

# TD n°3 Java EE

- [Sujet TD n°3](#)
- [Archive Web à importer dans Eclipse](#)
- [Librairie Jar du projet en mode console](#)
- [Zip du projet en mode console](#)
- [Documentation des classes Web](#)
- [Documentation des classes du mode console](#)

## Classes

### Servlet Controller

- Créer la servlet **MainController** dans le package **web.controller**

```
/**
 * Servlet implementation class MainController
 */
@WebServlet("*.do")
public class MainController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Retourne le flux de sortie (de quoi écrire sur la page)
     * @param response réponse HTTP
     * @return le flux de sortie out
     * @throws IOException
     */
    private PrintWriter getOut(HttpServletResponse response) throws IOException{
        response.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        return out;
    }
    /**
     * Test la méthode de la requête
     * @param request
     * @return vrai si la méthode est POST
     */
    private boolean isPost(HttpServletRequest request){
        return request.getMethod().equalsIgnoreCase("post");
    }
    /**
     * Retourne la session active
     * @param request requête HTTP
     * @return la session en cours
     */
    private HttpSession getSession(HttpServletRequest request){
        return request.getSession();
    }
    /**
     * Retourne l'action sollicitée par la requête
     */
}
```

```
* @param request Requête Http
* @return action demandée
*/
private String getAction(HttpServletRequest request){
    String result="";
    String[] parts=request.getRequestURI().split("/");
    if(parts.length>0)
        result=parts[parts.length-1];
    if(request.getContextPath().contains(result))
        result="index";
    result=result.replace(".do","");
    return result;
}
/**
 * Charge une vue (jsp située dans le dossier protégé WEB-INF)
 * @param viewName nom de la vue
 * @param request Requête Http
 * @param response Réponse HTTP
 * @throws ServletException
 * @throws IOException
 */
private void loadView(String viewName,HttpServletRequest
request,HttpServletResponse response){
    try {
        request.getRequestDispatcher("WEB-INF/"+viewName).include(request,
response);
    } catch (ServletException | IOException e) {
        e.printStackTrace();
    }
    System.out.println("redirection vers WEB-INF/"+viewName);
}
/**
 * Exécute l'action actionName en appelant la méthode du même nom de la servlet
 * @param actionName nom de l'action à exécuter
 * @param request requête HTTP
 * @param response réponse HTTP
 * @return Vrai si l'action a été trouvée
 */
private boolean executeAction(String actionName,HttpServletRequest request,
HttpServletResponse response){
    Method method=null;
    boolean result=true;
    try {
        method=this.getClass().getMethod(actionName,
HttpServletRequest.class,HttpServletResponse.class);
    } catch (NoSuchMethodException | SecurityException e) {
        System.out.println("Action "+actionName+" inexistante");
        result=false;
    }
    if(method!=null){
        try {
            method.invoke(this, request,response);
        } catch (IllegalAccessException | IllegalArgumentException |
InvocationTargetException e) {
            System.out.println("Erreur d'exécution de l'action "+actionName +"
: "+e.getMessage());
        }
    }
}
```

```

        }
    }
    return result;
}
/**
 * @see HttpServlet#HttpServlet()
 */
public MainController() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    String action=getAction(request);
    if(action!=null && !"".equals(action)){
        if(!executeAction(action, request, response)){
            //TODO Implémenter si action inexistante
        }
    }
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    doGet(request, response);
}
public void index(HttpServletRequest request,HttpServletResponse response){
    loadView("index.jsp", request, response);
}
//TODO Autres actions à implémenter
}

```

## Exemple d'implémentation

Gestion de la suppression d'un utilisateur

### Vue frmDeleteUser

```

<form name="frmDelete" id="frmDelete">
    <%
        SessionApp sessionApp=Utils.getSessionApp(session);
        out.print(ConvertToHtml.mapToSelect(sessionApp.getApplication().getLesUtilisateurs(
), "txtLogin",10));
    %>

```

```
</form>  
<input type="button" id="btDeleteUser" class="btn" value="Supprimer">
```

### Méthode du contrôleur deleteUser

```
/**  
 * Gestion de la suppression d'un utilisateur  
 * @param request  
 * @param response  
 * @throws IOException  
 */  
public void deleteUser(HttpServletRequest request, HttpServletResponse  
response) throws IOException{  
    PrintWriter out=getOut(response);  
    if(isPost(request)){  
        SessionApp sessionApp=Utils.getSessionApp(request);  
        sessionApp.getApplication().setView(new HtmlView(out));  
        Commande cmd=new CmdDeleteUser(sessionApp);  
        cmd.addArg(request.getParameter("txtLogin"));  
        cmd.run();  
        out.print(JSFormsUtils.get("/deleteUser.do", "", "divFrmCommande"));  
    }else{  
        loadView("forms/frmDeleteUser.jsp", request, response);  
        out.print(JSFormsUtils.postFormAndBindTo("#btDeleteUser", "click",  
"/deleteUser.do", "frmDelete", "divResultCommande"));  
    }  
}
```

