

Travaux Pratique de la programmation WEB : JAVA/J2EE

JSP et Servlets

Partie : 1

Objectifs :

L'objectif de cette première partie de notre TP est :

- Savoir utiliser une Template HTML déjà prête et de la préparer pour votre projet J2EE.
- Travailler avec un projet existant
- Créer et préparer la base de données et les tables nécessaires pour votre TP
- Modéliser le problème avec les classes POJO et DAO
- Programmer le cas d'utilisation d'authentification avec des identifiants
- Programmer le cas d'utilisation de déconnexion
- Faire passer des informations entre les servlets et les pages JSP
- Créer et utiliser des sessions
- Invalidier une session

Description

Nous allons dans cette première partie de notre projet « Etape par Etape », importer un projet partiellement créé, ce projet est à partir d'une Template déjà prête téléchargée depuis «<http://www.html5webtemplates.co.uk> » (vous trouverez le code original de la Template sous le fichier zip « textured_blue », il est adapter puis intégrer dans un projet JAVA Web que vous devez importer sur votre espace de travail sous l'IDE « eclipse »).

Prérequis :

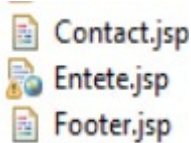
Pour commencer cette première partie vous devez importer le projet « Tutorialstartup » dans le fichier RAR « TutoStepByStep P1et P2 », ensuite vérifier si vous avez le driver « Mysql » sur le projet sinon il faut le télécharger et l'ajouter au « BuildPath » de votre projet.

Notez bien qu'il faut avoir un SGBDR installé, MySQL est recommandé pour débiter, sinon vous pouvez utiliser un de votre choix. En suivant les étapes de ce tutoriel, faites attention au login et mot de passe utiliser pour accéder au serveur SGBDR. Et finalement, pour mener à bien et à termes ce tutoriel, il est

recommandé de respecter les étapes à la lettre.

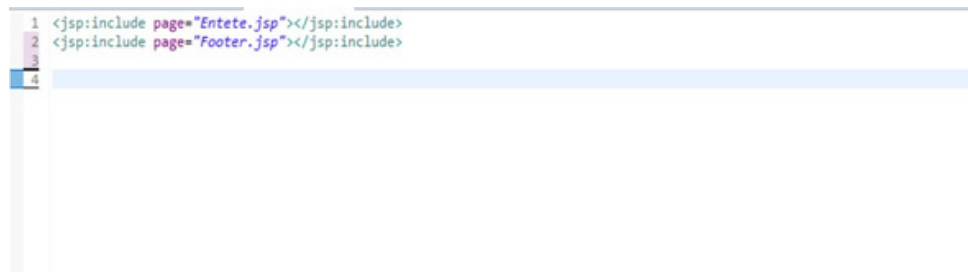
PARTIE I : Développement du processus d'authentification

1. Vous allez ouvrir le projet nommé «Tutorialstartup » et essayer d'ouvrir les pages suivantes :



Exécuter chaque page par la suite pour voir son contenu.

2. Créer une nouvelle page JSP nommée index.jsp
3. Modifier le contenu de la page index comme suite :



4. La page footer doit contenir :

```
<div id="footer">
  Copyright &copy; textured_blue | <a
href="http://validator.w3.org/check?uri=referer">HTML5</a> | <a href="http://jigsaw.w3.org/css-
validator/check/referer">CSS</a> | <a href="http://www.html5webtemplates.co.uk">Free CSS
Templates</a>
</div>
</body>
</html>
```

5. La page entete doit avoir le code suiivant :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<meta http-equiv="content-type" content="text/html; charset=windows-1252" />
```

