
JAVA TP08: Scene Builder

I. Introduction	1
II. Configuration de l'environnement	1
III. Création du projet	2
IV. Conception de l'interface avec Scene Builder	3
V. Implémentation du Contrôleur (MainController.java)	5
VI. Implémentation de l'Application (MainApp.java)	5
VII. Test et Validation	6
VIII. Conclusion	6

I. Introduction

Dans ce TP, nous allons voir comment configurer un environnement de travail avec JavaFX.

II. Configuration de l'environnement

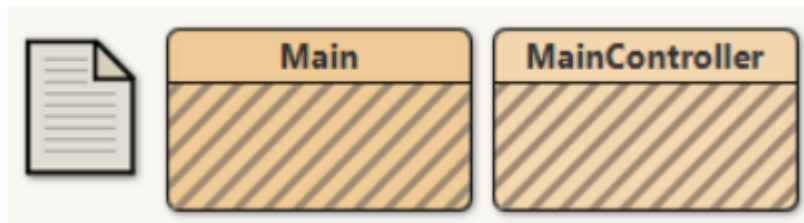
Pour réaliser ce projet, les étapes de configuration suivantes sont nécessaires :

- Logiciel de Développement : Utiliser l'environnement de développement BlueJ (disponible sur www.bluej.org/).

- Installation de JavaFX : Télécharger et installer le SDK JavaFX depuis le site de Gluon (<https://gluonhq.com/products/javafx/>).
- Configuration de BlueJ :
 - Dézipper le fichier téléchargé. Le dossier extrait contient les sous-dossiers bin, legal, et lib, ainsi qu'un fichier src.zip.
 - Ouvrir BlueJ, aller dans Outils > Préférences.
 - Dans l'onglet Bibliothèque, ajouter tous les fichiers .jar contenus dans le dossier lib en cliquant sur le bouton Add.
 - Redémarrer BlueJ pour finaliser la configuration.

III. Création du projet

1. Créer un nouveau projet dans BlueJ, nommé par exemple Scenebuilder.
2. Créer les classes Java : Deux classes sont nécessaires, Main et MainController.

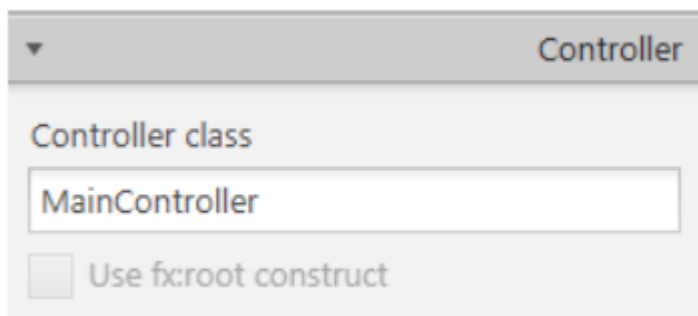


3. Créer le fichier FXML : Étant donné que BlueJ ne le génère pas automatiquement, créer manuellement un fichier nommé Mainview.fxml dans le répertoire du projet :
C:\Users\eleve\Scenebuilder

Main.java	09/12/2025 14:11	Fichier JAVA	1 Ko
MainController.java	09/12/2025 14:11	Fichier JAVA	1 Ko
package.bluej	09/12/2025 14:13	BlueJ Project F...	1 Ko
README.TXT	09/12/2025 14:11	Document texte	1 Ko
Mainview.fxml	09/12/2025 14:20	Fichier FXML	0 Ko

IV. Conception de l'interface avec Scene Builder

1. Ouvrir Scene Builder séparément.
2. Charger le FXML : Cliquer sur Open project et sélectionner le fichier Mainview.fxml créé précédemment.
3. Définir le Contrôleur : Dans la section Controller class, définir la classe de contrôle comme MainController.



4. Ajouter les Composants de Base :
 - Rechercher AnchorPane dans la Library (colonne de gauche) et le glisser-déposer sur la zone de travail (colonne du milieu).
 - Ajouter un TextField (pour le nom d'utilisateur) et un PasswordField (pour le mot de passe).
 - En même temps sur la droite ajouter un nom pour informer l'utilisateur
 - Ajouter un Button (texte : Se connecter).
 - Voici le résultat :

A login form with two text input fields. The first field is labeled 'username' and the second is labeled 'password'. Below the fields is a button labeled 'Se connecter'.

5. Ajouter les Messages de Statut :

- Ajouter deux Labels supplémentaires, un pour le message d'erreur

An error message label with the text 'Identifiant ou mdp incorrect' in red. Below the text is a checkbox labeled 'Visible'.

et un pour le message de réussite :

A success message label with the text 'Connexion réussie' in green. Below the text is a checkbox labeled 'Visible'.

6. Définir les Identifiants (fx:id) : Pour que Java puisse reconnaître les éléments, définir leurs fx:id dans la section Code de la colonne de droite.

- Champ Utilisateur : usernameField.
- Champ Mot de passe : passwordField.
- Label d'erreur : errorLabel.
- Label de succès : successLabel.

7. Définir l'Action du Bouton : Pour le bouton Se connecter, ajouter la méthode #handleLogin dans la section On Action.

V. Implémentation du Contrôleur (MainController.java)

1. Récupérer le Squelette : Dans Scene Builder, aller dans View puis cliquer sur Show Sample Controller Skeleton pour générer le code de base du contrôleur.

2. Copier le code et le coller dans la classe MainController.java dans BlueJ.

3. Remplir la méthode handleLogin pour gérer la logique de connexion (vérification des identifiants) et l'affichage des labels de statut.

Les identifiants de test utilisés sont admin pour l'utilisateur et 1234 pour le mot de passe.

Le code complété doit vérifier les identifiants et basculer la visibilité des labels:

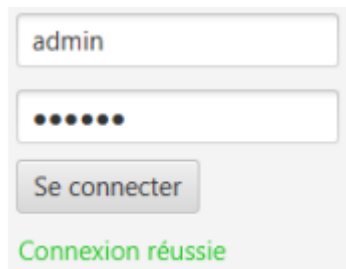
- Si succès : successLabel visible, errorLabel invisible.
- Si échec : errorLabel visible, successLabel invisible.

VI. Implémentation de l'Application (MainApp.java)

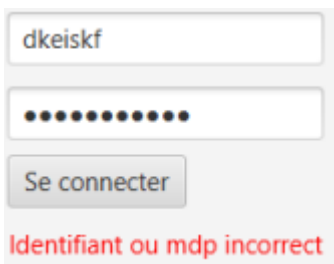
Remplir la classe MainApp pour lancer l'application, en chargeant la vue FXML (Mainview.fxml).

VII. Test et Validation

1. Faire un clic droit sur la classe MainApp dans BlueJ et sélectionner l'option Run JavaFX Application.
2. L'application de connexion s'affiche.
3. Tester la connexion avec les identifiants corrects (admin/1234) pour obtenir le message Connexion réussie.



4. Tester avec des identifiants incorrects pour afficher le message identifiant ou mdp incorrect.



VIII. Conclusion

Ce TP a permis de maîtriser les bases de la création d'une application de bureau avec JavaFX.

Nous avons réussi à :

1. Configurer l'environnement (JavaFX, BlueJ).
2. Concevoir l'interface de connexion graphiquement avec Scene Builder (FXML).
3. Implémenter la logique de vérification des identifiants dans le contrôleur Java.

L'exercice a démontré l'efficacité de la collaboration entre le FXML (Vue) et le Contrôleur Java, posant les fondations pour le développement d'applications JavaFX plus complexes en utilisant le modèle MVC.