

Hentges  
Mathieu

## Java TP n 5

But : Créer un système de réservation de pièce de théâtre, en utilisant l'ensemble des techniques développées au cours des précédents TP (interface, BDD, sockets).

Tout d'abord mon code n'est pas optimisé et ne comporte que deux classes dans deux projets différents, le traitement est fait dans plusieurs méthodes.

Pour la base de données j'ai utilisé une base mysql en local (ci-joint dans le mail un fichier contenant le code nécessaire à sa création)

### 1 Liaison base de données

Toutes les relations avec la base de données sont gérées par la partie server.  
On se connecte à notre BDD mysql grâce au code suivant

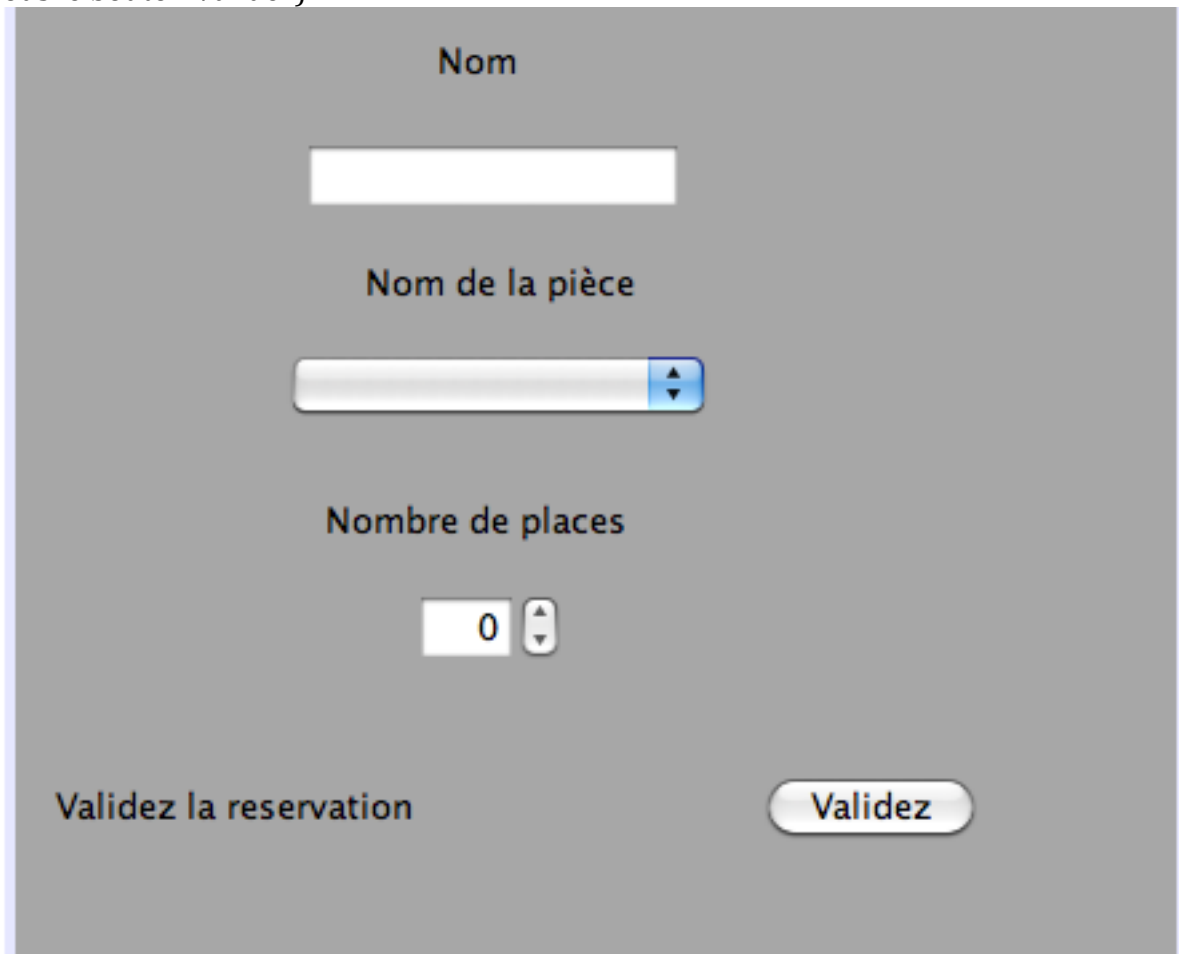
```
Class.forName("com.mysql.jdbc.Driver").newInstance();  
String url = "jdbc:mysql://localhost/TP5";  
conn = DriverManager.getConnection(url, "root", "root");
```