```

<link rel="stylesheet" type="text/css" href="style/style.css" title="style" />
</head>
<body>
  <div id="main">
    <div id="header">
      <div id="logo">
        <div id="logo_text">
          <!-- class="logo_colour", allows you to change the colour of the text -->
          <h1><a href="index.html">Tutoriel<span class="logo_colour">J2EE</span></a></h1>
          <h2>TP etape par etape</h2>
        </div>
        <div id="logintext">
          Se connecter | Déconnexion
        </div>
      </div>
      <div id="menubar">
        <ul id="menu">
          <!-- put class="selected" in the li tag for the selected page - to highlight which page you're on -->
          <li class="selected"><a href="index.html">Login</a></li>
          <li><a href="examples.html">Gérer les employees</a></li>
          <li><a href="page.html">Ajouter un employee</a></li>
          <li><a href="another_page.html">Afficher le détail</a></li>
          <li><a href="contact.html">Creer Compte</a></li>
        </ul>
      </div>
      <div id="site_content">
        <div class="sidebar">
          <!-- La barre des news a droite -->
          <h3>Dernier Ajouté</h3>
          <h4>Employee Recent</h4>
          <h5>La date d'aujourd'hui</h5>
          <p>Ici nous allons ajouter le salarié le plus recent ajouter au niveau de notre base de
données<br /><a href="#">Plus ...</a></p>
          <p></p>
          <h4>Nouveau Département Ajouté</h4>
          <h5>La date</h5>
          <p>Ici nous allons mettre le departement avec le plus grand nombre des employees<br /><a
href="#">Plus ...</a></p>
          <h3>Liste des liens</h3>
          <ul>
            <li><a href="#">link 1</a></li>
            <li><a href="#">link 2</a></li>
            <li><a href="#">link 3</a></li>
            <li><a href="#">link 4</a></li>
          </ul>
          <h3>Search</h3>
          <form method="post" action="#" id="search_form">
            <p>

```

```

    <input class="search" type="text" name="search_field" value="Enter keywords....." />
    <input name="search" type="image" style="border: 0; margin: 0 0 -9px 5px;"
src="style/search.png" alt="Search" title="Search" />
  </p>
</form>
</div>

```

6. Exécuter encore la page index.jsp, qu'est ce que vous remarquez ?

7. La position du pied de page est un peu au milieu de la page, ouvrez le fichier « Style.cee » et modifiez le code comme ci-dessous:

```

193
194 #content
195 { text-align: left;
196   width: 595px;
197   padding: 0;
198   height: 800px }

```

8. Modifier ensuite la page index.jsp, le code doit etre comme suite :

```

<jsp:include page="Entete.jsp"></jsp:include>
<div id="content">

</div>

<jsp:include page="Footer.jsp"></jsp:include>

```

9. Executer encore la page index.jsp, qu'est ce que vous remarquez ?

Dans cette partie nous avons simplement créer et configurer de façon basique notre template pour qu'il soit adapter au developpement web dynamique avec JSP/Servelet.

La partie suivante consiste à paramettrer l'authentification.

Pour cette partie, nous aurons besoin de créer une une interface dédiée et spécifique, ensuite nous devons préparer une table dans la base de données ou on va stocker les login et le mot de passe de chaque employee, et à la fin nous devons créés les classes nécessaires pour faire ce traitement.

10. Créer une nouvelle page nommée : login.jsp avec le code suivant :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<meta http-equiv="content-type" content="text/html; charset=windows-1252" />
<link rel="stylesheet" type="text/css" href="style/style.css" title="style" />
</head>
<body>
  <div id="main">

```

```
<div id="header">
  <div id="logo">
    <div id="logo_text">
      <!-- class="logo_colour", allows you to change the colour of the text -->
      <h1><a href="index.html">Tutoriel<span class="logo_colour">J2EE</span></a></h1>
      <h2>TP etape par etape</h2>
    </div>
    <div id="logintext">
      Se connecter | Déconnexion
    </div>
  </div>
  <div id="menubar">
    <ul id="menu">
      <!-- put class="selected" in the li tag for the selected page - to highlight which page you're on --
      <li class="selected"><a href="Login.jsp">LOGIN</a></li>
    </ul>
  </div>
</div>

  <div id="content" style="height: 400px">
    <!-- insert the page content here -->
    <h1>Login</h1>
    <p>Veuillez saisir votre Nom d'utilisateur et votre Mot de passe</p>
    <form action="#" method="post">
      <div class="form_settings">
        <p><span>Nom d'utilisateur</span><input class="contact" type="text" name="your_name"
value="" /></p>
        <p><span>Password</span><input class="contact" type="text" name="your_email" value=""
/></p>
        <p style="padding-top: 15px"><span>&nbsp;</span><input class="submit" type="submit"
name="contact_submitted" value="Se Connecter" /></p>
      </div>
    </form>
  </div>
</div>
</body>
</html>
```

