



Servlet and JSP Filters

Core Servlets & JSP book: www.coreservlets.com
 More Servlets & JSP book: www.moreservlets.com
 Servlet and JSP Training Courses: courses.coreservlets.com

Slides © Marty Hall, <http://www.moreservlets.com>, book © Sun Microsystems Press

Agenda

- Filter basics
- Accessing the servlet context
- Using initialization parameters
- Blocking responses
- Modifying responses
- Online info
 - The chapter on filters from *More Servlets and JavaServer Pages* is available online in PDF at the Sun Java Developer's Connection. See http://developer.java.sun.com/developer/Books/javaserverpages/servlets_javaserver/.

www.moreservlets.com

Filters: Overview

- Associated with any number of servlets or JSP pages
- Examine request coming into servlets or JSP pages, then:
 - Invoke the resource (i.e., the servlet or JSP page) in the normal manner.
 - Invoke the resource with modified request information.
 - Invoke the resource but modify the response before sending it to the client.
 - Prevent the resource from being invoked and instead redirect to a different resource, return a particular status code, or generate replacement output.
- Available only in servlets 2.3

www.moreservlets.com

Advantages of Filters

- Encapsulate common behavior.
 - Have 30 different servlets or JSP pages that need to compress their content to decrease download time? Make 1 compression filter and apply it to all 30 resources.
- Separate high-level access decisions from presentation code.
 - Want to block access from certain sites without modifying the individual pages to which these access restrictions apply? Create an access restriction filter and apply it to as many pages as you like.
- Apply wholesale changes to many different resources.
 - Have a bunch of existing resources that should remain unchanged except that the company name should be changed? Make a string replacement filter and apply it wherever appropriate.

www.moreservlets.com

Steps to Creating Filters

- Create class that implements Filter interface.
 - Methods: doFilter, init, destroy
- Put filtering behavior in doFilter.
 - Args: ServletRequest, ServletResponse, FilterChain
- Call doFilter method of the FilterChain.
 - This invokes next filter (if any) or actual resource
- Register the filter with the appropriate servlets and JSP pages.
 - Use filter and filter-mapping in *web.xml*.
- Disable invoker servlet.
 - See earlier slide

www.moreservlets.com

The doFilter Method

- Basic format


```
public void doFilter(ServletRequest request,
                    ServletResponse response,
                    FilterChain chain)
    throws ServletException, IOException {
    ...
    chain.doFilter(request, response);
}
```
- Note on first two arguments
 - They are of type ServletRequest and ServletResponse, not HttpServletRequest and HttpServletResponse.
 - Do a typecast if you need HTTP-specific capabilities
- Note on final argument
 - It is a FilterChain, not a Filter. Its doFilter method is different – two arguments only.

www.moreservlets.com

A Simple Reporting Filter

```
public class ReportFilter implements Filter {
    public void doFilter(ServletRequest request,
        ServletResponse response,
        FilterChain chain)
        throws ServletException, IOException {
        HttpServletRequest req =
            (HttpServletRequest)request;
        System.out.println(req.getRemoteHost() +
            " tried to access " +
            req.getRequestURL() +
            " on " + new Date() + ".");
        chain.doFilter(request, response);
    }
}
```

Servlet and JSP Filters

www.moreservlets.com

A Simple Reporting Filter (Continued)

```
public void init(FilterConfig config)
    throws ServletException {
}

public void destroy() {
}
```

Servlet and JSP Filters

www.moreservlets.com

Declaring the Reporting Filter

```
...
<web-app>
  <!-- Register the name "Reporter"
    for ReportFilter. -->
  <filter>
    <filter-name>Reporter</filter-name>
    <filter-class>
      moreservlets.filters.ReportFilter
    </filter-class>
  </filter>