## Éléments de correction

### Script.js

Associer les items du menu aux actions en \*.do (méthodes de la classe **MainController**) en modifiant le fichier script.js :

```
run=function(obj){  
    cmd=obj.id;  
    new Forms.Ajax('divFrmCommande',cmd+'.do').get();  
    $("divResultCommande").innerHTML=obj.title;  
};
```

### main.jsp

Modifier les ids des éléments du menu, pour les faire correspondre aux actions (méthodes du contrôleur) :

```
<div id="grp-utilisateurs" class="categorie">Utilisateurs</div>
```

```

        <ul id="grp-child-utilisateurs">
            <li id="addUser" class="item" title="Entrer le login de
l'utilisateur à créer">Ajouter</li>
            <li id="updateUser" class="item" title="Sélectionner
l'utilisateur à modifier">Modifier</li>
            <li id="deleteUser" class="item" title="Sélectionner
l'utilisateur à supprimer">Supprimer</li>
            <li class="item">Affecter à ...</li>
            <li class="item">Liste</li>

```

L'item addUser pour ajouter un utilisateur correspond à la méthode du contrôleur :

```

public void addUser(HttpServletRequest request, HttpServletResponse response){
}

```

## Ajout d'un utilisateur

contrôleur **addUser**, en remplacement de **action/doAddUser.jsp**

```

/**
 * Ajout d'un utilisateur
 * @param request
 * @param response
 * @throws IOException
 */
public void addUser(HttpServletRequest request, HttpServletResponse response)
throws IOException{
    PrintWriter out=getOut(response);
    if(isPost(request)){
        SessionApp
sessionApp=(SessionApp)request.getSession().getAttribute("sessionApp");
        sessionApp.getApplication().setView(new HtmlView(out));
        Commande cmd=new CmdAddUser(sessionApp);
        cmd.addArg(request.getParameter("txtLogin"));
        cmd.run();
    }else{
        loadView("forms/frmAddUser", request, response);
        out.print(JSFormsUtils.postFormAndBindTo("#btAddUser", "click",
"addUser.do", "frm", "divResultCommande"));
    }
}
}

```

## Connexion de l'utilisateur

Contrôleur **login**, en remplacement de **action/doLogin.jsp**

```

/**
 * Connexion à l'application après soumission du formulaire frmLogin
 * @param request

```

```

    * @param response
    * @throws IOException
    */
    public void login(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        PrintWriter out=getOut(response);
        HttpSession session=getSession(request);
        SessionApp sessionApp=Utils.getSessionApp(session);
        Utils.setOut(sessionApp, out);
        Commande cmd=new CmdLogin(sessionApp);
        cmd.addArg(request.getParameter("txtLogin"));
        cmd.addArg(request.getParameter("txtPassword"));
        cmd.run();
        out.print("<br>");
        out.flush();
        if(!sessionApp.isUserLogIn())
            request.setAttribute("bad", true);
        else
            out.print(JSFormsUtils.get("ajaxMain.do", "", "divMain"));
        printMe(request, response);
    }

```

## Affichage user en cours

Contrôleur **printMe**, en remplacement de **action/doPrintMe.jsp**

```

/**
 * Affichage de l'utilisateur connecté ou du formulaire de connexion
 * si aucun utilisateur n'est connecté
 * @param request
 * @param response
 * @throws IOException
 */
public void printMe(HttpServletRequest request, HttpServletResponse response)
throws IOException{
    PrintWriter out=getOut(response);
    HttpSession session=getSession(request);
    SessionApp sessionApp=Utils.getSessionApp(session);

    if(sessionApp.isUserLogIn()){
        Utils.setOut(sessionApp, out);
        Commande cmd=new CmdMe(sessionApp);
        cmd.setSessionApp(sessionApp);
        cmd.run();
        out.print("<input class='btn' id='btDisconnect' type='button'
value='Déconnexion'>");
        out.print(JSFormsUtils.getAndBindTo("#btDisconnect", "click",
"disconnect.do", "", "divMain"));
    }else
        if(isPost(request) && request.getAttribute("bad")==null){
            login(request, response);
        }
        else{

```

```
        loadView("forms/frmLogin.jsp", request, response);
        out.print(JSFormsUtils.postFormAndBindTo("#btnSubmit", "click",
"login.do", "frmLogin", "divLogin"));
    }
}
```

## Restructuration de l'index

La page index est coupée en 3 et les 3 fichiers seront placés dans **WEB-INF/mainPages/**:

- un header : header.jsp
- la partie centrale : main.jsp
- la fin : footer.jsp

### header.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript" src="js/forms.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/sizzle.js"></script>
<link href="css/style.css" rel="stylesheet" media="all" type="text/css">
<title>Gestion des utilisateurs et des groupes</title>
</head>
<body onkeyup="Forms.Utils.onKeyUpFireEvent(event,13,'click',$('btnSubmit'));">
<div id="top"></div>
<div id="master">
    <div id="divMain">
        <!-- Fichier main.jsp -->
    </div>
</div>
</body>
</html>
```

### main.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<div id="divContent">
    <div id="divMenu">
        <div id="navigation">
            <div id="grp-utilisateurs" class="categorie">Utilisateurs</div>
            <ul id="grp-child-utilisateurs">
                <li id="addUser" class="item" title="Entrer le login de
l'utilisateur à créer">Ajouter</li>
                <li id="updateUser" class="item" title="Sélectionner
l'utilisateur à modifier">Modifier</li>
                <li id="deleteUser" class="item" title="Sélectionner
l'utilisateur à supprimer">Supprimer</li>
            </ul>
        </div>
    </div>
</div>
```

```

        <li id="moveTo" class="item" title="Affecter un utilisateur à
un groupe">Affecter à ...</li>
        <li id="listUsers" class="item" title="Afficher la liste des
utilisateurs">Liste</li>
    </ul>
    <div id="grp-groupes" class="categorie">Groupes</div>
    <ul id="grp-child-groupes">
        <li id="addGroup" class="item" title="Entrer le libellé du
groupe à créer">Ajouter</li>
        <li id="updateGroup" class="item" title="Sélectionner le groupe
à modifier">Modifier</li>
        <li id="deleteGroup" class="item" title="Sélectionner le groupe
à supprimer">Supprimer</li>
        <li id="listGroups" class="item" title="Afficher la liste des
groupes">Liste</li>
    </ul>
</div>
</div>
<div id="divCommande">
    <div id="divFrmCommande"></div>
    <div id="divBorderResultCommande">
        <div id="divResultCommande">Sélectionner un élément...</div>
    </div>
</div>
</div>

```

## footer.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!-- Fermeture de la divMain -->
</div>
</div>
</body>
</html>

```

## Contrôleur index

Le contrôleur index correspond à la page d'accueil et fait les traitements qui se faisaient avant dans la page **main.jsp** : Il utilise le contrôleur **ajaxMain**, dont le rôle est d'assurer l'affichage du menu si l'utilisateur est connecté, dans la div **divMain** :

```

/**
 * Affichage du menu et des fonctionnalités si l'utilisateur est connecté<br>
 * appel possible via ajax
 * @param request
 * @param response
 * @throws IOException
 */
public void ajaxMain(HttpServletRequest request,HttpServletResponse response)

```



```
throws IOException{
    PrintWriter out=getOut(response);
    out.print("<div id='divActiveUser'>");
    printMe(request, response);
    out.print("</div>");
    if(isLogIn(request)){
        loadView("mainPages/main.jsp", request, response);
        out.print(JSFormsUtils.onDOMReady("(new Forms.Accordion($('grp-child-
utilisateurs'),5,24)).attach($('grp-utilisateurs'));" +
        "(new Forms.Accordion($('grp-child-
groupes'),5,24)).attach($('grp-groupes'));" +
        "(new $selector('#navigation
.item','click',false,0,run,{backgroundColor:'#E8CA7A',fontWeight:'bold'})");"));
    }
}
```

```
loadView("mainPages/header.jsp", request, response);
ajaxMain(request, response);
loadView("mainPages/footer.jsp", request, response);
}
```

## Modification d'utilisateur

La modification se fait en trois étapes :

1. Choix de l'utilisateur à modifier (avec affichage de la vue **frmChoiceUser.jsp**)
2. Affichage de l'utilisateur et modification de ses attributs dans **frmUpdateUser.jsp**
3. Post du formulaire et modification de l'objet Utilisateur

## Vues

```
<%@page import="web.gui.ConvertToHtml"%>
<%@page import="web.technics.Utills"%>
<%@page import="net.bo.SessionApp"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<form name="frmUserChoice" id="frmUserChoice">
    <%
        SessionApp sessionApp=Utills.getSessionApp(session);
        out.print(ConvertToHtml.mapToSelect(sessionApp.getApplication().getLesUtilisateurs(
), "choiceLogin",10));
    %>
</form>
<input type="button" id="btUserChoice" class="btn" value="Modifier...">
<div id="divUserToUpdate">
</div>
```

