

## Travaux Pratique de la programmation WEB : JAVA/J2EE

### JSP et Servlets

#### Partie : 2

#### **Objectifs :**

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

- Résoudre le problème du processus de la validation d'un compte utilisateur
- Faire passer des paramètres et des attributs entre les pages JSP et Servlet
- Modifier une table SQL existante et ajouter des contraintes
- Savoir implémenter les méthodes dans une classe existante
- Programmer la méthode de l'insertion et de la modification dans une classe DAO
- Adapter une classe POJO a un nouveau besoin
- Profiter de la modélisation par objectif pour modifier un projet Java/JEE
- Affichage conditionnel du contenu d'une page JSP, dans notre cas « le rôle de l'utilisateur »
- Utiliser une seule servlet comme contrôleur a plusieurs requêtes Http

#### **Prérequis :**

Cette partie, et la suite d'une première, si vous venez d'avoir ce fichier, et vous êtes débutants, il est fortement recommandé de commencer par la première partie, sinon, il est possible d'enchaîner avec cette partie en se basant sur le projet « TutorialdesignV1 » qui se trouve sur le fichier Rar nommé « TutoStepByStep P1et P2 ».

Il faut avoir un SGBDR installé et d'ajouter le driver adéquat à votre projet, ensuite une base de données créé et avec au moins une table nommée « Logins » les champs et les types vous pouvez utiliser le script suivant :

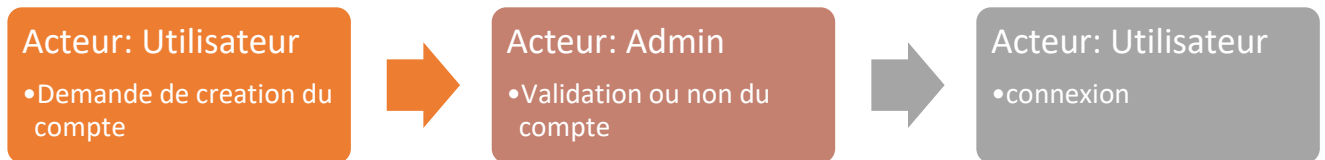
```
CREATE DATABASE `gestionemployees` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;

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,
  `etat` varchar(45) NOT NULL DEFAULT 'Inactif',
  PRIMARY KEY (`id`),
  CONSTRAINT `etatcc` CHECK ((`etat` in (_utf8mb4'Inactif',_utf8mb4'Actif',_utf8mb4'Bloque')))
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
'1', 'test', 'test', '100', 'admin', 'Actif'  
'2', 'user', 'user', '200', 'user', 'Actif'  
'3', 'test2', 'test2', '300', NULL, 'Inactif'  
'4', 'yyyyy', 'yyyy', '0', NULL, 'Inactif'  
'10', 'test3', 'test3', '153', 'Utilisateur', 'Bloqué'
```

## PARTIE II : Développement du processus de création du compte utilisateur

Dans cette partie nous allons préparer le processus de la création du compte, et dans ce tutoriel nous allons considérer que la création du compte se fait avec le scénario suivant :



Pour répondre à ce besoin nous aurons besoin de :

- Modifier la structure de notre table « Logins » pour inclure un nouveau champ qui représente l'état du compte, on peut préciser trois états :
  - **Inactif** : par défaut après la demande de l'utilisateur
  - **Actif** : activé par l'administrateur
  - **Bloqué** : explicitement bloqué par l'utilisateur
- Créer une page JSP avec un formulaire pour permettre à l'utilisateur de demander la création de son compte.
- Créer une servlet qui va permettre de recevoir la requête et invoquer la méthode DAO adéquate.
- Modifier la classe DAO correspondante pour permettre l'insertion des données saisies par l'utilisateur.
- Créer une page JSP qui va permettre à l'administrateur de valider les comptes inactifs.
- Créer la servlet adéquate pour gérer cette requête, et invoquer la(es) méthode(s) DAO nécessaires.
- Modifier la class DAO pour programmer la(es) méthode(s) nécessaires

N.B : Si vous venez de commencer ce tutoriel à partir de cette Partie, ouvrez le projet « TutorialAuthentification » sur le fichier compressé « TutorialAuthentification.rar »

