

Rapport_TP4

Objectif :

Une base de données sera créée pour gérer les comptes.

- La base de données s'appellera « jeu ».
- Le compte pour s'y connecter a pour login « jeu » et le mot de passe « java123 »
- La table des comptes s'appellera « comptes ».

Un lien « créer un compte » sera ajouté sur la page de login. En cliquant dessus, l'utilisateur basculera sur une autre page lui permettant de créer un compte. Ce compte sera enregistré dans la base de données et permettra à l'utilisateur de s'authentifier sur la page principale.

Application :

Afin de mettre en place la base de données nous allons utiliser MySQL.

Tout d'abord nous allons créer la base de données :

```
root@Lenovo:/home/networks# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 85
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE Jeu;
Query OK, 1 row affected (0,001 sec)
```

Le compte pour s'y connecter doit être créé et des droits doivent lui être attribués :

```
MariaDB [(none)]> CREATE USER 'jeu'@'localhost' IDENTIFIED BY 'java123';
Query OK, 0 rows affected (0,003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON Jeu.* TO 'jeu'@'localhost';
Query OK, 0 rows affected (0,007 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)
```

Nous allons maintenant insérer des données dans la base de données :

```
root@Lenovo:/home/networks/Téléchargements/Licence professionnelle - CNAM/Cours/Programmation Web et mobiles/NT2/TP4/tp4_comptes_sql# mysql -u jeu -p
```

```
MariaDB [(none)]> use Jeu;
Database changed
MariaDB [Jeu]> source tp4_comptes.sql;
Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected, 1 warning (0,008 sec)

Query OK, 0 rows affected (0,002 sec)

Query OK, 3 rows affected (0,001 sec)
Records: 3  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)

Query OK, 0 rows affected (0,000 sec)
```

(Contenu du script SQL)

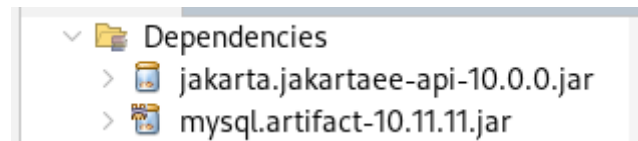
```
DROP TABLE IF EXISTS `comptes`;
CREATE TABLE IF NOT EXISTS `comptes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `Nom` varchar(50) NOT NULL,
  `Prenom` varchar(50) NOT NULL,
  `Age` int(11) NOT NULL,
  `Login` varchar(50) NOT NULL,
  `Motdepasse` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

--
-- Déchargement des données de la table `comptes`
--

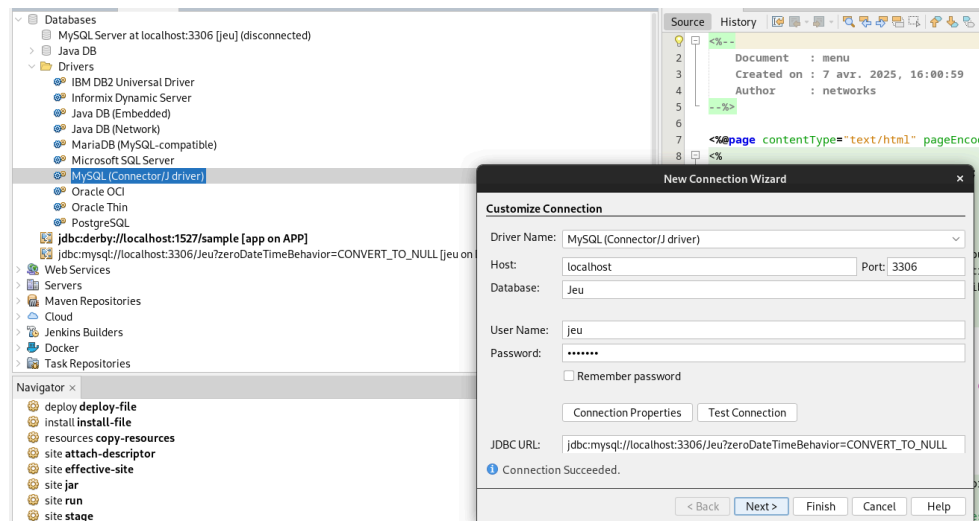
INSERT INTO `comptes` (`id`, `Nom`, `Prenom`, `Age`, `Login`, `Motdepasse`) VALUES
(1, 'Meyer', 'Luc', 30, 'meyer', '1111'),
(2, 'Dupont', 'René', 40, 'dupont', '2222'),
(3, 'Legrand', 'Lisa', 16, 'legrand', '3333');
COMMIT;
```

Une fois la base de données implémentée, nous devons procéder à l'installation du driver MySQL dans le projet Netbeans.

(Capture du projet Netbeans)



(Capture du projet Netbeans)



Un lien "Créer un compte" est ajouté à l'interface authentification.

(Capture de l'interface authentification)



Une servlet Inscription.java est mise en place afin de récupérer les informations saisies par l'utilisateur pour ensuite effectuer une insertion en base de données. Si l'utilisateur ne remplit pas tous les champs du formulaire, l'insertion n'est pas effectuée et il en est informé.