```

Servlet and JSP Filters

www.moreservlets.com

Associating Reporting Filter with Given URLs

```
<!-- Apply Reporter filter to home page. -->
<filter-mapping>
  <filter-name>Reporter</filter-name>
  <url-pattern>/index.jsp</url-pattern>
</filter-mapping>

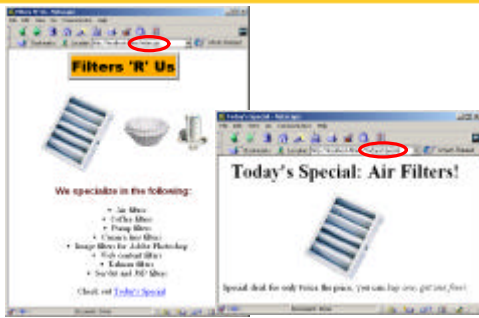
<!-- Also apply Reporter filter to
  servlet named "TodaysSpecial". -->
<filter-mapping>
  <filter-name>Reporter</filter-name>
  <servlet-name>TodaysSpecial</servlet-name>
</filter-mapping>
...
</web-app>

```

Servlet and JSP Filters

www.moreservlets.com

Reporting Filter: Results



Servlet and JSP Filters

www.moreservlets.com

Reporting Filter (Results Continued)

- Printouts to standard output akin to the following will result from the two accesses shown on previous page:
 - purchasing.sun.com tried to access <http://www.filtersrus.com/filters/index.jsp> on Fri Apr 12 13:19:14 EDT 2002.
 - admin.microsoft.com tried to access <http://www.filtersrus.com/filters/TodaysSpecial> on Fri Apr 12 13:21:56 EDT 2002.
- Point: A single filter can apply to lots of different resources in transparent manner
 - The individual resources do not need any special code

Servlet and JSP Filters

www.moreservlets.com

Accessing the Servlet Context

- **What if filter wants to read or write Web application-wide parameters? Or it simply wants to log data?**
 - You use methods in ServletContext for this
- **Surprisingly, the doFilter method provides no access to the ServletContext**
 - Neither ServletRequest nor ServletResponse provides access to it either
- **Solution: store the ServletContext in init**
 - Call getServletContext on the FilterConfig argument that is passed to init
 - Store the result in an instance variable (field) of the filter
 - Access the field from the doFilter method

Servlet and JSP Filters www.moreservlets.com

A Logging Filter

```
public class LogFilter implements Filter {
    protected FilterConfig config;
    private ServletContext context;
    private String filterName;

    public void init(FilterConfig config)
        throws ServletException {
        // In case it is needed by subclass.
        this.config = config;
        context = config.getServletContext();
        filterName = config.getFilterName();
    }
}
```

Servlet and JSP Filters www.moreservlets.com

A Logging Filter (Continued)

```
public void doFilter(ServletRequest request,
    ServletResponse response,
    FilterChain chain)
    throws ServletException, IOException {
    HttpServletRequest req =
        (HttpServletRequest)request;
    context.log(req.getRemoteHost() +
        " tried to access " +
        req.getRequestURL() +
        " on " + new Date() + ". " +
        "(Reported by " +
        filterName + ".)");
    chain.doFilter(request, response);
}
```

Servlet and JSP Filters www.moreservlets.com

Applying Logging Filter to Entire Web Application

```
<web-app>
...
<filter>
  <filter-name>Logger</filter-name>
  <filter-class>
    moreservlets.filters.LogFilter
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>Logger</filter-name>
  <url-pattern>*/</url-pattern>
</filter-mapping>
...
</web-app>
```

Servlet and JSP Filters www.moreservlets.com

Logging Filter: Results

- **Log file:**
 - audits.irs.gov tried to access
http://www.filtersrus.com/filters/business-plan.jsp
on Fri Apr 12 15:16:15 EDT 2002.
(Reported by Logger.)
 - ceo.enron.com tried to access
http://www.filtersrus.com/filters/tax-shelter/
on Tue Feb 12 10:24:11 EDT 2002.
(Reported by Logger.)