- 1. Modifier la structure de la table « logins » on ajoutant un nouveau champs nommé « etat » qui accepte trois valeur « inactif, actif, bloqué » avec la requête suivante :**

```

ALTER TABLE `gestionemployees`.`logins`
ADD COLUMN `etat` VARCHAR(45) NOT NULL DEFAULT 'Inactif' AFTER `role`,
ADD CONSTRAINT etatcc CHECK (etat in ("Inactif","Actif","Bloque"));

UPDATE `gestionemployees`.`logins` SET `etat` = 'Actif' WHERE (`id` = '1');
UPDATE `gestionemployees`.`logins` SET `etat` = 'Actif' WHERE (`id` = '2');
  
```

- 2. Créer une nouvelle Page JSP nommée : « Nouveaucompte.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 >
<a href="<%=application.getContextPath()%>/Nouveaucompte.jsp" >Cr  er Compte</a>
</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><a href="Login.jsp">LOGIN</a></li>
<li class="selected"><a href="Login.jsp">Cr  ation d'un nouveau compte</a></li>
</ul>
</div>
</div>
<div id="content" style="height: 400px">
<!-- insert the page content here -->
<h1>Contact Us</h1>
<p>Below is an example of how a contact form might look with this template:</p>
<form action="LoginManagement?Action=creer" method="post">
<div class="form_settings">
<p><span>Login</span><input class="contact" type="text" name="your_name" value=""
/></p>
<p><span>Mot de passe</span><input class="contact" type="password"
name="your_password" value="" /></p>
<p><span>Id Employee</span><input class="contact" type="text" name="your_empid"
value="" /></p>
<p style="padding-top: 15px">
<span>&nbsp;</span>

```

```

    <input class="submit" type="submit" name="contact_submitted" value="Creer Compte" />
    <span>&nbsp;</span>
    <input class="submit" type="reset" value="Annuler"/>

  </p>
</div>
</form>
<p><br /><br />NOTE: Votre Compte sera par default inactif apres la validation de
l'administrateur du systeme</p>
</div>
<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>

```

3. Modifier la classe « credentialsC » pour qu'elle prenne en considération le nouvel attribut « etat »
4. Implémenter ensuite une nouvelle méthode qui insert les informations saisies par l'utilisateur dans la table « login », le prototype de cette méthode est :
  - public static Integer insertNewCredentials(credentialsC c)

5. modifier la page "Login .JSP" pour remplacer le code

```

20 <div id="logintext">
21   Se connecter
22 </div>

```

par le code suivant

```

20 <div >
21   <a href="<%=application.getContextPath()%>/Nouveauecompte.jsp" >Créer Compte</a>
22 </div>
23 </div>

```

6. Vous allez maintenant créer une nouvelle servlet dans la package « Controlers » nommée « LoginManagement.java » qui implémente la méthode poste comme suite :

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    if(request.getParameter("Action").equals("creer"))
    {
        String erro=null;
        try {
            Integer.parseInt(request.getParameter("your_empid"));
        } catch (NumberFormatException | NullPointerException e) {
            // TODO: handle exception
            erro="emp ID doit etre numerique";
        }
        credentialsC = new credentialsC();
        c.setLogin(request.getParameter("your_name"));
        c.setPassword(request.getParameter("your_password"));
        c.setIdemp(Integer.parseInt(request.getParameter("your_empid")));
        int ret=CredentialsDAO.insertNewCredentials(c);
        if(ret>0)
        {
            request.setAttribute("login", c.getLogin());
            request.getRequestDispatcher("confirmationCreation.jsp").forward(request, response);
        }
    }
}
  
```

7. Créer ensuite une nouvelle page jsp nommée « confirmationCreation », pour afficher un message de confirmation à l'utilisateur si la demande est bien insérée. Cette page a 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 >
<a href="<%=application.getContextPath()%>/Nouveauecompte.jsp" >Créer Compte</a>
</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><a href="Login.jsp">LOGIN</a></li>
<li class="selected"><a href="Login.jsp">Création d'un nouveau compte</a></li>
</ul>
  
```

```

    </div>
  </div>

  <div id="content" style="height: 400px">
    <h3>Confirmation de la demande de création du compte</h3>
    <p>
      le compte avec le login <%=request.getAttribute("login") %> est en cours d'approbation par
      l'administrateur...
    </p>
  </div>

