of this session, you will be able to understand,

- Need and functionality of Filters
- Need and functionality of Listeners

Agenda

Following points to be covered in the session,

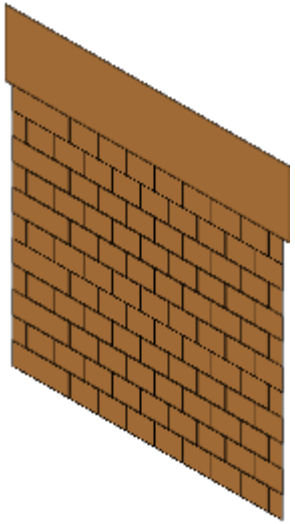
- Filters and their functionality
- Creating user defined filters
- Types of Listeners
- Creating and implementing Listeners

Filters

TLS

Rewards and Recognition

Scenario of Security gate in IT company..



Visitor Security Gate / Entrance Company

All the visitors of the company have to pass through security gate where checking is done during entry and exit both. This is done to ensure security is achieved. Security threats are stopped at entry itself to avoid unnecessary challenges.

Filter Introduction

- Servlet Filters are Java classes that can be used in Servlet Programming for the following purposes:

1. To intercept requests from a client before they access a resource at back end.
2. To manipulate responses from server before they are sent back to the client.

- **Usage of Filter**

1. Record all incoming requests
2. logs the IP addresses of the computers from which the requests originate
3. conversion
4. data compression
5. encryption and decryption
6. input validation
7. logging the application

Filter Introduction (Cont..)

- **Advantage of Filter**
  1. Filter is pluggable.
  2. One filter don't have dependency onto another resource.
  3. Less Maintenance

- **Filter API** : Like servlet filter have its own API. The javax.servlet package contains the three interfaces of Filter API.
  1. Filter
  2. FilterChain
  3. FilterConfig

Steps for creating user defined filter with E.g.

1)Creating UI to access Filter

index.html

```
<a href="servlet1">click here</a>
```

2)Creating a Filter class

MyFilter.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
public class MyFilter implements Filter{
public void init(FilterConfig arg0) throws ServletException {}
public void doFilter(ServletRequest req, ServletResponse resp,
    FilterChain chain) throws IOException, ServletException {
    PrintWriter out=resp.getWriter();
    out.print("filter is invoked before");
    chain.doFilter(req, resp);//sends request to next resource
    out.print("filter is invoked after");
    }
    public void destroy() {}
}
```

Steps for creating user defined filter

3)Creating servlet

HelloServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class HelloServlet 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<br>");
        }
}
```

