

TDM03 de Technologie Web 2: Servlets

ASI4 - INSA Rouen
CORRECTION

1 Filtrage

Développez trois filtres :

1. Le premier filtre écrit simplement deux messages dans un fichier Log, un à la réception de la requête et un à la fin de son traitement.
2. Le second filtre fixe l'encodage des requêtes de l'application web par un paramètre d'initialisation de la Servlet. La sortie est, quant à elle, fixée à de l'UTF-8.
3. Le dernier filtre ajoute en pied de page à toutes les Servlets et JSP de l'application web la date de premier appel du filtre.

Vous appliquerez le premier filtre à tous les fichiers (textes, pages HTML, JSP et Servlets). Le deux autres filtres seront appliqués uniquement aux Servlets et JSP.

Ajoutez des écritures dans le fichier Log pour chacun des filtres et vérifiez l'ordre d'appel.

Correction

web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <filter>
    <filter-name>FiltreLog</filter-name>
    <filter-class>FiltreLog</filter-class>
  </filter>
  <filter>
    <filter-name>FiltreRequete</filter-name>
    <filter-class>FiltreRequete</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>
  <filter>
    <filter-name>FiltrePied</filter-name>
    <filter-class>FiltrePied</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>FiltreLog</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>FiltreRequete</filter-name>
    <url-pattern>*.jsp</url-pattern>
    <url-pattern>/servlet/*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>FiltrePied</filter-name>
    <url-pattern>*.jsp</url-pattern>
    <url-pattern>/servlet/*</url-pattern>
  </filter-mapping>

  <servlet>
    <servlet-name>Salut</servlet-name>
    <servlet-class>Salut</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Salut</servlet-name>
    <url-pattern>/servlet/salut</url-pattern>
  </servlet-mapping>
</web-app>
```

FiltreLog.java

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FiltreLog implements Filter {
    private FilterConfig fconfig;

    public void init(FilterConfig fconfig) {
        this.fconfig = fconfig;
    }

    public void doFilter(ServletRequest requete, ServletResponse reponse, FilterChain
        chain)
        throws ServletException, IOException {
        ServletContext context = this.fconfig.getServletContext();
        context.log("Appel de FiltreLog avant le doFilter");
        chain.doFilter(requete, reponse);
        context.log("Appel de FiltreLog après le doFilter");
    }

    public void destroy() {
        this.fconfig = null;
    }
}
```

FiltreRequete.java

```
import java.io.IOException;
import javax.servlet.*;
import java.util.Locale;

public class FiltreRequete implements Filter {
    private String encoding;
    private FilterConfig fconfig;

    public void init(FilterConfig filterConfig) throws ServletException {
        this.fconfig = filterConfig;
        this.encoding = filterConfig.getInitParameter("encoding");
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
        filterChain) throws IOException, ServletException {
        ServletContext context = this.fconfig.getServletContext();
        context.log("Appel de FiltreRequête avant le doFilter");
        request.setCharacterEncoding(encoding);
        filterChain.doFilter(request, response);
        context.log("Appel de FiltreRequête après le doFilter");
    }

    public void destroy() {
        this.encoding = null;
        this.fconfig = null;
    }
}
```

FiltrePied.java

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FiltrePied implements Filter {
    private SimpleDateFormat formateur;
    private Date date;
    private FilterConfig fconfig;

    public void init(FilterConfig fconfig) {
        this.fconfig = fconfig;
        this.formateur = new SimpleDateFormat("E d MMM yyyy, H:m:s", Locale.FRANCE);
        this.date = new Date();
        this.date.setTime(System.currentTimeMillis());
    }

    public void doFilter(ServletRequest requete, ServletResponse reponse, FilterChain
        chain)
        throws ServletException, IOException {
        ServletContext context = this.fconfig.getServletContext();
        PrintWriter out = reponse.getWriter();
        context.log("Appel de FiltrePied avant le doFilter");
        chain.doFilter(requete, reponse);
    }
}
```

```

        out.println("<p>Dernière réinitialisation du serveur : " + this.formateur.format(
this.date)+"</p>");
        context.log("Appel de FiltrePied après le doFilter");
    }

    public void destroy() {
        this.formateur = null;
        this.date = null;
        this.fconfig = null;
    }
}

```

hello.jsp

```

<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>
  <head>
    <title>Hello filtré</title>
  </head>
  <body>
    <% String prenom=request.getParameter("prenom"); %>
    <h1>Bonjour <%= (prenom!=null && prenom.length() !=0)?prenom:"bel(le) inconnu(e)"%></h1>
    <% if (prenom!=null && prenom.equals("le monde")) { %>
      <h2>Bien joué !!!</h2>
    <% } %>
    <form action="hello.jsp" method="post">
      <label>Prénom : </label><input type="text" name="prenom" size="30">
      <input type="submit" value="envoyer">
    </form>
  </body>
</html>

```

Salut.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Salut extends HttpServlet {
    public void doGet(HttpServletRequest requete, HttpServletResponse reponse) throws
        ServletException, IOException {
        PrintWriter out = reponse.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<body><p>Salut, salut !!!</p></body>");
        out.println("</html>");
    }
}