  <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>

```

8. Exécuter la page « NouveauCompte.jsp » et saisissez les informations demandées pour vérifier si le programme enregistre le compte sur la base de données et vous redirige vers la page de la confirmation.

Nous allons maintenant gérer le cas où il y a des problèmes pour réaliser l'insertion, entre autres, saisie d'un ID employée non entier.

9. Modifier le code de la servlet « loginmanagement » comme suite, et essayer de comprendre la logique :

```

package controllers;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import Entities.credentialsC;
import dao.CredentialsDAO;

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

```

```

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

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        if(request.getParameter("Action").equals("creer"))
        {
            String erro=null;
            credentialsC = new credentialsC();

            try {
                Integer.parseInt(request.getParameter("your_empid"));
            } catch (NumberFormatException e) {
                // TODO: handle exception
                request.setAttribute("errocreation", "votre demande avec le login" +
                    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

                request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
            }

            /***** Ou vous pouvez utiliser
            * if(!(request.getParameter("your_empid").matches("\\D")) ||
request.getParameter("your_empid").isBlank())
            {
                request.setAttribute("errocreation", "votre demande avec le login" +
                    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

                request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
            }

            *
            */

            if (request.getParameter("your_empid")==null ||
request.getParameter("your_password")==null || request.getParameter("your_name")== null){
                request.setAttribute("errocreation", "votre demande "
                    + "à échouée, vérifier que vous avez bien remplis le
formulaire");

                request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);

```



```

    }

    else if (request.getParameter("your_password").isBlank() ||
request.getParameter("your_name").isBlank())
    {
        request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
        "à échouée, vérifier que vous avez bien remplis le
formulaire");

        request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
    }
    c.setLogin(request.getParameter("your_name"));
    c.setPassword(request.getParameter("your_password"));
    c.setIdemp(Integer.parseInt(request.getParameter("your_empid")));

    int ret=CredentialsDAO.insertNewCredentials(c);
    if(ret>0)
    {
        request.setAttribute("login", c.getLogin());

        request.getRequestDispatcher("confirmationCreation.jsp").forward(request, response);
    }
    else
    {
        request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
        "à échouée, la requete n'a pas pu etre terminée");

        request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
    }

    }
}
}

```

**10. Ensuite modifier la page jsp « nouveaucompte » par le code suivant :**

```

49         </div>
50     </form>
51     <% if(request.getAttribute("errocreation")!=null){ %>
52
53         <p><br /><br />NOTE: Votre Compte sera par défaut inactif apres la validation de l'administrateur du systeme</p>
54     <%>
55     else {%>
56         <p style="color: red; font-style: italic;"><br /><br /> <%=request.getAttribute("errocreation") %> </p>
57     <%> %>
58     </div>

```

**11. Tester la création des comptes avec plusieurs scénarios erronés.**

Exécuter la page « Login.jsp » et essayer de se connecter avec le compte que vous venez de créer, est ce que ça marche ? Pourquoi ?

Nous devons faire des modifications pour gérer un tel cas, en suivant les instructions suivantes :

**12. Ajouter cette ligne dans l'emplacement adéquat dans la Classe « CredentialsDAO.java »**

```
res.setEtat(rs.getString(6));
```

**13. Notre programme jusqu'à maintenant affiche une erreur de login ou mot de passe incorrect sur la page login au cas où l'authentification à échouée, nous voudrions gérer même le cas où les identifiants sont corrects mais le compte n'est pas « Actif », pour cela ajouter le code suivant à la servlet « LoginConroler » dans l'endroit adéquat :**

```

if(!c.getEtat().equals("Actif"))
{
    request.setAttribute("ERROR", "Votre Compte est "+c.getEtat()+" Merci de contacter l'administrateur");
    request.getRequestDispatcher("Login.jsp").forward(request, response);
}

```

Nous allons maintenant programmer la deuxième étape de notre process qui est la validation des comptes par l'administrateur. La première étape à faire, on considère une page ou on cherche un login bien définis grâce a une barre de recherche.

**14. On va créer une nouvelle page nommée « Activer\_compte.jsp », qui n'est accessible que par les administrateurs, le code est le suivant :**