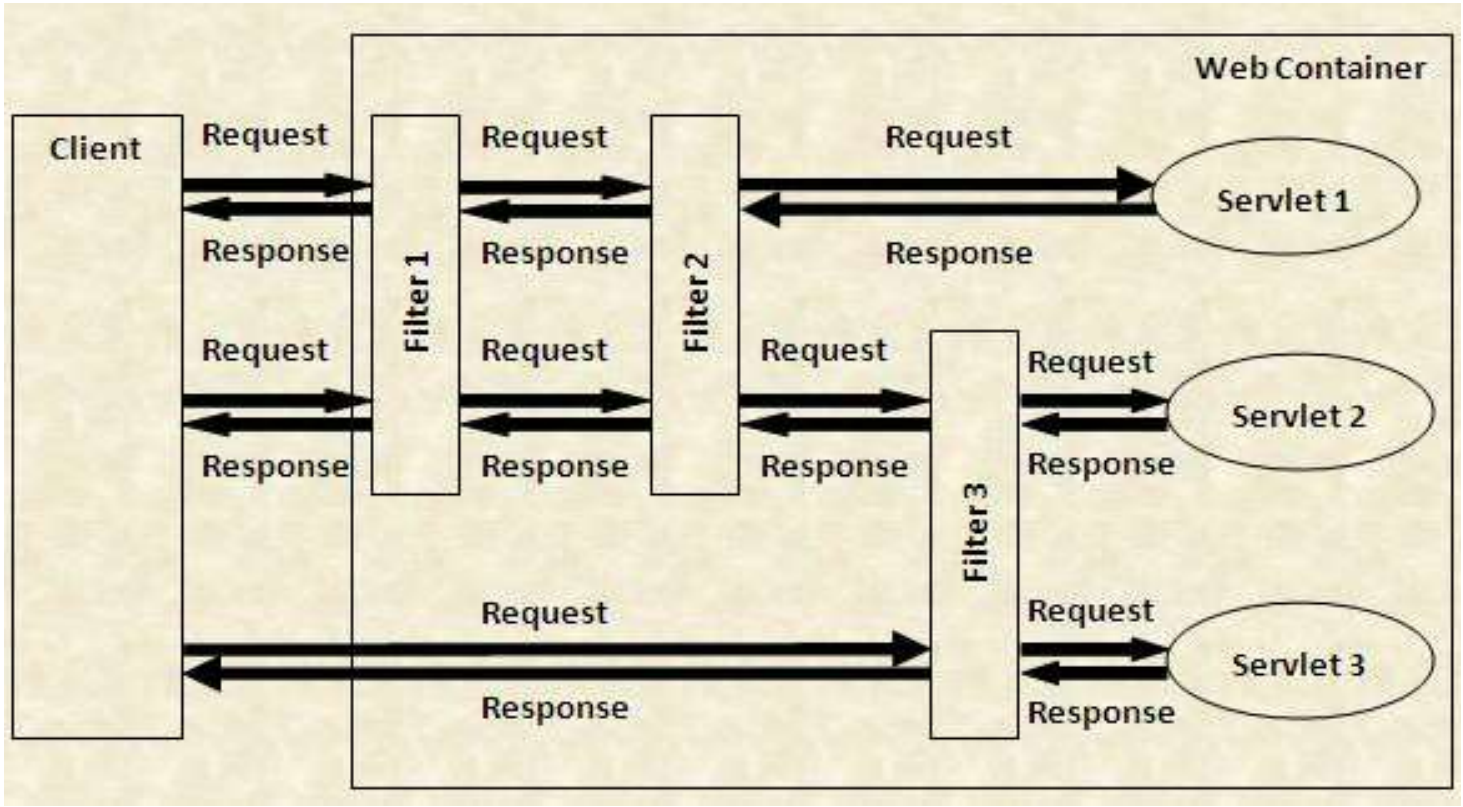
Steps for creating user defined filter

4)Configuring filter in web.xml

```
<web-app>
<servlet>
        <servlet-name>s1</servlet-name>
        <servlet-class>HelloServlet</servlet-class>
</servlet>
        <servlet-mapping>
        <servlet-name>s1</servlet-name>
        <url-pattern>/servlet1</url-pattern>
</servlet-mapping>
<filter>
        <filter-name>f1</filter-name>
        <filter-class>MyFilter</filter-class>
</filter>
<filter-mapping>
        <filter-name>f1</filter-name>
        <url-pattern>/servlet1</url-pattern>
</filter-mapping>
</web-app>
```

Filter (Cont..)

- Following diagram explains possible combinations in filter :



Filters (Cont..)

- **Best Practices** : Use Filters is recommanded for writing pre-processing and post-processing logic for servlets. E.g.

✓ Preprocessing – User authentication before control is passed to servlet. In case authentication fails, servlet execution time can be saved.

Or checking session for a user and allowing user access only if new session is created.

✓ Post Processing – Log out code implementation after the user clicks on LOG OUT option in website. Also session can be terminated in post processing so that user can not use that session again by clicking on back button.

[**Note** : Java Framework named "**Struts**" is based on filters to automate memory management ,exception handling tasks etc. For achieving this truts uses modified version of Filters called Interceptors.]

Listeners

TLS

Rewards and Recognition

Listeners Introduction

- In a web application various tasks need to be completed. E.g.
  1. counting total and current logged-in users,
  2. creating tables of the database at time of deploying the project
  3. creating database connection object etc.

These tasks must be performed based upon certain event occurrence. JEE gives us the way to identify the events and perform some logic after they are triggered through various event classes and listener interfaces available in Servlet APIs. This technique of handling events with code is called as **“EVENT HANDLING”.**

- In servlet all related APIs are available inside javax.servlet and javax.servlet.http package

Listener APIs

- **Event classes** : The event classes are as follows:

ServletRequestEvent

ServletRequestAttributeEvent

ServletContextEvent

ServletContextAttributeEvent

HttpSessiontEvent

HttpSessionBindingEvent

Listener APIs

- **Event interfaces** : The event interfaces are as follows:

ServletRequestListener

ServletContextListener

ServletRequestAttributeListener

ServletContextAttributeListener

HttpSessionListener

HttpSessionAttributeListener

HttpSessionBindingListener

HttpSessionActivationListener

Listeners working in servlets

- Application lifecycle is controlled by container. To map task execution with application component state, listeners are used.
- Container listens to the events occurring and checks if a corresponding listener is configured for application. During this it also check for corresponding event handling method and executes it.
- Eg . In Shopping web site daily offers and discount rates should be assigned as soon as application starts for the day. This can be done by listening to application life cycle events.
- Listener is configured in web.xml

```
<web-app>
 <listener-class>package.classname</listener-class>
</web-app>
```

- Based upon requirement Java class should be created which implements any of the listens mentioned earlier. Corresponding method must be implemented in java class to handle respective events.

Summary

In this session, we have covered,

- Filters
- Listeners

Thank You

Disclaimer

Tech Mahindra Limited, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.