**Dharmsinh Desai University, Nadiad**

**Department of Information Technology**

**Advanced Java Technology, IT619**

**B.Tech. IT, Sem: VI**

**Experiment – 05**

**Submitted By**

**Roll No.:**IT076

**Name:Dishant Modh**

**Aim:**  Write code for implementation of the two filters, Log Filter and Authentication

Filter, in filter chain. Client calls the Log Filter. The Log filter logs the time of

arrival of request and IP address of the client. The Log filter forwards the request

to Authentication Filter. The authentication filter authenticates the client and allow

to access the targeted servlet.

**Code:**

**index.html**

# Login Form

| Name: |  |
| --- | --- |
| Password: |  |

# <html>

# <body style="background-color: powderblue">

# <h1>Login Form</h1>

# <form action="login" >

# <table>

# <tr>

# <td> <label> Name: </label> </td>

# <td> <input type="text" name="name"/></td>

# </tr>

# <tr>

<td><label>Password:</label></td>

<td><input type="password" name="password"/></td>

</tr>

</table>

<input type="submit" value="login">

</form>

</body>

</html>

**login.java**

package filter;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class login extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

//processRequest(request, response);

response.setContentType("text/html");

PrintWriter out = response.getWriter();

out.print("welcome ADMIN");

out.close();

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**logfilter.java**

package filter;

import java.io.IOException;

import java.io.PrintStream;

import java.io.PrintWriter;

import java.io.StringWriter;

import java.util.Date;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

public class logfilter implements Filter {

private static final boolean debug = true;

private FilterConfig filterConfig = null;

public logfilter() {

}

private void doBeforeProcessing(ServletRequest request, ServletResponse response)

throws IOException, ServletException {

if (debug) {

log("logfilter:DoBeforeProcessing");

}

}

private void doAfterProcessing(ServletRequest request, ServletResponse response)

throws IOException, ServletException {

if (debug) {

log("logfilter:DoAfterProcessing");

}

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain)

throws IOException, ServletException {

String ipAddress = request.getRemoteAddr();

System.out.println("Ip: "+ ipAddress + "Time: " + new Date().toString() + "\n\n");

chain.doFilter(request,response);

}

public FilterConfig getFilterConfig() {

return (this.filterConfig);

}

public void setFilterConfig(FilterConfig filterConfig) {

this.filterConfig = filterConfig;

}

public void destroy() {

}

public void init(FilterConfig filterConfig) {

this.filterConfig = filterConfig;

if (filterConfig != null) {

if (debug) {

log("logfilter:Initializing filter");

}

}

}

@Override

public String toString() {

if (filterConfig == null) {

return ("logfilter()");

}

StringBuffer sb = new StringBuffer("logfilter(");

sb.append(filterConfig);

sb.append(")");

return (sb.toString());

}

private void sendProcessingError(Throwable t, ServletResponse response) {

String stackTrace = getStackTrace(t);

if (stackTrace != null && !stackTrace.equals("")) {

try {

response.setContentType("text/html");

PrintStream ps = new PrintStream(response.getOutputStream());

PrintWriter pw = new PrintWriter(ps);

pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body>\n");

pw.print("<h1>The resource did not process correctly</h1>\n<pre>\n");

pw.print(stackTrace);

pw.print("</pre></body>\n</html>"); //NOI18N

pw.close();

ps.close();

response.getOutputStream().close();

} catch (Exception ex) {

}

} else {

try {

PrintStream ps = new PrintStream(response.getOutputStream());

t.printStackTrace(ps);

ps.close();

response.getOutputStream().close();

} catch (Exception ex) {

}

}

}

public static String getStackTrace(Throwable t) {

String stackTrace = null;

try {

StringWriter sw = new StringWriter();

PrintWriter pw = new PrintWriter(sw);

t.printStackTrace(pw);

pw.close();

sw.close();

stackTrace = sw.getBuffer().toString();

} catch (Exception ex) {

}

return stackTrace;

}

public void log(String msg) {

filterConfig.getServletContext().log(msg);

}

}

**authentication.java**

package filter;

import java.io.IOException;

import java.io.PrintStream;

import java.io.PrintWriter;

import java.io.StringWriter;

import javax.servlet.\*;

public class authentication implements Filter {

private static final boolean debug = true;

private FilterConfig filterConfig = null;

public authentication() {

}

private void doBeforeProcessing(ServletRequest request, ServletResponse response)

throws IOException, ServletException {

if (debug) {

log("authentication:DoBeforeProcessing");

}

}

private void doAfterProcessing(ServletRequest request, ServletResponse response)

throws IOException, ServletException {

if (debug) {

log("authentication:DoAfterProcessing");

}

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain)

throws IOException, ServletException {

PrintWriter out=response.getWriter();

String password=request.getParameter("password");

if(password.equals("admin")){

chain.doFilter(request, response);//sends request to next resource

}

else{

out.print("username or password error!");

RequestDispatcher rd=request.getRequestDispatcher("index.html");

rd.include(request, response);

}

}

public FilterConfig getFilterConfig() {

return (this.filterConfig);

}

public void setFilterConfig(FilterConfig filterConfig) {

this.filterConfig = filterConfig;

}

public void destroy() {

}

public void init(FilterConfig filterConfig) {

this.filterConfig = filterConfig;

if (filterConfig != null) {

if (debug) {

log("authentication:Initializing filter");

}

}

}

@Override

public String toString() {

if (filterConfig == null) {

return ("authentication()");

}

StringBuffer sb = new StringBuffer("authentication(");

sb.append(filterConfig);

sb.append(")");

return (sb.toString());

}

private void sendProcessingError(Throwable t, ServletResponse response) {

String stackTrace = getStackTrace(t);

if (stackTrace != null && !stackTrace.equals("")) {

try {

response.setContentType("text/html");

PrintStream ps = new PrintStream(response.getOutputStream());

PrintWriter pw = new PrintWriter(ps);

pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body style=\"background-color: powderblue\">\n");

pw.print("<h1>The resource did not process correctly</h1>\n<pre>\n");

pw.print(stackTrace);

pw.print("</pre></body>\n</html>"); //NOI18N

pw.close();

ps.close();

response.getOutputStream().close();

} catch (Exception ex) {

}

} else {

try {

PrintStream ps = new PrintStream(response.getOutputStream());

t.printStackTrace(ps);

ps.close();

response.getOutputStream().close();

} catch (Exception ex) {

}

}

}

public static String getStackTrace(Throwable t) {

String stackTrace = null;

try {

StringWriter sw = new StringWriter();

PrintWriter pw = new PrintWriter(sw);

t.printStackTrace(pw);

pw.close();

sw.close();

stackTrace = sw.getBuffer().toString();

} catch (Exception ex) {

}

return stackTrace;

}

public void log(String msg) {

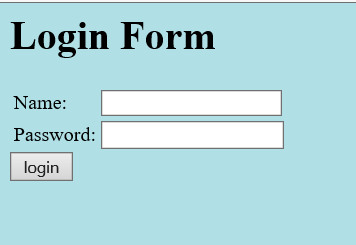
filterConfig.getServletContext().log(msg);

}

}

**Input/Output:**

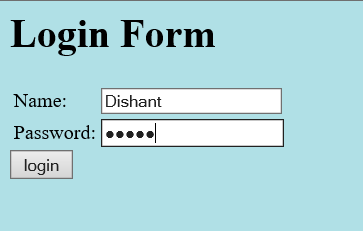
This is simple Login Form



When we enter any wrong password or the Username



After entering the Perfect/Correct data



It get successfully logged in