```

<%@ include file="Entete.jsp"%>
<div id="content">

    <form method="post" action="LoginManagement?Action=Activercompte"
id="search_form">
        <p>
            <input class="search" type="text" name="tlogin" value="Enter un Login....." />
            <input name="search" type="image" h style="border: 0; margin: 0 0 -9px 5px;"
src="style/search.png" alt="Search" title="Search" />
        </p>
    </form>

    <% if(request.getAttribute("message")!=null)
        {%>
            <p style="color: red;"><%=request.getAttribute("message") %>
            <%} %>

</div>

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

```

15. Nous aurons besoin d'implémenter une nouvelle méthode dans la classe « CredentialsDAO » qui va nous permettre d'extraire un login déterminé et non actif le code est ci-dessous :

```

public static credentialsC getInactifCredential(String vlogin)
{
    credentialsC res= new credentialsC();
    String sql= "select * from logins where login =? and (etat =? or etat = ?)";
    PreparedStatement ps=null;
    ResultSet rs= null;
    Connection con=MySQLConnection.getConnectionmysql();

    try {
        ps=con.prepareStatement(sql);
        ps.setString(1, vlogin);
        ps.setString(2, "Inactif");
        ps.setString(3, "Bloque");
    }

    catch (SQLException e) {

        // TODO Auto-generated catch block
        e.printStackTrace();
        System.out.println("probleme avec prepared statment");
    }

    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));
            res.setEtat(rs.getString(6));
        }
        else
            return null;
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        System.out.println("probleme lors de l'execution de la requete");
    }

    return res;
}
  
```

}

**16. Ensuite vous allez modifier la servlet « LoginManagement » pour quelle ressemble au code ci-après :**

```
package controlers;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import Entities.credentialsC;
import dao.CredentialsDAO;

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

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

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        //TRAITEMENT DE l'action CREER dun nouveau compte
        if(request.getParameter("Action").equals("creer"))
        {
            String erro=null;
            credentialsC c= new credentialsC();

            try {
                Integer.parseInt(request.getParameter("your_empid"));
            } catch (NumberFormatException e) {
```

```

// TODO: handle exception
request.setAttribute("errocreation", "votre demande avec le login" +
    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
}

/***** Ou vous pouvez utiliser
 * if(!(request.getParameter("your_empid").matches("\\D")) ||
request.getParameter("your_empid").isBlank())
{
    request.setAttribute("errocreation", "votre demande avec le login" +
    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
}

*
* */

if (request.getParameter("your_empid")==null ||
request.getParameter("your_password")==null || request.getParameter("your_name")== null){
    request.setAttribute("errocreation", "votre demande "
        + "à échouée, vérifier que vous avez bien remplis le
formulaire");

request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
}

else if (request.getParameter("your_password").isBlank() ||
request.getParameter("your_name").isBlank())
{
    request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
    "à échouée, vérifier que vous avez bien remplis le
formulaire");

request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
}
c.setLogin(request.getParameter("your_name"));
c.setPassword(request.getParameter("your_password"));
c.setIdemp(Integer.parseInt(request.getParameter("your_empid")));
  
```

```

        int ret=CredentialsDAO.insertNewCredentials(c);
        if(ret>0)
        {
            request.setAttribute("login", c.getLogin());

            request.getRequestDispatcher("confirmationCreation.jsp").forward(request, response);
        }
        else
        {
            request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
                                "à échouée, la requete n'a pas pu etre terminée");

            request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
        }
    }
    /*
    *
    *
    *
    */
    //TRAITEMENT DE l'action recherche dun compte Inactif
    else if(request.getParameter("Action").equals("Activercompte"))
    {

        if(CredentialsDAO.getInactifCredential(request.getParameter("tlogin"))==null)
        {
            request.setAttribute("message", "le Login
"+request.getParameter("tlogin")+" est introuvable ou déjà actif ");

            request.getRequestDispatcher("Activer_compte.jsp").forward(request, response);
        }
        else
        {
            credentialsC =
CredentialsDAO.getInactifCredential(request.getParameter("tlogin"));
            request.setAttribute("credential", c);

            request.getRequestDispatcher("Activer_compte.jsp").forward(request, response);
        }
    }
}

```