Servlet and JSP Filters www.moreservlets.com

Using Filter Initialization Parameters

- **Reminder: who needs to customize servlet and JSP behavior?**
 - Developers.
 - They customize the behavior by changing the code of the servlet or JSP page itself.
 - End users.
 - They customize the behavior by entering values in HTML forms.
 - Deployers.
 - This third group is the one served by initialization parameters. Members of this group are people who take existing Web applications (or individual servlets or JSP pages) and deploy them in a customized environment.
- **Resources with initialization parameters**
 - Servlets, JSP pages, servlet context, **filters**, listeners.

Servlet and JSP Filters www.moreservlets.com

Declaring Filter Initialization Parameters

```
<filter>
<filter-name>LateAccessFilter</filter-name>
<filter-class>
  moreservlets.filters.LateAccessFilter
</filter-class>
<init-param>
  <param-name>startTime</param-name>
  <param-value>2</param-value>
</init-param>
<init-param>
  <param-name>endTime</param-name>
  <param-value>10</param-value>
</init-param>
</filter>
```

16

Servlet and JSP Filters

www.moreservlets.com

Reading Init Params: An Access Time Filter

```
public void init(FilterConfig config)
  throws ServletException {
  this.config = config;
  context = config.getServletContext();
  formatter =
    DateFormat.getDateInstance(DateFormat.MEDIUM,
      DateFormat.MEDIUM);

  try {
    startTime = Integer.parseInt
      (config.getInitParameter("startTime"));
    endTime = Integer.parseInt
      (config.getInitParameter("endTime"));
  } catch (NumberFormatException nfe) { // Malformed/null
    // Default: access at or after 10 p.m. but before 6
    // a.m. is considered unusual.
    startTime = 22; // 10:00 p.m.
    endTime = 6; // 6:00 a.m.
  }
}
```

20

Servlet and JSP Filters

www.moreservlets.com

An Access Time Filter (Continued)

```
public void doFilter(ServletRequest request,
  ServletResponse response,
  FilterChain chain)
  throws ServletException, IOException {
  HttpServletRequest req =
    (HttpServletRequest)request;
  GregorianCalendar calendar =
    new GregorianCalendar();
  int currentTime =
    calendar.get(calendar.HOUR_OF_DAY);
  if (isUnusualTime(currentTime, startTime, endTime)) {
    context.log("WARNING: " +
      req.getRemoteHost() +
      " accessed " +
      req.getRequestURL() +
      " on " +
      formatter.format(calendar.getTime()));
  }
  chain.doFilter(request, response);
}
```

21

Servlet and JSP Filters

www.moreservlets.com

Blocking the Response

- **Idea**
 - Normal situation: call doFilter on FilterChain object
 - Unusual situation: redirect response or generate custom output

- **Generic Example**

```
public void doFilter(ServletRequest request,
  ServletResponse response,
  FilterChain chain)
  throws ServletException, IOException {
  HttpServletRequest req =
    (HttpServletRequest)request;
  HttpServletResponse res =
    (HttpServletResponse)response;
  if (isUnusualCondition(req)) {
    res.sendRedirect("http://www.somesite.com");
  } else {
    chain.doFilter(req, res);
  }
}
```

22

Servlet and JSP Filters

www.moreservlets.com

A Banned Site Filter

```
public class BannedAccessFilter implements Filter {
  private HashSet bannedSiteTable;

  public void init(FilterConfig config)
    throws ServletException {
    bannedSiteTable = new HashSet();
    String bannedSites =
      config.getInitParameter("bannedSites");
    StringTokenizer tok =
      new StringTokenizer(bannedSites);
    while(tok.hasMoreTokens()) {
      String bannedSite = tok.nextToken();
      bannedSiteTable.add(bannedSite);
      System.out.println("Banned " + bannedSite);
    }
  }
  public void destroy() {}
}
```

