STALTER Claire
N° candidat : 02245165513

Projet Java « musée des beaux-arts »

Guide d'installation

Sommaire

1. Introduction

Ce guide décrit les étapes à suivre pour faire tourner localement le projet Java « musée des beaux-arts de Rennes » que j'ai retenu pour l'épreuve E5, l'objectif étant de reproduire aussi fidèlement que possible l'environnement de développement. Les caractéristiques du projet sont les suivantes :

- il utilise la version 17 de JDK,
- son interface graphique est conçue en JavaFX et avec l'aide de SceneBuilder,
- il est lié à une base de données SQL Server administrée graphiquement via Microsoft SQL Server Management Studio.

Les étapes permettant l'installation de cet environnement sont données ci-dessous dans les détails. Noter que ce guide est valable pour une installation sur Windows. Par ailleurs, les processus d'installation et les copies d'écran ont été réalisés en avril 2023 et peuvent ne plus être entièrement d'actualité.

2. Installation des outils d'exécution et de programmation en Java

2.1. Installation de JDK 17

Sur le site d'Oracle, à l'adresse www.oracle.com/java/technologies/downloads/#jdk17-windows (ou équivalent pour un autre système d'exploitation), télécharger JDK 17, puis lancer l'exécutable d'installation. La version est **Java SE Development Kit 17.0.6**, qui s'installe par défaut dans C:\Program Files\Java\jdk-17.

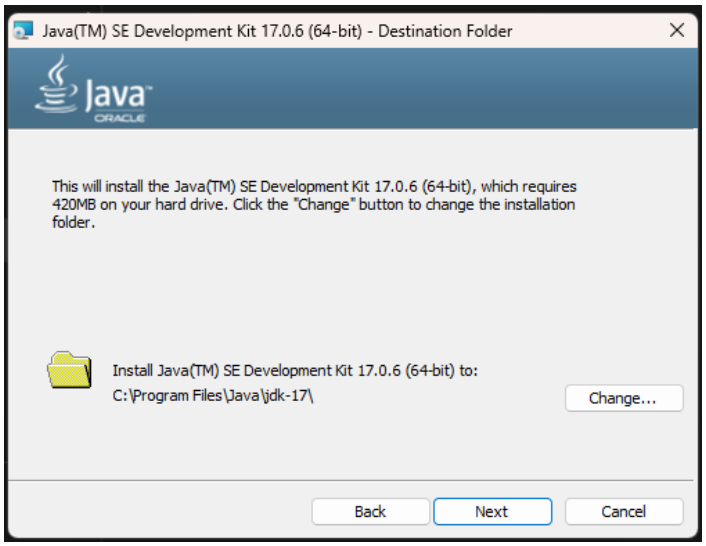

Fig. 1 - Assistant d'installation de JDK

Une manipulation importante doit généralement être faite ensuite pour pouvoir utiliser Java : la configuration de la variable d'environnement, qu'on appelle communément « path ». Pour cela, sous Windows, accéder aux propriétés système, puis aux variables d'environnement.