### 2 Liaison server/client

Afin que les serveurs communiquent entre eux j'ai réutilisé le système du TP précédents utilisant les sockets

### 3 Interface

Pour l'interface j'utilise SWIN GUI FORM afin de gagner du temps  
Voici une capture d'écran de la forme de ma fenêtre (un dernier JLabel vide est présent sous le bouton Validez)



### 4 Déroulement

Tout d'abord on lance le serveur qui attend que le client se connecte.  
On lance le client, on vérifie que la connexion fonctionne en échangeant plusieurs texte de 'bienvenue'.

```
System.out.println("Waiting answer from server");  
if ("Hello".equals(inSocket.readLine())) {  
    System.out.println("he is here !");  
}
```

*Le client attend de recevoir hello du serveur*

Une fois la connexion réussie le serveur appelle la méthode ListPlay qui récupère la liste des pièces disponibles afin de l'envoyer au client qui l'affichera dans un JComboBox.

```
private ArrayList listPlay(){
    ArrayList play=new ArrayList();

    System.out.println("SELECT en cours");
    String query = "SELECT nom FROM piecesTheatre";
    try
    {
        Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(query);
        while (rs.next())
        {

            play.add( rs.getString("nom"));
            System.out.println("List of play found");

        }
    }
    catch (SQLException ex)
    {
        System.err.println(ex.getMessage());
    }
    return play;
}
```

*Méthode listPLay*

```
for(int i=0;i<list.size();i++) { jComboBox1.addItem(list.get(i));}
```

*Boucle for afin de remplir le jCombo (list étant la liste contenant les pièces)*

L'utilisateur remplit alors ses champs. Lorsque celui clique sur le bouton on vérifie que l'on possède assez de place pour lui fournir les places. Pour cela il faut envoyer le nb de place sélectionner aux serveur pour qu'il interroge la BDD on utilise pour cela la méthode suivante qui retourne le nombre de place disponible pour la pièce sélectionner par l'utilisateur.

```

public Integer availableSeet() throws IOException, ClassNotFoundException{
    String play=jComboBox1.getSelectedItem().toString();
    outSocket.println(play);
    System.out.println("Waiting for remaining ticket");
    Object object = in.readObject();
    int nbOfPlace;
    nbOfPlace= (Integer)object;
    return nbOfPlace;
}

```

*Méthode availableSeet*

On appelle la méthode nbOfPlaces coté serveur qui permet de récupérer le nombre de place disponible.

Une fois ce nombre récupérer on le compare au nombre de place voulu par l'utilisateur, si on en a pas assez en stock on prévient l'utilisateur et rien d'autre ne se passe, sinon on passe à la réservation.

Pour réserver on utilise deux méthodes, une qui permet de mettre à jour le nb de place disponible pour la pièce et de sauvegarder la réservation dans la BDD et donne le numéro de réservation. Il faut pour cela récupérer l'iD de la pièce et l'autre qui donne le numéro de réservation

```

private void keepReservation(String name, String play, String nb){
    String id="";
    System.out.println("select");
    String query = "SELECT ID FROM piecesTheatre WHERE nom='"+play+"'";
    try
    {
        Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(query);
        while (rs.next())
        {
            id = rs.getString("ID");
        }
    }
    catch (SQLException ex)
    {
        System.err.println(ex.getMessage());
    }
    try
    {
        Statement st = conn.createStatement();
        st.executeUpdate("INSERT INTO Réservation(nom,PieceID,nbplace) VALUES('"+name+"','"+id+"','"+nb+"')");
        System.out.println("reservation Done");
    }
    catch (SQLException ex)
    {
        System.err.println(ex.getMessage());
    }
}

```

*Méthode permettant de stocker la réservation dans la BDD*