11. Exécuter la fenêtre pour voir sa forme et son contenu.
12. Créer une nouvelle base de données nommée « gestionemployees » dans votre SGBR (dans ce tutoriel je vais utiliser le SGBDR MYSQL), on utilisant le script suivant :

```
CREATE SCHEMA `gestionemployees`;
```
13. Dans cette base de données vous allez créer une nouvelle table nommée « logins », avec le script (MYSQL) suivant :

```
CREATE TABLE `logins` (
```

```

`id` int NOT NULL AUTO_INCREMENT,
`login` varchar(45) NOT NULL,
`password` varchar(45) NOT NULL,
`idemp` int DEFAULT NULL,
`role` varchar(45) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
  
```

14. Insérer ensuite une première ligne directement sur cette table pour test en utilisant le script suivant :

```
INSERT INTO `gestionemployees`.`logins` (`login`, `password`, `idemp`) VALUES ('test', 'test', '100');
```

15. En utilisant votre IDE JAVA (eclipse ou NetBeans) vous allez créer les packages suivants :



16. Créer ensuite une la première classe nommée « CredentialsC » sous le package Entities, avec les meme attributs que la table « logins » crée précédement, et les constructeurs nécessaires et les getters and setters.

```

package Entities;

public class credentialsC {

    private int idcredential, idemp;
    private String login, password, role;

    ....

}

Optionnellement, en cas de besoin vous pouvez sur définir la fonction « ToString() » avec le code ci-
apres :
@Override
public String toString() {
    return "credentialsC [idcredential=" + idcredential + ", idemp=" + idemp + ", login=" +
    login + ", password="
  
```

```
        + password + ", role=" + role + "]"");
    }
}
```

17. Sous le package « tools » créer une classe nommée « MySqlConnection », qui nous servira et générer une « connection », son code sera comme suite (modifier le login et le mot de passe de votre serveur :

```
package tools;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

import org.apache.tomcat.util.bcel.classfile.ClassElementValue;

public class MySqlConnection {

    public static Connection con=null;

    public static Connection getconnectionmysql() {

        try {
            Class.forName("com.mysql.jdbc.Driver");

            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/gestionemployees","votre
login","votre mot de passe");

        } catch (ClassNotFoundException | SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return con;
    }

    public static void main(String[] args) {

        Connection c=getconnectionmysql();
        System.out.println("salamo ");

    }
}
```

18. Créer une nouvelle classe nommée « CredentialsDAO » sous le package dao, son code est le suivant :

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import Entities.credentialsC;
import tools.MySqlConnection;

public class CredentialsDAO {

    public static credentialsC checkcredentials(String l, String p) {
        credentialsC res= new credentialsC();

        String sql= "select * from logins where login =? and password =?";
        PreparedStatement ps=null;
        ResultSet rs= null;
        Connection con=MySqlConnection.getconnectionmysql();

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, l);
            ps.setString(2, p);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            System.out.println("erreur avec PS");
        }

        try {
            rs= ps.executeQuery();
            if(rs.next())
            {
                res.setldcredential(rs.getInt(1));
                res.setLogin(rs.getString(2));
                res.setPassword(rs.getString(3));
                res.setldemp(rs.getInt(4));
                res.setRole(rs.getString(5));
            }
            else
                return null;
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            System.out.println("erreur avec RS");
        }
    }
}
```



```

        return null;
    }

    return res;
}

```

```

}

```

19. Modifier le code de la partie « formulaire » de la page « login.jsp » pour qu'il soit identique à ce code :

```

<form action="Logincontroler" method="post">

    <div class="form_settings">

        <p><span>Nom d'utilisateur</span><input class="contact" type="text" name="tlogin"
value="" /></p>

        <p><span>Password</span><input class="contact" type="password" name="tpassword"
value="" /></p>

        <p style="padding-top: 15px"><span>&nbsp;</span><input class="submit" type="submit"
name="contact_submitted" value="Se Connecter" /></p>