23

Servlet and JSP Filters

www.moreservlets.com

A Banned Site Filter (Continued)

```
public void doFilter(ServletRequest request,
  ServletResponse response,
  FilterChain chain)
  throws ServletException, IOException {
  HttpServletRequest req = (HttpServletRequest)request;
  String requestingHost = req.getRemoteHost();
  String referringHost =
    getReferringHost(req.getHeader("Referer"));
  String bannedSite = null;
  boolean isBanned = false;
  if (bannedSiteTable.contains(requestingHost)) {
    bannedSite = requestingHost; isBanned = true;
  } else if (bannedSiteTable.contains(referringHost)) {
    bannedSite = referringHost; isBanned = true;
  }
  if (isBanned) {
    showWarning(response, bannedSite); // Custom response
  } else {
    chain.doFilter(request, response);
  }
}
```

24

Servlet and JSP Filters

www.moreservlets.com

A Banned Site Filter (Continued)

```
private String getReferringHost
(String referringURLString) {
    try {
        URL referringURL =
            new URL(referringURLString);
        return(referringURL.getHost());

        // Malformed or null
    } catch (MalformedURLException mue) {
        return(null);
    }
}
```

26

Servlet and JSP Filters

www.moreservlets.com

A Banned Site Filter (Continued)

```
private void showWarning(ServletResponse response,
String bannedSite)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String docType =
        "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 " +
        " Transitional//EN\">\n";
    out.println
        (docType +
        "<HTML>\n" +
        "<HEAD><TITLE>Access Prohibited</TITLE></HEAD>\n" +
        "<BODY BGCOLOR=\"WHITE\">\n" +
        "<H1>Access Prohibited</H1>\n" +
        "Sorry, access from or via " + bannedSite + "\n" +
        "is not allowed.\n" +
        "</BODY></HTML>");
```

26

Servlet and JSP Filters

www.moreservlets.com

Registering the Banned Site Filter in web.xml

```
<web-app>
  <filter>
    <filter-name>BannedAccessFilter</filter-name>
    <filter-class>
      moreservlets.filters.BannedAccessFilter
    </filter-class>
    <init-param>
      <param-name>bannedSites</param-name>
      <param-value>
        www.competingsite.com
        www.bettersite.com
        www.moreservlets.com
      </param-value>
    </init-param>
  </filter>
```

27

Servlet and JSP Filters

www.moreservlets.com

Registering the Banned Site Filter (Continued)

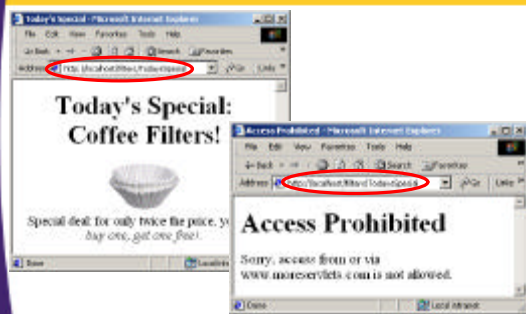
```
<filter-mapping>
  <filter-name>BannedAccessFilter</filter-name>
  <servlet-name>TodaysSpecial</servlet-name>
</filter-mapping>
<servlet>
  <servlet-name>TodaysSpecial</servlet-name>
  <servlet-class>
    moreservlets.TodaysSpecialServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>TodaysSpecial</servlet-name>
  <url-pattern>/TodaysSpecial</url-pattern>
</servlet-mapping>
...
</web-app>
```

28

Servlet and JSP Filters

www.moreservlets.com

Filter in Action



29

Servlet and JSP Filters

www.moreservlets.com

Advanced Filters: Modifying the Response

- **Create a response wrapper.**
 - Extend HttpServletResponseWrapper.
- **Provide a PrintWriter that buffers output.**
 - Override getWriter method to return a PrintWriter that saves everything sent to it and stores that result in a field.
- **Pass that wrapper to doFilter.**
 - This call is legal because HttpServletResponseWrapper implements HttpServletResponse.
- **Extract and modify the output.**
 - After call to doFilter method of the FilterChain, output of the original resource is available to you through whatever mechanism you provided in Step 2. Modify or replace it as appropriate.
- **Send the modified output to the client.**
 - Original resource no longer sends output to client (output is stored in your response wrapper instead). You have to send the output. So, filter needs to obtain the PrintWriter or OutputStream from original response object and pass modified output to that stream.