```

createFiltres.sh

```

CLASSPATH=./src:$JBOSS_HOME/modules/system/layers/base/javax/servlet/api/main/jboss-
servlet-api_4.0_spec-1.0.0.Final.jar
#jboss-servlet-api_3.1_spec-1.0.0.Final.jar

javac -cp $CLASSPATH -sourcepath src -d ./WEB-INF/classes src/*.java

jar cf Filtres.war WEB-INF hello.html hello.jsp hello.txt
cp Filtres.war $JBOSS_HOME/standalone/deployments

```

2 Sessions et filtres

Créez diverses ressources (JSP, pages HTML ou textuelles, etc.) dans un répertoire dont vous restreindrez l'accès. Pour cela, vous mettrez en place un filtrage qui n'affichera la ressource demandée que si l'utilisateur est connecté. Dans le cas contraire, une redirection sera effectuée vers une page de connexion.

La page de connexion sera une simple JSP, accessible depuis la partie publique de votre application et contiendra uniquement un formulaire pour saisir son login (pour simplifier, aucune vérification de mot de passe ne sera faite dans cet exercice). Cette JSP créera la session avec les informations correspondantes.

Vous ajouterez également une page de déconnexion.

Correction

index.jsp

```

<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>

```

```

<head>
  <title >Bienvenue</title >
</head>
<body>
  <p>Bienvenue sur notre site !<p>
  <a href="login.jsp">Connexion</a>
</body>
</html>

```

login.jsp

```

<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>
  <body>
    <%
      String nom=request.getParameter("nom");
      if(request.getParameter("nom") != null)
        session.setAttribute("nom", nom);
      if(session.getAttribute("nom") != null) {
    %>
    <p>Vous êtes connecté en tant que <%=session.getAttribute("nom") %>.</p>
    <a href="logout.jsp">Déconnexion</a>
    <% } else { %>
    <form action="login.jsp" method="post">
      <label>Nom : </label><input type="text" name="nom" size="30">
      <input type="submit" value="envoyer">
    </form>
    <% } %>
  </body>
</html>

```

logout.jsp

```

<%@ page contentType="text/html; charset=utf-8" %>
<!DOCTYPE html>
<html>
  <body>
    <% session.invalidate(); %>
    <p>Vous êtes maintenant déconnecté.</p>
    <a href="login.jsp">Connexion</a>
  </body>
</html>

```

Restriction.java

```

import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Restriction implements Filter {
  private FilterConfig fconfig;

  public void init(FilterConfig fconfig) {
    this.fconfig = fconfig;
  }

  public void doFilter(ServletRequest requete, ServletResponse reponse, FilterChain
    chain)
    throws ServletException, IOException {
    ServletContext context = this.fconfig.getServletContext();
    /* Cast des objets request et response */
    HttpServletRequest request = (HttpServletRequest) requete;
    HttpServletResponse response = (HttpServletResponse) reponse;

    /* Récupération de la session depuis la requête */
    HttpSession session = request.getSession();

    /**
     * Si l'objet utilisateur n'existe pas dans la session en cours, alors
     * l'utilisateur n'est pas connecté.
     */
    if ( session.getAttribute("nom")==null ) {
      /* Redirection vers la page de login */
      response.sendRedirect( request.getContextPath() + "/login.jsp" );
    }
    else if( !session.getAttribute("nom").equals("Bob") ){
      /* Redirection vers une page publique */
      response.sendRedirect( request.getContextPath() + "/connected.jsp" );
    }
    else {
      /* Affichage de la page restreinte */
      chain.doFilter( request, response );
    }
  }
}

```

```

    }
}

public void destroy() {
    this.fconfig = null;
}
}

```

web.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <session-config>
    <session-timeout>1</session-timeout>
  </session-config>
  <filter>
    <filter-name>Restriction</filter-name>
    <filter-class>Restriction</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Restriction</filter-name>
    <url-pattern>/restreint/*</url-pattern>
  </filter-mapping>
</web-app>

```

Remarques

1. Les applications JEE nécessitent un serveur d'application web JEE (*ex* : Tomcat ou JBoss/WildFly). Un tel serveur doit être installé **en local** sur votre machine en salle de TP, dans le répertoire **/tmp**. Au cours de ce TP vous utiliserez WildFly dont vous pourrez récupérer l'archive dans le répertoire **/uv/asi/technoweb/**. Vérifiez bien que vos variables systèmes **\$JBoss_HOME** et **\$JAVA_HOME** pointent vers les bons répertoires.
2. Pensez à vérifier vos pages HTML et CSS sur le site du W3C (<http://validator.w3.org/>) ou n'importe quel autre validateur.
3. À l'issue de la séance, vous aurez accès à la correction de ce TDM au format PDF.