Client :

public class UtilisateurDAOImpl\_client implements UtilisateurDAO {

private ObjectOutputStream toServer;

private ObjectInputStream fromServer;

public UtilisateurDAOImpl\_client(ObjectOutputStream toServer, ObjectInputStream fromServer) {

this.toServer = toServer;

this.fromServer = fromServer;

}

@Override

public Utilisateur getUtilisateur(String mail) throws IOException, ClassNotFoundException {

Message clientAuth = new ClientAuth(mail, "motdepasse");

//Envoie du message au serveur

toServer.writeObject(clientAuth);

toServer.flush();

//Debug

System.out.println("Message " + clientAuth.toString() + " envoyée au serveur");

//On attend la réponse

ClientSession response = (ClientSession) fromServer.readObject();

Utilisateur utilisateur = null;

if(response.getSQLerr() == null){

//On récupère l'utilisateur

utilisateur = response.getUtilisateur();

}else{

//Afficher l'erreur sql à l'utilisateur

System.out.println("Erreur sql : "+response.getSQLerr());

}

//Debug

System.out.println("\t \* " + response + " : Réponse reçue " + response.toString());

return utilisateur;

}

}

Serveur :

public class UtilisateurDAOImpl\_serveur implements UtilisateurDAO {

private Statement sqlStmt;

public UtilisateurDAOImpl\_serveur(Statement sqlStmt) {

this.sqlStmt = sqlStmt;

}

@Override

public Utilisateur getUtilisateur(String mail) throws SQLException {

ResultSet rs=this.sqlStmt.executeQuery("SELECT \* FROM Utilisateurs INNER JOIN Possede ON " + "Utilisateurs\_idUtilisateur = idUtilisateur INNER JOIN Roles ON Roles\_idRole = idRole WHERE courriel='"+mail+"';");

Utilisateur user = null;

while (rs.next()) {

int idUtilisateur = Integer.parseInt(rs.getString("idUtilisateur"));

String nom = rs.getString("nom");

String prenom = rs.getString("prenom");

String courriel = rs.getString("courriel");

String motdepasse = rs.getString("motDePasse\_hash");

String sel = rs.getString("motDePasse\_salt");

String role = rs.getString("nomRole");

if(role.equals("Candidat")){

user = new Candidat(idUtilisateur, nom, prenom, courriel, motdepasse);

}

}

return user;

}

}

Message :

package app.message;

public class ClientAuth extends Message {

private String login;

private String motDePasse;

public ClientAuth(String login, String motDePasse) {

this.login = login;

this.motDePasse = motDePasse;

}

@Override

public String toString() {

return "ClientAuth{" +

"login='" + login + '\'' +

", motDePasse='" + motDePasse + '\'' +

'}';

}

public String getLogin() {

return login;

}

public String getMotDePasse() {

return motDePasse;

}

}

Modèle :

// ok

Test :

public class ClientConnexion implements Runnable {

private Socket connexion = null;

private ObjectOutputStream toServer = null;

private ObjectInputStream fromServer = null;

private static int count = 0;

private String name = "Client-";

public ClientConnexion(String host, int port) {

name += ++count;

try {

connexion = new Socket(host, port);

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

public void run() {

try {

toServer = new ObjectOutputStream(connexion.getOutputStream());

fromServer = new ObjectInputStream(connexion.getInputStream());

//Procédure d'Authentifcation au serveur (suite à l'appuie sur un button)

Utilisateur utilisateur = new UtilisateurDAOImpl\_client(toServer, fromServer).getUtilisateur("TonyaPolanco@mail.com");

} catch (IOException | ClassNotFoundException e1) {

e1.printStackTrace();

}

//toServer.close();

}

}