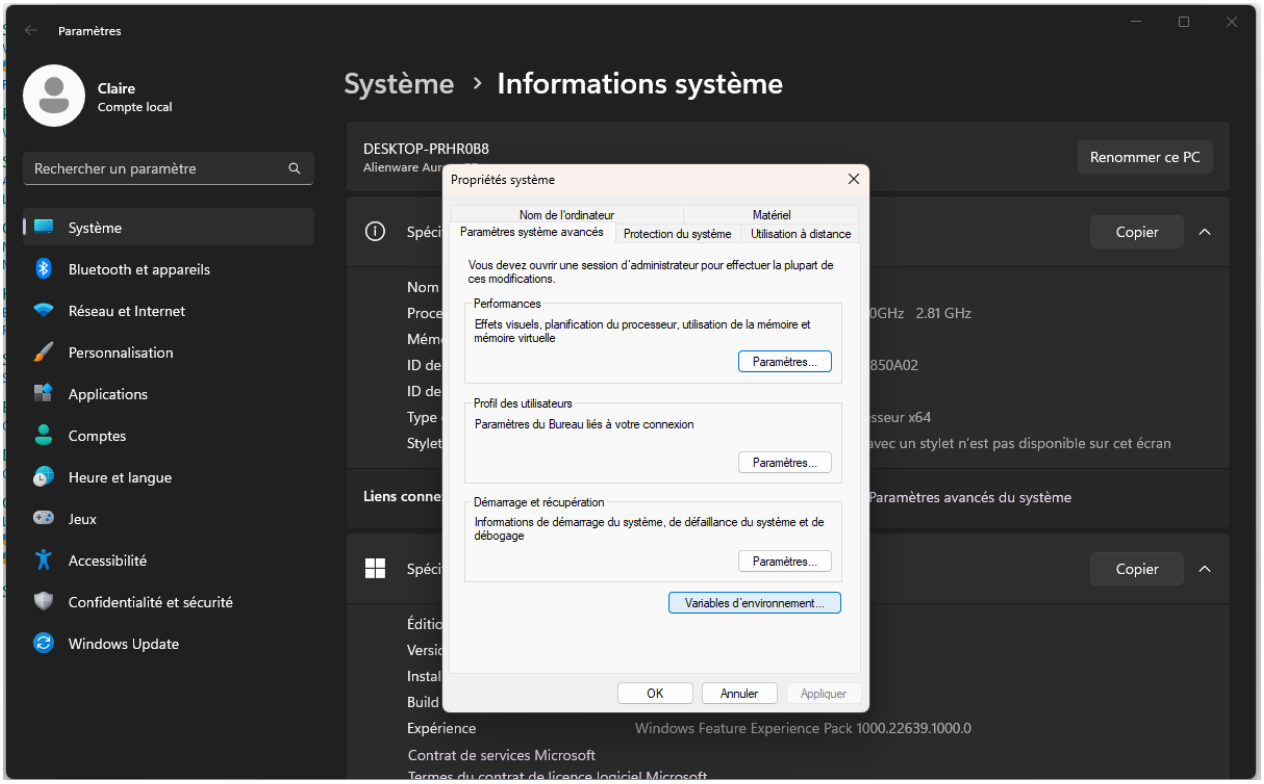

Fig. 2 - Fenêtre des propriétés systèmes de Windows 11

Dans la fenêtre Modifier la variable d'environnement, cliquer sur le bouton Nouveau, et ajouter le chemin absolu du dossier bin de JDK, de type C:\Program Files\Java\jdk-17\bin.

