

# TDM01 de Technologies Web 2: Servlets (1)

## ASI4 - INSA Rouen

### CORRECTION

## Forum ASI (1) : Servlets

L'objectif de l'exercice est de développer le code nécessaire au dépôt de messages dans un forum, à l'aide de Servlets. Vous développerez une page affichant un formulaire identique à celui fourni en annexe, ainsi que les messages déjà laissés par les précédents visiteurs.

Les différents messages seront sauvegardés dans un fichier côté serveur, dans le répertoire `/tmp`. Vous pourrez par exemple créer une classe `UnMessage` que vous sèrialiserez et dé-sèrialiserez (remarque : si besoin, la classe `Message` existe déjà dans l'API Servlets).

### Correction

```
<?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>
  <servlet>
    <servlet-name>forum</servlet-name>
    <servlet-class>Forum</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>forum</servlet-name>
    <url-pattern>/Forum</url-pattern>
  </servlet-mapping>
</web-app>

import java.io.*;
import java.text.*;
import java.util.*;

// Cette classe s'appelle UnMessage au lieu de Message car cette derniere existe dans l'
// API servlet
// Implementation de l'interface Serializable qui permet l'utilisation de flux
// manipulant les objets,
// donc l'écriture/lecture d'objets dans des fichiers
public class UnMessage implements Serializable {
  private static SimpleDateFormat formatter = new SimpleDateFormat("E d MMM yyyy, H:m:s"
    , Locale.FRANCE);
  private static Date date = new Date();

  private long timestamp;
  private String email;
  private String texte;

  public UnMessage(String _email, String _texte) {
    timestamp = System.currentTimeMillis();
    email = _email;
    texte = _texte;
  }

  public long getTimeStamp() {
    return timestamp;
  }

  public String getDate() {
    date.setTime(getTimeStamp());
    return formatter.format(date);
  }

  public String getEmail() {
    return email;
  }
}
```

```

    }

    public String getTexte() {
        return texte;
    }
}

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

public class Forum extends HttpServlet {
    private String cheminFichierMessages;

    public void init() {
        // recuperation du nom du fichier stockant les messages et construction du chemin ou
        // il sera stocke
        this.cheminFichierMessages = "/tmp/messages.txt";
    }

    // Post est appelee lorsqu'il y a un message a stocker (cf le formulaire)
    public void doPost(HttpServletRequest requete, HttpServletResponse reponse)
        throws ServletException, IOException {
        String texte = requete.getParameter("message");
        String email = requete.getParameter("email");
        this.enregistrerMessage(email, texte);
        this.afficher(requete, reponse);
    }

    // Get est appelee au premier acces a la page
    public void doGet(HttpServletRequest requete, HttpServletResponse reponse)
        throws ServletException, IOException {
        this.afficher(requete, reponse);
    }

    // affiche le code HTML enrobant, le formulaire et les messages
    public void afficher(HttpServletRequest requete, HttpServletResponse reponse) throws
        IOException {
        reponse.setContentType("text/html");

        PrintWriter out = reponse.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Forum ASI</title>");
        out.println("<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>");
        out.println("<link href='" + requete.getContextPath() + "/Ressources/style.css' rel
            ='stylesheet' type='text/css' />");
        out.println("</head>");
        out.println("<body>");
        this.afficherFormulaire(requete, reponse);
        this.afficherMessages(requete, reponse);
        out.println("</body>");
        out.println("</html>");
    }

    public void enregistrerMessage(String email, String texte) {
        UnMessage msg = new UnMessage(email, texte);
        try {
            // Utilisation de la serialisation de UnMessage pour les stocker dans un fichier
            File file = new File(this.cheminFichierMessages);
            FileOutputStream fos = new FileOutputStream(file, true);
            ObjectOutputStream oos;
            if (file.length() > 0)
                // en cas d'ajout de message au fichier, l'entete de serialisation ne doit pas
                // etre ajoute
                oos = new ObjectOutputStream(fos) {
                    protected void writeStreamHeader()
                        throws IOException {
                    }
                };
            else
                oos = new ObjectOutputStream(fos);
            // serialisation de l'objet dans le fichier via le flux cos
            oos.writeObject(msg);
            fos.close();
        }
        catch (FileNotFoundException exception) {
            log("Forum exception : fichier inexistant pour ecriture");
        }
        catch (IOException exception) {

```

```

        log("Forum exception : probleme d'ecriture");
        log("Forum exception : "+exception);
    }
}

public void afficherMessages(HttpServletRequest requete, HttpServletResponse reponse)
{
    try {
        PrintWriter out = reponse.getWriter();
        // Utilisation de la serialisation de UnMessage pour les lire d'un fichier
        FileInputStream fis = new FileInputStream(this.cheminFichierMessages);
        ObjectInputStream ois = new ObjectInputStream(fis);

        UnMessage msg;
        while (fis.available() > 0) {
            // lecture de l'objet dans le flux
            msg = (UnMessage)ois.readObject();
            out.println("<table width='80%' border='1'>");
            out.println("<tr>");
            out.println("<td>"+msg.getDate()+"</td>");
            out.println("<td>"+msg.getEmail()+"</td>");
            out.println("</tr>");
            out.println("<tr>");
            out.println("<td colspan='2'><pre>"+msg.getTexte()+"</pre></td>");
            out.println("</tr>");
            out.println("</table>");
        }
        fis.close();
    }
    catch (FileNotFoundException exception) {
        log("Forum exception : fichier inexistant pour lecture");
    }
    catch (IOException exception) {
        log("Forum exception : probleme de lecture");
        log("Forum exception : "+exception);
    }
    catch (ClassNotFoundException exception) {
        log("Forum exception : tentative d'instanciation d'une classe inexistante");
    }
}

public void afficherFormulaire(HttpServletRequest requete, HttpServletResponse reponse)
{
    try {
        PrintWriter out = reponse.getWriter();
        out.println("<header id='top'>");
        out.println("<img id='logo' src='\" + requete.getContextPath() + \"/Images/logo-asi.png' alt='ASI' width='125' height='58' />");
        out.println("<h1>Forum ASI</h1>");
        out.println("</header>");
        out.println("<form name='formulaire' action='Forum' method='post'>");
        out.println("<fieldset>");
        out.println("<legend>Coordonnées :</legend>");
        out.println("<label for='name'>Nom : </label><input placeholder='Saisissez votre nom' id='name' name='name' type='text' size='30' />");
        out.println("<label for='email'>E-mail : </label><input placeholder='Saisissez votre email' id='email' name='email' type='text' size='30' />");
        out.println("</fieldset>");
        out.println("<fieldset>");
        out.println("<legend>Message :</legend>");
        out.println("<textarea rows='4' cols='50' id='message' name='message'></textarea>");
        out.println("</fieldset>");
        out.println("<input type='submit' value='Poster le message' />");
        out.println("<input type='reset' value='Effacer' />");
        out.println("</form>");
    }
    catch (IOException exception) {
        log("Forum exception : probleme de lecture");
        log("Forum exception : "+exception);
    }
}
}
}

```

## Remarques

1. Les applications J2EE nécessitent un serveur d'application web J2EE (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.