```
}

```

Exécuter et vérifier si la recherche ne génère pas d'erreur d'application, et elle retourne un message dans le cas où vous avez choisis un login déjà actif ou qui n'existe pas. Si vous saisissez un login qui n'est pas activé, l'application va rien faire, c'est l'objectif qu'on va réaliser durant les étapes suivantes :

**17. Modifier encore le code de la page « Activer\_compte.jsp » comme suite :**

```

<%@ include file="Entete.jsp"%>
<div id="content">

    <form method="post" action="LoginManagement?Action=Activercompte"
id="search_form">
    <p>
    <input class="search" type="text" name="tlogin" value="Enter un Login....." />
    <input name="search" type="image" h style="border: 0; margin: 0 0 -9px 5px;"
src="style/search.png" alt="Search" title="Search" />
    </p>
</form>

    <% if(request.getAttribute("message")!=null)
    {%>
        <p style="color: red;"><%=request.getAttribute("message") %>
    <%} else if(request.getAttribute("credential")!=null)
    {
        credentialsC c=(credentialsC)request.getAttribute("credential");
    <%>
    <form action="LoginManagement?Action=modifierinactifcompte" method="post">
    <div class="form_settings">
    <input class="contact" type="text" hidden="" name="tid" value="<%=c.getIdcredential()%>"
/></p>
    <p><span>Login</span><input class="contact" type="text" disabled="disabled"
name="tname" value="<%=c.getLogin() %>" /></p>
    <p><span>Id Employee</span><input class="contact" type="text" disabled="disabled"
name="tempid" value="<%=c.getIdemp() %>" /></p>
    <p>
    <span>Etat du Compte</span>
    <select name="tetat" style="border: 1px solid #E5E5DB;">
        <option value="Actif">Actif</option>
        <option value="Bloqué">Bloqué</option>
        <option value="Inactif" selected="selected">Inactif</option>
    </select>

```

```

    </p>

    <p>
    <span>Role</span>
    <select name="trole" style="border: 1px solid #E5E5DB;">
        <option value="Administrateur">Administrateur</option>
        <option value="Manager">Manager</option>
        <option value="Utilisateur"
selected="selected">Utilisateur</option>
    </select>

    </p>

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

    </p>
    </div>
    </form>
    <%}%>
</div>

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

Si vous essayez maintenant avec un login existant et inactif, la page va vous afficher l'information du login. Maintenant que nous avons les données du login, l'administrateur doit lui octroyer un rôle, et active ou non son compte.

**18. Nous devons modifier la classe « CredentialsDAO » pour implémenter une nouvelle méthode qui permet de modifier les informations d'un compte donné, la méthode est comme suite :**

```

public static Integer updatecredentials(credentialsC c) {

    PreparedStatement ps=null;
    Integer res=null;
    Connection con=MySQLConnection.getconnectionmysql();
    String sql= "UPDATE logins SET role = ?, "
        + "etat = ? WHERE id = ?";

    try {
        ps=con.prepareStatement(sql);
        ps.setString(1, c.getRole());
        ps.setString(2, c.getEtat());
    }
  
```



```

        ps.setInt(3, c.getIdcredential());

        res=ps.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        try {
            con.close();
        } catch (SQLException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
            System.out.println("probleme lors de la fermeture de la connexion");
        }
    }

    try {
        con.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        System.out.println("probleme lors de la fermeture de la connexion");
    }

    return res;
}

```

**19. Le code final aussi de la servlet « LoginManagement » est :**

```

package controlers;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import Entities.credentialsC;
import dao.CredentialsDAO;

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

```

```

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

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        //TRAITEMENT DE l'action CREER dun nouveau compte
        if(request.getParameter("Action").equals("creer"))
        {
            String erro=null;
            credentialsC c= new credentialsC();

            try {
                Integer.parseInt(request.getParameter("your_empid"));
            } catch (NumberFormatException e) {
                // TODO: handle exception
                request.setAttribute("errocreation", "votre demande avec le login" +
                    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

                request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
            }

            /***** Ou vous pouvez utiliser
            * if(!(request.getParameter("your_empid").matches("\\D")) ||
request.getParameter("your_empid").isBlank())
            {
                request.setAttribute("errocreation", "votre demande avec le login" +
                    "à échouée, vérifier que vous avez bien saisie un ID
employee est correct format");

                request.getRequestDispatcher("Nouveaucompte.jsp").forward(request, response);
            }

            *
            */