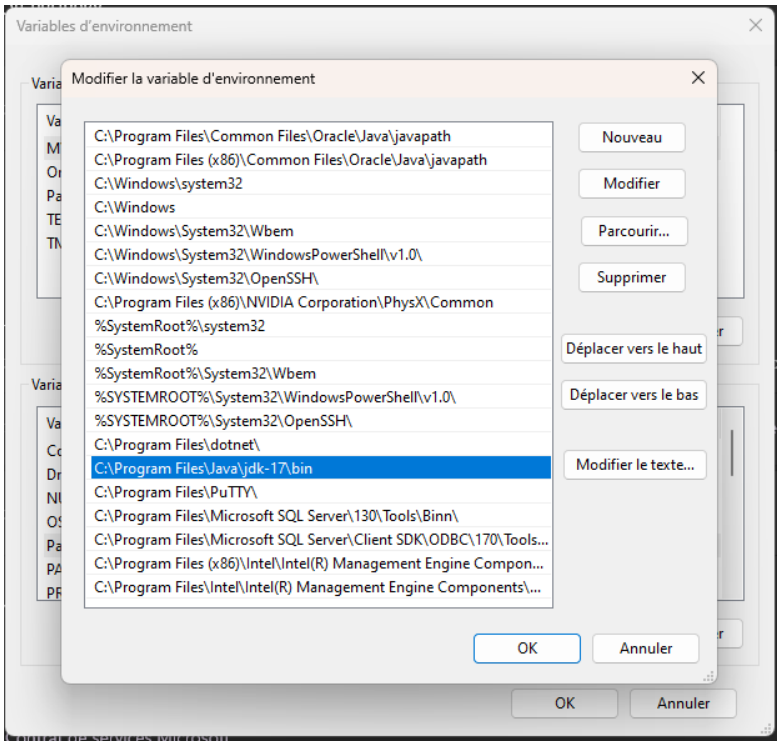

Fig. 3 - Modification de la variable d'environnement sous Windows 11

2.2. Installation de l'IDE Eclipse

Un IDE (Integrated Development Environment) est vivement recommandé pour écrire le code, et c'est Eclipse que j'ai choisi, téléchargeable sur https://eclipseide.org/. En l'occurrence, il s'agit de **Eclipse IDE 2023-03**. Eclipse fournit d'ailleurs un guide d'installation à l'adresse https://www.eclipse.org/downloads/packages/installer.

L'exécutable téléchargé, ici *eclipse-inst-jre-win64.exe*, propose différentes options d'installation. En l'occurrence, nous choisissons Eclipse IDE for Enterprise Java and Web Developers.

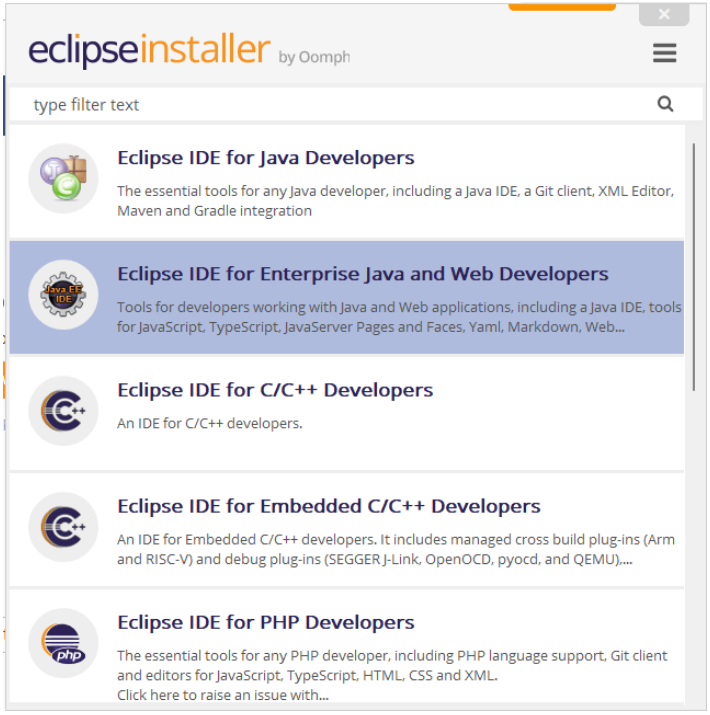

Fig. 4 - Choix de la version d'installation d'Eclipse

Par défaut, Eclipse s'installe dans le dossier racine de l'utilisateur.

En essayant de lancer Eclipse, on peut avoir l'erreur « Incompatible JVM », qui indique que la version actuelle de la machine virtuelle Java (en l'occurrence 1.8.0_351) n'est pas compatible avec cette version d'Eclipse.

