



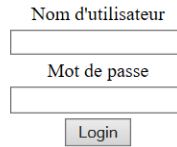
# 1 Page Login

## 1.1 Page d'identification

Créons premièrement un fichier Login.jsp dans lequel on aura la description de la page d'identification des différents utilisateurs.

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Login</title>
8 </head>
9 <body>
10 <div align = "center">
11   <form action="login" method="post">
12     <table>
13       <tr>
14         <td align ="center">Nom d'utilisateur</td>
15       </tr>
16       <tr>
17         <td><input type="text" name="nom"></td>
18       </tr>
19       <tr>
20         <td align ="center">Mot de passe</td>
21       </tr>
22       <tr>
23         <td><input type="password" nom="mot de passe"></td>
24       </tr>
25       <tr>
26         <td align ="center"><input type="submit" value="Login"></td>
27       </tr>
28     </table>
29   </form>
30 </div>
31 </body>
32 </html>
```

On aura le resultat simple suivant



The image shows a simple web form for login. It consists of two text input fields stacked vertically. The first field is labeled 'Nom d'utilisateur' and the second is labeled 'Mot de passe'. Below the second field is a button labeled 'Login'.

Figure 1: Login.jsp

## 1.2 Connexion avec une base de données MySQL : vérification des identifiants

### 1.2.1 Classe Session

On aura besoin d'une classe *Session* (package : loginsession) qui va récupérer pendant chaque identification le nom et le mot de passe entrés.

```
1
2 package loginsession;
3
4 public class session {
5     private String nom;
6     private String passe;
7     public String returnNom() {
8         return nom;
9     }
10    public String affecteNom(String nom) {
11        this.nom = nom;
12    }
13    public String returnPasse() {
14        return passe;
15    }
16    public String affectePasse(String passe) {
17        this.passe = passe;
18    }
19 }
```

### 1.2.2 Classe DB

La classe DB (Package : base\_donnees) va permettre dans un premier lieux la connexion avec une base de données MySQL (userdb) qu'on va créer par la suite, puis vérifie si les identifiants (nom et mot de passe) entrés figurent dans cette base.

```
1 package base_donnees;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import login.session.*;
8 public class DB {
9     private String dbUrl = "jdbc:mysql://localhost:3306/userdb?
10                             useJDBCCompliantTimezoneShift=true&
11                             useLegacyDatetimeCode=false&serverTimezone=UTC";
12     private String dbUsername = "AzzimDriouich";
13     private String dbPassword = "0000";
14     private String dbDriver = "com.mysql.cj.jdbc.Driver";
15     public void loadDriver(String dbDriver)
16     {
17         try {
18             Class.forName(dbDriver);
19         } catch (ClassNotFoundException e) {
20             e.printStackTrace();
21         }
22     }
23     public Connection getConnection()
24     {
25         Connection con = null;
26         try {
27             con = DriverManager.getConnection(dbUrl, dbUsername, dbPassword);
28         } catch (SQLException e) {
29             e.printStackTrace();
30         }
31         return con;
32     }
33     public boolean valider_donnees(Session session)
34     {
35         boolean status = false;
36
37         loadDriver(dbDriver);
38         Connection con = getConnection();
39         String sql = "SELECT *
```

```

40         FROM login
41         WHERE nom = ?
42         AND mot_de_passe =?";
43     PreparedStatement ps;
44     try {
45         ps = con.prepareStatement(sql);
46         ps.setString(1, session.returnNom());
47         ps.setString(2, session.returnPasse());
48         ResultSet rs = ps.executeQuery();
49         status = rs.next();
50
51     } catch (SQLException e) {
52         e.printStackTrace();
53     }
54     return status;
55 }
56 }

```

### 1.2.3 Création de la base de données "userdb"

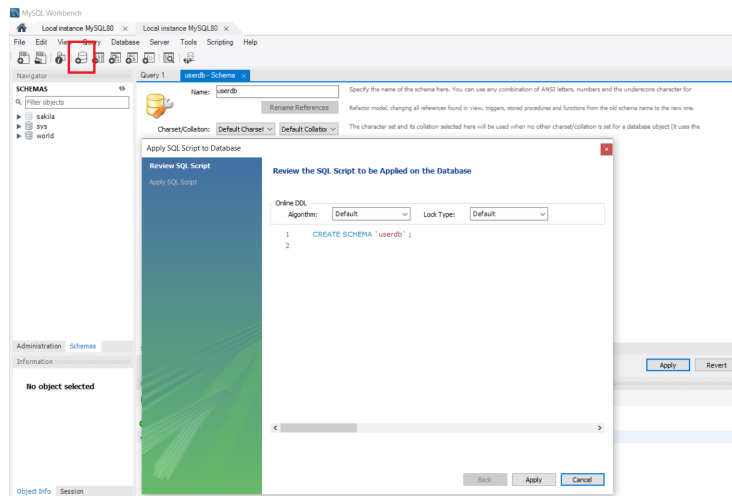


Figure 2: create schema

Après, on doit créer note tableau *login* avec les deux colonnes *nom* (Clé primaire et non null) et *mot\_de\_passe* (non null).

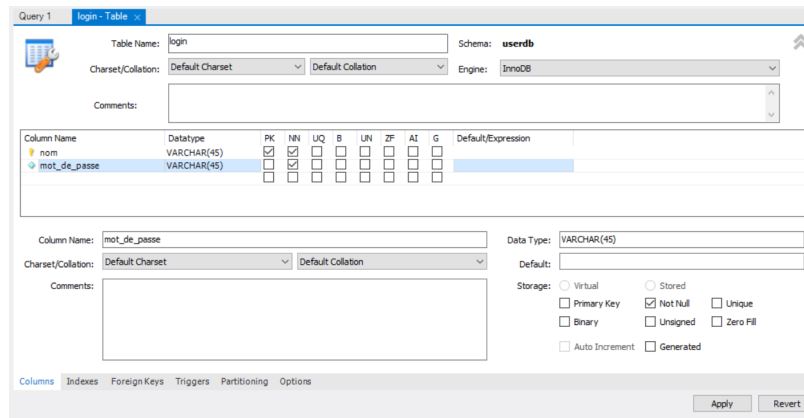


Figure 3: Création de la table

Afin de se connecter, on va insérer quelques utilisateurs à la table *login*.

Review the SQL Script to be Applied on the Database