(Extrait de la servlet Inscription.java)

```
78      @Override
79      protected void doPost(HttpServletRequest request, HttpServletResponse response)
80          throws ServletException, IOException {
81
82          String message = null;
```

(Extrait de la servlet Inscription.java)

```
84      try {
85          //Déclaration du driver :
86          Class.forName("com.mysql.cj.jdbc.Driver");
87          //Url de connexion
88          String dbUrl="jdbc:mysql://localhost:3306/Jeu?useSSL=false";
89          String user = "jeu";
90          String password = "java123";
91          //connexion à la base
92          java.sql.Connection dbcon = DriverManager.getConnection(dbUrl,user,password);
93
94          String nom = request.getParameter("nom");
95          String prenom = request.getParameter("prenom");
96          String login = request.getParameter("login");
97          String mdp = request.getParameter("password");
98          int age = Integer.parseInt(request.getParameter("age"));
99
100         PreparedStatement psmt = dbcon.prepareStatement("insert into comptes(Nom, Prenom, Age, Login, Motdepasse) values(?, ?, ?, ?, ?)");
101         psmt.setString(1, nom);
102         psmt.setString(2, prenom);
103         psmt.setInt(3, age);
104         psmt.setString(4, login);
105         psmt.setString(5, mdp);
106         psmt.executeUpdate();
107
108         message = "Bravo ! Vous êtes maintenant enregistré.";
109
110         request.setAttribute("message", message);
111         request.getRequestDispatcher("inscription").forward(request, response);
112
113     } catch (ClassNotFoundException ex) {
114         Logger.getLogger(Inscription.class.getName()).log(Level.SEVERE, null, ex);
115     } catch (SQLException ex) {
116         Logger.getLogger(Inscription.class.getName()).log(Level.SEVERE, null, ex);
117     }
```

(Extrait de la servlet Inscription.java)

```
118
119         request.setAttribute("message", message);
120         request.getRequestDispatcher("inscription").forward(request, response);
121     }
```

(Capture de la JSP inscription)

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <% String message = null;
9 if(request.getAttribute("message") != null){
10     message = (String) request.getAttribute("message");
11 } %>
12 <!DOCTYPE html>
13 <html>
14 <head>
15     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
16     <title>Inscription</title>
17 </head>
18 <body>
19     <h1>Création de compte</h1>
20     <h4>Entrez votre login et votre mot de passe</h4>
21     <form action="Inscription" method="POST">
22     <div>
23         <label for="nom">Nom :</label>
24         <input type="text" name="nom" id="nom" maxlength="50" required/>
25     </div>
26     <br/>
27     <div>
28         <label for="prenom">Prénom :</label>
29         <input type="text" name="prenom" id="prenom" maxlength="50" required/>
30     </div>
```

(Capture de la JSP inscription)

```
30 </div>
31 <br/>
32 <div>
33     <label for="login">Login :</label>
34     <input type="text" name="login" id="login" maxlength="50" required/>
35 </div>
36 <br/>
37 <div>
38     <label for="password">Mot de passe :</label>
39     <input type="password" name="password" id="password" maxlength="50" required/>
40 </div>
41 <br/>
42 <div>
43     <label for="age">Age :</label>
44     <input type="number" name="age" id="age" min="0" max="150" required/>
45 </div>
46 <br/>
47 <input type="submit" value="Créer"><br/>
48 </form>
49 <% if(message != null){ %>
50     <br/>
51     <h4><%= message %></h4>
52 <% } %>
53 <br/>
54 <a href="connexion">Retour à l'accueil</a>
55 </body>
56 </html>
```

Si l'insertion est effectuée, un message informant l'utilisateur est généré.

(Capture de l'interface inscription)

← → ↻ localhost:8080/NT2-4/Inscription

Création de compte

Entrez votre login et votre mot de passe

Nom :

Prénom :

Login :

Mot de passe :

Age :

Bravo ! Vous êtes maintenant enregistré.

[Retour à l'accueil](#)

La servlet Authentification.java est mise à jour pour permettre aux nouveaux utilisateurs de se connecter.

(Extrait de la servlet Authentification.java)

```
80 @Override
81 protected void doPost(HttpServletRequest request, HttpServletResponse response)
82     throws ServletException, IOException {
83
84     // Recherche du joueur après la validation du formulaire
85     String login = request.getParameter("Login");
86     String mdp = request.getParameter("Mdp");
87     Joueur joueur = null;
88
```

(Extrait de la servlet Authentification.java)

```
89 try {
90     //Déclaration du driver :
91     Class.forName("com.mysql.cj.jdbc.Driver");
92     //Url de connexion
93     String dbUrl="jdbc:mysql://localhost:3306/Jeu?useSSL=false";
94     String user = "jeu";
95     String password = "java123";
96     //connexion à la base
97     java.sql.Connection dbcon = DriverManager.getConnection(dbUrl,user,password);
98
99     PreparedStatement psmt = dbcon.prepareStatement("select * from comptes where Login = ? and Motdepasse = ?");
100     psmt.setString(1, login);
101     psmt.setString(2, mdp);
102     java.sql.ResultSet rs = psmt.executeQuery();
103     if(rs.next()){
104         joueur = new Joueur(rs.getInt("Id"), rs.getString("Nom"), rs.getString("Prenom"), rs.getInt("Age"), rs.getString("Login"), rs.getString("Motdepasse"));
105         System.out.println(joueur);
106     }
107
108 } catch (ClassNotFoundException ex) {
109     Logger.getLogger(Inscription.class.getName()).log(Level.SEVERE, null, ex);
110 } catch (SQLException ex) {
111     Logger.getLogger(Inscription.class.getName()).log(Level.SEVERE, null, ex);
112 }
```

(Extrait de la servlet Authentification.java)

```
114 // Création de la session et de ses attributs si les identifiants sont correct
115 if (joueur != null) {
116     HttpSession session = request.getSession();
117     session.setAttribute("nom", joueur.getNom());
118     session.setAttribute("prenom", joueur.getPrenom());
119     session.setAttribute("age", joueur.getAge());
120     request.getRequestDispatcher("menu").forward(request, response);
121 } else {
122     request.getRequestDispatcher("connexion").forward(request, response); // Retour au formulaire si échec de l'authentification
123 }
124 }
```