

EPSI – JAVA – 3ème année –JDBC et JavaFX

Reprenez votre travail du TP3 ou repartez du corrigé présent sur le dépôt GitHub.

1) Prise en main de JDBC

Le stockage actuel de vos données est effectué de manière peu industrielle à l'aide d'objets sérialisés sur votre disque local. L'utilisation d'une base de données permettra de simplifier la gestion de ces données.

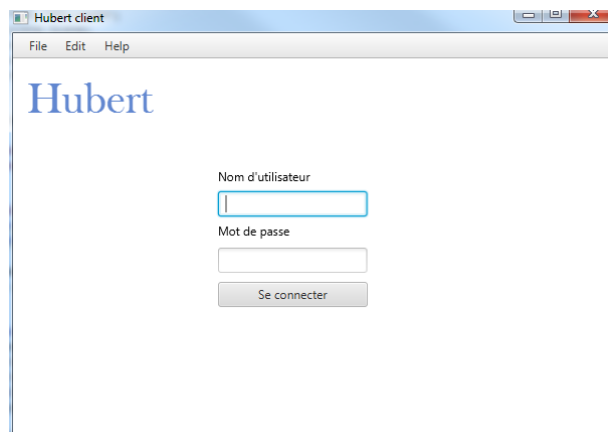
1. Si ce n'est pas déjà le cas, installez MySQL sur votre poste
2. A l'aide d'un outil graphique (phpMyAdmin, MySQL Workbench) ou en ligne de commande, exécutez le script de création de la base se trouvant sur le dépôt du cours <https://github.com/idassonvil/epsi-java/tree/master/cours/S4>
3. Dans votre projet, importez le Driver JDBC se trouvant également dans le dépôt
 - a. Clic droit sur le projet puis *Build Path > Add external archive*
4. Testez le bon fonctionnement du Driver
5. Remplacez le code de sauvegarde et de chargement des utilisateurs se trouvant dans la classe *Application* par des appels à la base de données
 - a. Dans un premier temps implémentez la sauvegarde dans la table Utilisateur
 - b. Implémentez ensuite la gestion de la relation Chauffeur <-> Véhicule
 - c. Implémentez enfin le chargement depuis la BDD

2) Premiers pas avec Java FX

- 1) Ajoutez la bibliothèque Java Fx au projet
 - a. Clic droit sur le projet puis *Java Build Path > Librairies > Add Library* puis sélectionnez *JavaFx*
- 2) Dans le package `fr.hubert.ui.view` ajoutez une classe *MainView* qui étend la classe `javafx.Application`
- 3) Démarrez cette interface depuis la méthode *main* à l'aide la méthode `Application.launch`
- 4) Afin de pouvoir continuer à utiliser les fonctionnalités existantes, gérez un nouveau paramètre de démarrage pour différencier un lancement en mode client lourd d'un lancement en mode console
- 5) Lancez votre application en mode client lourd, un écran vide doit s'afficher
- 6) Créez un fichier `app.fxml` dans le package `fr.hubert.ui.view`
- 7) Ouvrez ce fichier avec *SceneBuilder* depuis Eclipse
- 8) Utilisez un Layout de type *BorderPane* puis ajoutez une barre de menu
- 9) Dans le fichier *MainView*, charger la vue `app.fxml` puis affichez là à l'aide d'une scène

3) Premier écran

L'objectif est de réaliser un écran de login de ce type :



- 1) Créez un fichier `login.fxml` dans le package `fr.hubert.ui.view`

- 2) A l'aide de *SceneBuilder* editez ce fichier
- 3) Ajoutez le *Layout* qui vous semble le plus pertinent (*BorderPane*, *GridPane*, *AnchorPane*...)
- 4) Ajoutez un champ de saisie pour le nom d'utilisateur et un pour le mot de passe, ainsi qu'un bouton de validation
- 5) Modifiez le fichier *MainView* pour que la vue *loginView* soit utilisée comme racine de la première scène
- 6) Testez le rendu de votre application

4) Gestion des événements

- 1) Ajoutez un contrôleur *LoginController* dans le package *fr.hubert.ui.controller*
- 2) Associez ce contrôleur à sa vue dans le fichier *FXML*
- 3) Gérez l'événement de click sur le bouton se connecter
 - a. Si le nom d'utilisateur ou le mot de passe n'est pas renseigné, le champ en question doit s'afficher en rouge et aucune requête ne doit être envoyée
 - b. Si les paramètres sont bien saisis une requête de login doit être envoyée au serveur
 - i. Si la requête échoue, un message d'erreur doit s'afficher
 - ii. Si la requête réussit, la scène correspondant au type d'utilisateur doit s'afficher (voir exercice suivant)

5) Gérer plusieurs scènes

Lorsqu'un utilisateur est parvenu à s'authentifier, un écran différent doit s'afficher selon son type d'utilisateur (chauffeur, client, admin).

- 1) Créez deux nouvelles vues pour le type d'utilisateur *Chauffeur* et *Client*
 - a. Affichez dans chacune des vues les actions disponibles
- 2) En utilisant une référence vers la classe *MainView*, implémentez le changement de scène depuis la classe *LoginController*

6) Pour aller plus loin

Gérez les fonctionnalités suivantes dans l'UI :

- 1) Recherche d'un taxi est affichage des résultats
- 2) Ajout du véhicule d'un chauffeur
- 3) Affichage des statistiques pour un profil admin
- 4) Déconnexion