```

1  INSERT INTO 'userdb'..'login' ('nom', 'mot_de_passe') VALUES ('Azzim', '111');
2  INSERT INTO 'userdb'..'login' ('nom', 'mot_de_passe') VALUES ('Driouich', '222');
3

```

Figure 4: Insertion des utilisateurs

### 1.3 Login servlet

Dans le package web, introduisant la première servlet qui va se servir de l'authentification et diriger l'utilisateur vers son compte si les identifiants sont corrects ou actualiser la page login sinon.

Pour cela, ajoutons un simple fichier Succes.jsp

```

1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2      pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6  <meta charset="ISO-8859-1">
7  <title>Succes</title>
8  </head>
9  <body>
10  Succes
11  </body>
12  </html>

```

Puis la servlet serait comme suit :

```
1 package web;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import base_donnees.*;
10 import loginsession.*;
11
12
13 @WebServlet("/login")
14 public class LoginServlet extends HttpServlet {
15
16     protected void doPost(HttpServletRequest request,
17                           HttpServletResponse response)
18         throws ServletException, IOException {
19         String nom = request.getParameter("nom");
20         String passe = request.getParameter("mot de passe");
21
22         Session session = new Session();
23         session.affecteNom(nom);
24         session.affectePasse(passe);
25
26         DB connexion_db = new DB();
27         if(connexion_db.valider_donnees(session)) {
28             response.sendRedirect("Succes.jsp");
29         }
30         else {
31             response.sendRedirect("login.jsp");
32         }
33     }
34
35 }
```

## 1.4 Tester le login

Exécution du programme et essayons une authentification avec l'un des utilisateurs déclarés dans la base de données :

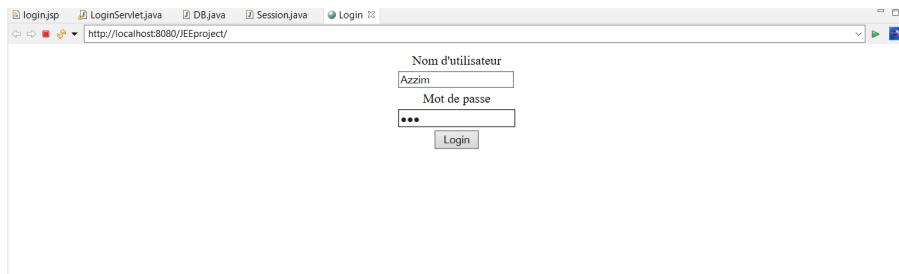


Figure 5: Exécution

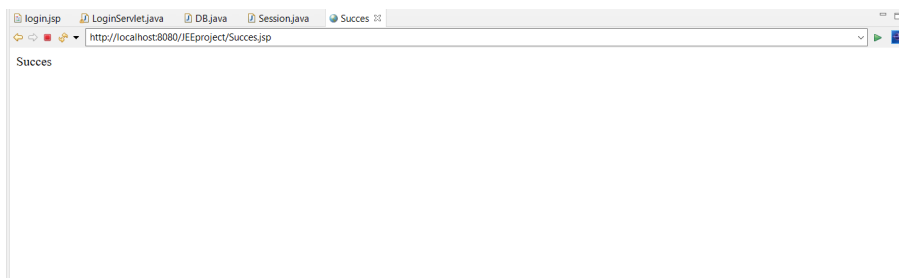


Figure 6: Authentification avec succès