```

<%@page import="net.bo.Utilisateur"%>
<%@page import="web.technics.JSFormsUtils"%>
<%@page import="static web.technics.WRequest.*" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%Utilisateur u=GETAttr(request, "user", new Utilisateur(""));%>
<fieldset>
<legend>Modifier un Utilisateur :</legend>
<form name="frmUpdateUser" id="frmUpdateUser">
    <input type="hidden" id="originalLogin" name="originalLogin"
value="<%=u.getName()%>">
    <label>Login: <input type="text" id="login" name="login"
value="<%=u.getName()%>"></label>
    <label>Nom :<input type="text" id="firstName" name="firstName"
value=<%=u.getFirstName()%>"></label>
    <label>Prénom :<input type="text" id="lastName" name="lastName"
value="<%=u.getLastName()%>"></label>

</form>
<input type="button" class="btn" value="Modifier l'utilisateur" id="btUpdateUser">
</fieldset>

```

## Contrôleur updateUser

```

    public void updateUser(HttpServletRequest request, HttpServletResponse
response) throws IOException{
        PrintWriter out=getOut(response);
        if(isPost(request)){
            if(GET(request,"choiceLogin")!=null){
                Utilisateur user= GatewaySession.getOneUser(request,
GET(request,"choiceLogin"));
                request.setAttribute("user", user);
                loadView("forms/frmUpdateUser.jsp", request, response);
                out.print(JSFormsUtils.postFormAndBindTo("#btUpdateUser", "click",
"updateUser.do", "frmUpdateUser","divResultCommande"));
            }else{
                if(GET(request,"login")!=null &&
GET(request,"originalLogin")!=null){
                    Utilisateur user= GatewaySession.getOneUser(request,
GET(request,"originalLogin"));
                    user.setName(GET(request,"login"));
                    user.setFirstName(GET(request, "firstName"));
                    user.setLastName(GET(request, "lastName"));
                    out.print("Utilisateur "+user+" modifié");
                    out.print(JSFormsUtils.setHtml("divUserToUpdate"));
                    out.print(JSFormsUtils.get("updateUser.do", "",
"divFrmCommande"));
                }
            }
        }else{
            loadView("forms/frmChoiceUser.jsp", request, response);
            out.print(JSFormsUtils.postFormAndBindTo("#btUserChoice", "click",
"updateUser.do", "frmUserChoice", "divUserToUpdate"));
        }
    }

```

```
    }  
}
```

## Utilitaires

### Dans MainController

Pour simplifier le test sur l'authentification :

```
/**  
 * Retourne vrai si l'utilisateur est logué  
 */  
protected boolean isLogIn(HttpServletRequest request){  
    SessionApp sessionApp=Utils.getSessionApp(request);  
    return sessionApp.isUserLogIn();  
}
```

### WRequest

Pour faciliter la récupération d'attributs ou de paramètres de la requête :

```
package web.technics;  
  
import javax.servlet.http.HttpServletRequest;  
  
public class WRequest {  
    /**  
     * Retourne le paramètre de nom parameterName de la requête  
     * @param request requête HTTP  
     * @param parameterName nom du paramètre  
     * @return  
     */  
    public static String GET(HttpServletRequest request,String parameterName){  
        return request.getParameter(parameterName);  
    }  
    /**  
     * Retourne le paramètre de nom parameterName de la requête et retourne  
     * defaultValue si le paramètre n'est pas trouvé  
     * @param request requête HTTP  
     * @param parameterName nom du paramètre  
     * @param defaultValue valeur par défaut  
     * @return paramètre de la requête  
     */  
    @SuppressWarnings("unchecked")  
    public static <T> T GET(HttpServletRequest request,String parameterName,T  
defaultValue){  
        T result;  
        String p=request.getParameter(parameterName);  
        if(p==null)
```

```
        result= defaultValue;
    else{
        try{
            result=(T)request.getParameter(parameterName);
        }catch(Exception e){
            result= defaultValue;
        }
    }
    return result;
}
/**
 * Retourne l'attribut de nom attributeName de la requête
 * @param request requête HTTP
 * @param attributeName nom de l'attribut
 * @return
 */
public static Object GETAttr(HttpServletRequest request,String attributeName){
    return request.getAttribute(attributeName);
}
/**
 * Retourne l'attribut de nom attributeName de la requête et retourne
defaultValue si l'attribut n'est pas trouvé
 * @param request requête HTTP
 * @param attributeName nom de l'attribut
 * @param defaultValue valeur par défaut
 * @return
 */
@SuppressWarnings("unchecked")
public static <T> T GETAttr(HttpServletRequest request,String attributeName,T
defaultValue){
    T result=(T)request.getAttribute(attributeName);
    if(result==null)
        result= defaultValue;
    return result;
}
}
```

From:  
<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:  
<http://slamwiki2.kobject.net/javaee/td3?rev=1385141576>

Last update: **2019/08/31 14:42**