30

Servlet and JSP Filters

www.moreservlets.com

A Reusable Response Wrapper

```
public class CharArrayWrapper
    extends HttpServletResponseWrapper {
    private CharArrayWriter charWriter;

    public CharArrayWrapper(HttpServletResponse response) {
        super(response);
        charWriter = new CharArrayWriter();
    }
    public PrintWriter getWriter() {
        return(new PrintWriter(charWriter));
    }
    public String toString() {
        return(charWriter.toString());
    }
    public char[] toCharArray() {
        return(charWriter.toCharArray());
    }
}
```

31

Servlet and JSP Filters

www.moreservlets.com

A Generic Replacement Filter

```
public abstract class ReplaceFilter implements Filter {
    private FilterConfig config;

    public void doFilter(ServletRequest request,
        ServletResponse response,
        FilterChain chain)
        throws ServletException, IOException {
        CharArrayWrapper responseWrapper =
            new CharArrayWrapper((HttpServletResponse)response);
        chain.doFilter(request,responseWrapper);
        String responseString = responseWrapper.toString();
        responseString =
            FilterUtils.replace(responseString,
                getTargetString(),
                getReplacementString());
        updateHeaders(response, responseString);
        PrintWriter out = response.getWriter();
        out.write(responseString);
    }
}
```

32

Servlet and JSP Filters

www.moreservlets.com

A Generic Replacement Filter (Continued)

```
public void init(FilterConfig config)
    throws ServletException {
    this.config = config;
}

protected FilterConfig getFilterConfig() {
    return(config);
}

public void destroy() {}

public abstract String getTargetString();

public abstract String getReplacementString();

public void updateHeaders(ServletResponse response,
    String responseString) {
    response.setContentLength(responseString.length());
}
}
```

33

Servlet and JSP Filters

www.moreservlets.com

FilterUtils

```
public static String replace(String mainString,
    String orig,
    String replacement) {

    String result = "";
    int oldIndex = 0;
    int index = 0;
    int origLength = orig.length();
    while((index = mainString.indexOf(orig, oldIndex))
        != -1) {
        result = result +
            mainString.substring(oldIndex, index) +
            replacement;
        oldIndex = index + origLength;
    }
    result = result + mainString.substring(oldIndex);
    return(result);
}
```

34

Servlet and JSP Filters

www.moreservlets.com

A Specific Replacement Filter

```
public class ReplaceSiteNameFilter
    extends ReplaceFilter {

    public String getTargetString() {
        return("filtersRus.com");
    }

    public String getReplacementString() {
        return("weBefilters.com");
    }
}
```

35

Servlet and JSP Filters

www.moreservlets.com

A Specific Replacement Filter (Continued)

```
<web-app>
...
<filter>
  <filter-name>ReplaceSiteNameFilter</filter-name>
  <filter-class>
    moreservlets.filters.ReplaceSiteNameFilter
  </filter-class>
</filter>

<filter-mapping>
  <filter-name>ReplaceSiteNameFilter</filter-name>
  <url-pattern>/plugSite/page2.jsp</url-pattern>
</filter-mapping>
...
</web-app>
```


36

Servlet and JSP Filters

www.moreservlets.com

More Information

- **Source code for examples**
 - <http://www.moreservlets.com>
- **More Servlets & JSP**
 - <http://www.moreservlets.com>
 - Site includes info on servlet and JSP training courses
- **Core Servlets & JSP**
 - Prequel to *More Servlets & JSP*
 - <http://www.coreservlets.com>
- **Servlet home page**
 - <http://java.sun.com/products/servlet/>
- **JavaServer Pages home page**
 - <http://java.sun.com/products/jsp/>



© Scott Stoltz, JSP Filters
www.moreservlets.com