```

```

        if (request.getParameter("your_empid")==null ||
request.getParameter("your_password")==null || request.getParameter("your_name")== null){
            request.setAttribute("errocreation", "votre demande "
                + "à échouée, vérifier que vous avez bien remplis le
formulaire");

            request.getRequestDispatcher("Nouveaumontepage.jsp").forward(request, response);
        }

        else if (request.getParameter("your_password").isBlank() ||
request.getParameter("your_name").isBlank())
        {
            request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
                "à échouée, vérifier que vous avez bien remplis le
formulaire");

            request.getRequestDispatcher("Nouveaumontepage.jsp").forward(request, response);
        }
        c.setLogin(request.getParameter("your_name"));
        c.setPassword(request.getParameter("your_password"));
        c.setIdemp(Integer.parseInt(request.getParameter("your_empid")));

        int ret=CredentialsDAO.insertNewCredentials(c);
        if(ret>0)
        {
            request.setAttribute("login", c.getLogin());

            request.getRequestDispatcher("confirmationCreation.jsp").forward(request, response);
        }
        else
        {
            request.setAttribute("errocreation", "votre demande avec le login" +
c.getLogin()+
                "à échouée, la requete n'a pas pu etre terminée");

            request.getRequestDispatcher("Nouveaumontepage.jsp").forward(request, response);
        }
    }
    /*
    *
    *
    *
    */

```

```

//TRAITEMENT DE l'action recherche dun compte Inactif
else if(request.getParameter("Action").equals("Activercompte"))
{

    if(CredentialsDAO.getInactifCredential(request.getParameter("tlogin"))==null)
    {
        request.setAttribute("message", "le Login
"+request.getParameter("tlogin")+" est introuvable ou déjà actif" );

        request.getRequestDispatcher("Activer_compte.jsp").forward(request, response);
    }
    else
    {
        credentialsC =
CredentialsDAO.getInactifCredential(request.getParameter("tlogin"));
        request.setAttribute("credential", c);

        request.getRequestDispatcher("Activer_compte.jsp").forward(request, response);
    }

}
//TRAITEMENT de l'action modification du compte inactif
else if (request.getParameter("Action").equals("modifierinactifcompte"))
{
    String erroupdate=null;
    credentialsC = new credentialsC();

    c.setldcredential(Integer.parseInt(request.getParameter("tid")));
    c.setEtat(request.getParameter("tetat"));
    c.setRole(request.getParameter("trole"));

    int ret=CredentialsDAO.updatecredentials(c);
    if(ret>0)
    {
        request.setAttribute("login", c);

        request.getRequestDispatcher("confirmationupdate.jsp").forward(request, response);
    }
    else
    {
        request.setAttribute("erroupdate", "votre demande de mise a jours
du login " + c.getLogin()+
                                "à échouée, la requete n'a pas pu etre terminée");

        request.getRequestDispatcher("Activer_compte.jsp").forward(request, response);
    }
}

```

```

    }

  }

}

```

**20. Pour rendre la chose claire aussi pour l'utilisateur de l'application apres l'exécution de la tâche, nous allons créer une nouvelle page jsp nommée « conifmratioupdate », le code est comme suite :**

```

<%@ include file="Entete.jsp"%>
<div id="content">

    <% if(request.getAttribute("login")!=null) {
        credentialsC = (credentialsC)request.getAttribute("login");
    %>

    <p>
        le compte avec le login <%=c.getLogin() %> a été changé par l'administrateur, <br/>
        son nouveau état est <%=c.getEtat() %> <br/>
        son role est <%=c.getRole() %>
    </p>
    <%} %>
</div>

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

```

**21. Pour organiser cette gestion de l'administrateur, nous allons créer un menu pour gérer l'ensemble des pages de l'administration, en créant une nouvelle page nommée « AdministrationMenu.jsp » avec le code suivant :**

```

<%@page import="Entities.credentialsC"%>
<%@ 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.jsp">Tutoriel<span class="logo_colour">J2EE</span></a></h1>
        <h2>TP etape par etape</h2>
      </div>
      <div id="logo_text">
        <% if(session.getAttribute("UTILISATEUR")==null)
        {
          response.sendRedirect("Login.jsp");
        }
        else
        {
          credentialsC user=(credentialsC)session.getAttribute("UTILISATEUR");
        } %>
        <div class="logo_text">
          Bienvenue <%=user.getLogin() %> | <a href="Logoutcontroler">Déconnexion</a>
        </div>
        <%} %>
      </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">Home</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>
        <% if(session.getAttribute("UTILISATEUR")!=null)
        { credentialsC c= (credentialsC)session.getAttribute("UTILISATEUR");
          if(c.getRole().equals("admin")){ %>
            <li><a href="AdministrationMenu.jsp">Administration</a></li>
          <%}
        } %>
      </ul>
    </div>
    <div id="site_content">
      <div class="sidebar">
        <!-- La barre des news a droite -->
      </div>
    </div>
  </div>