    </div>

</form>

```

20. Nous allons maintenant gérer l'action de la connexion d'un utilisateur, pour cela nous allons créer un nouveau package nommée « controlers » où on va mettre nos servlets, ensuite créer la première servlet nommée « Logincontroler.java », qui n'implémente que la méthode « POST », qui va faire le traitement suivant :
- Recuperer les deux valeurs de « tlogin » et « tpassword » saisie dans la page « Login.jsp »
 - Vérification et récupération des informations de l'authentification de la base de données en utilisant la méthode convenable de la classe « CredentialsDAO »
 - Si le login et le mot de passe sont corrects vous allez rediriger toutes les informations en utilisant un objet CredentialsC à l'aide d'une session à la page « index.jsp » si non vous allez rediriger l'utilisateur à la même page Login.jsp avec un attribut ERREUR.
 request.setAttribute("ERROR", "Login ou Mot de passe incorrecte");
21. Exécuter la page « Login.jsp » pour tester avec un login et mot de passe corrects et une deuxième tentative avec deux autres incorrects.
22. Nous allons maintenant paramétrer l'affichage pour que la page d'accueil affiche le login de l'utilisateur et le lien de « deconnexion », et dans le cas où il y a une erreur nous allons afficher un message sur la page « Login.jsp » pour montrer que login ou le mot de passe est incorrect.

Pour répondre à la dernière spécification, vous allez modifier le code du formulaire de la page « Login.jsp » comme suite (faites attention au nom des paramètre et attribut si vous avez utilisé d'autres):

```

<form action="Logincontroler" method="post">
  <div class="form_settings">
    <p><span>Nom d'utilisateur</span><input class="contact" type="text" name="tlogin"
value="" /></p>
    <p><span>Password</span><input class="contact" type="password"
name="tpassword" value="" /></p>
    <p style="padding-top: 15px"><span>&nbsp;</span><input class="submit"
type="submit" name="contact_submitted" value="Se Connecter" /></p>
    <div style="color: red;">
      <% if(request.getAttribute("ERROR")!=null)
      {
      %>
      <p><%=request.getAttribute("ERROR") %></p>
      <%
      }
      %>
    </div>
  </div>
</form>

```

23. Nous allons ensuite modifier le projet pour modifier l'affichage au niveau de l'entête de nos pages une fois l'utilisateur est bien authentifié, nos pages doivent être capables de rediriger l'utilisateur à la page d'authentification si il essaye d'accéder aux pages de notre site directement sans authentification, dans le cas contraire en haut de la page nous devons afficher [Bienvenu « LoginEmployee »] plus un lien qui permet de déconnecter l'utilisateur. Vous allez dans la page entête et vous allez la modifier comme suite :

```

15 <div id="main">
16   <div id="header">
17     <div id="logo">
18       <div id="logo_text">
19         <!-- class="logo_colour", allows you to change the colour of the text -->
20         <h1><a href="index.html">Tutoriel<span class="logo_colour">J2EE</span></a></h1>
21         <h2>TP etape par etape</h2>
22       </div>
23     <div id="logintext">
24       <% if(session.getAttribute("UTILISATEUR")==null)
25       {
26         response.sendRedirect("Login.jsp");
27       }
28       else
29       {
30         credentialsC user=(credentialsC)session.getAttribute("UTILISATEUR");
31       }
32       <%>
33       Bienvenue <%=user.getLogin() %> | <a href="Logoutcontroler">Déconnexion</a>
34     </div>
35   </div>
36   <div id="menubar">
37     <ul id="menu">
38       <!-- put class="selected" in the li tag for the selected page - to highlight which page you're on -->
39       <li class="selected"><a href="index.html">Home</a></li>

```

24. Exécuter le programme et vérifier le fonctionnement en utilisant un login/mot de passe incorrect et ensuite utiliser le couple test/test ; essayez encore d'exécuter la page « Index.jsp » directement, le programme doit vous rediriger vers la page d'authentification
25. Nous allons maintenant programmer le scénario de la déconnexion, et pour cela vous allez créer une servlet nommée « Logoutcontroler.java » sous le package « controlers » qui implémente la méthode doGet() pour :
 - Invalider la session
 - Rediriger la navigation vers la page d'authentification

**"Quelle est la capitale de la planète
 Zorglub ?"**
Réponse: Zorglub City.