```

```

    <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>

```

**22. Ensuite modifier le fichier « style.css » et ajouter le code suivant à la fin :**

```

/* ***** CSS Que pour le menu de la page d'administration*/

body {
font-family: Alegreya Sans;
background: inherit;
}

.menu {
position: relative;
background: #cd3e3d;
width: 3em;
height: 3em;
border-radius: 5em;
margin: auto;
margin-top: 5em;
margin-bottom: 5em;

```

```

    cursor: pointer;
    border: 1em solid #66ccff;
  }
  .menu:after {
    content: "";
    position: absolute;
    top: 1em;
    left: 1em;
    width: 1em;
    height: 0.2em;
    border-top: 0.6em double #fff;
    border-bottom: 0.2em solid #fff;
  }
  .menu ul {
    list-style: none;
    padding: 0;
  }
  .menu li {
    width: 5em;
    height: 1.4em;
    padding: 0.2em;
    margin-top: 0.2em;
    text-align: center;
    border-top-right-radius: 0.5em;
    border-bottom-right-radius: 0.5em;
    transition: all 1s;
    background: #0000ff;
    opacity: 0;
    z-index: -1;
  }
  .menu:hover li {
    opacity: 10;
  }
  /**
   * Add a pseudo element to cover the space
   * between the links. This is so the menu
   * does not lose :hover focus and disappear
   */
  .menu:hover ul::before {
    position: absolute;
    content: "";
    width: 0;
    height: 0;
    display: block;
    left: 50%;
    top: -5.0em;
  }

```



```

/**
 * The pseudo-element is a semi-circle
 * created with CSS. Top, bottom, and right
 * borders are 6.5em (left being 0), and then
 * a border-radius is added to the two corners
 * on the right.
 */
border-width: 6.5em;
border-radius: 0 7.5em 7.5em 0;
border-left: 0;
border-style: solid;
/**
 * Have to have a border color for the border
 * to be hoverable. I'm using a very light one
 * so that it looks invisible.
 */
border-color: rgba(0,0,0,0.01);
/**
 * Put the psuedo-element behind the links
 * (So they can be clicked on)
 */
z-index: -1;
/**
 * Make the cursor default so it looks like
 * nothing is there
 */
cursor: default;
}
.menu a {
  color: white;
  text-decoration: none;
}
/**
 * This is to vertically center the text on the
 * little tab-like things that the text is on.
 */
line-height: 1.5em;
}
.menu a {
  color: black;
  font-size: 12px;
  text-decoration: none;
}
.menu ul {
  transform: rotate(180deg) translateY(-2em);
  transition: 1s all;
}

```

```
.menu:hover ul {
  transform: rotate(0deg) translateY(-1em);
}
.menu li:hover {
  background: #cd3e3d;
  z-index: 10;
}

.menu li:nth-of-type(1) {
  transform: rotate(-90deg);
  position: absolute;
  left: -1.2em;
  top: -4.2em;
}
.menu li:nth-of-type(2) {
  transform: rotate(-45deg);
  position: absolute;
  left: 2em;
  top: -3em;
}
.menu li:nth-of-type(3) {
  position: absolute;
  left: 3.4em;
  top: 0.3em;
}
.menu li:nth-of-type(4) {
  transform: rotate(45deg);
  position: absolute;
  left: 2em;
  top: 3.7em;
}
.menu li:nth-of-type(5) {
  transform: rotate(90deg);
  position: absolute;
  left: -1.2em;
  top: 5em;
}
.hint {
  text-align: center;
}
```

**23. Exécuter la page pour voir le résultat et essayer de cliquer sur le premier menu.**