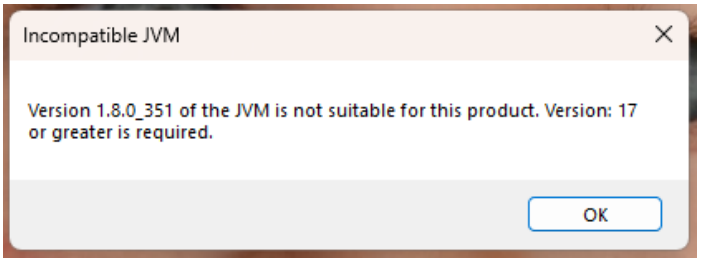

Fig. 5 - Erreur affichée si la version de JVM est incompatible avec Eclipse

La solution à ce problème se trouve à la page de la documentation Eclipse consacrée au fichier eclipse.ini : https://wiki.eclipse.org/Eclipse.ini. Trouver ce fichier dans le dossier d'installation du projet, et le modifier conformément aux instructions. Les lignes à insérer, qui précisent la version de JVM à utiliser, ne doivent pas être placées n'importe où. On insèrera donc, avant la ligne `-vmargs`, les deux lignes suivantes :

```
-vm
[chemin absolu de l'exécutable javaw.exe de la version de JDK à utiliser]
```

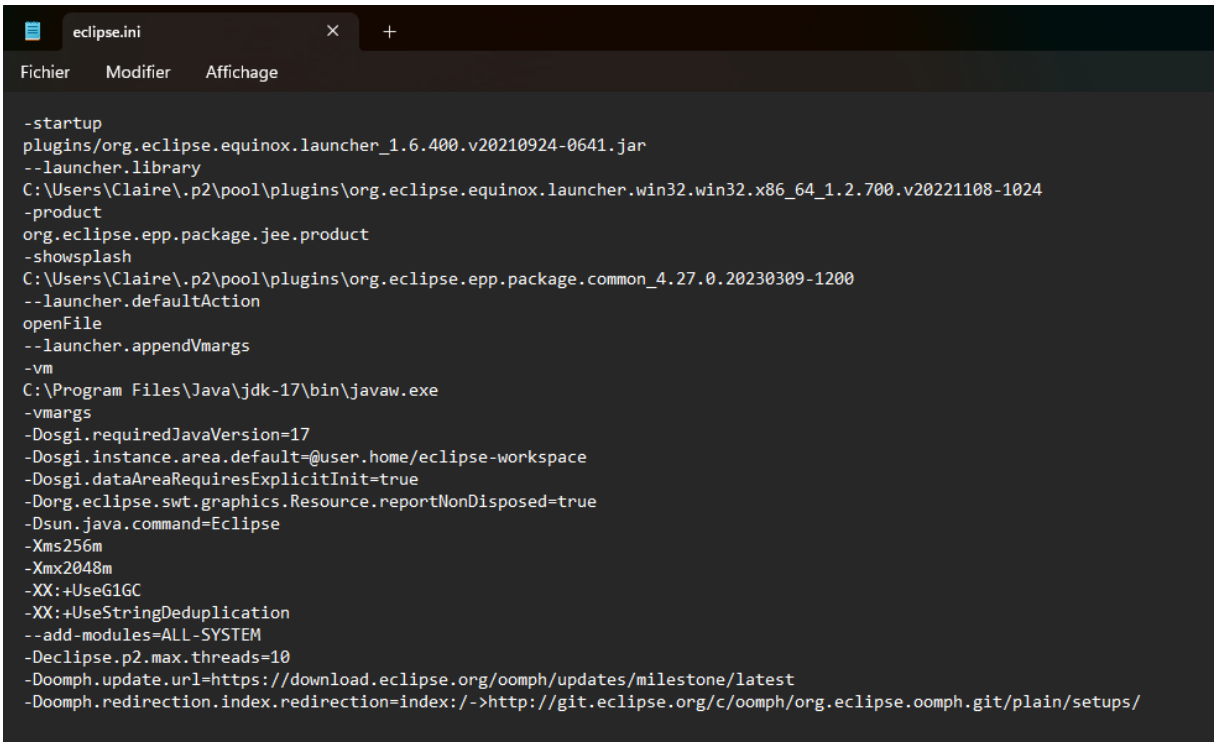

Fig. 6 - Fichier eclipse.ini après ajout des lignes précisant la JVM à utiliser

La version correcte de JVM étant spécifiée, Eclipse doit maintenant se lancer sans problème. Nous pouvons maintenant récupérer les sources du projet.

3. Téléchargement des fichiers sources du projet

Les fichiers sources du projet sont accessibles depuis le dépôt GitHub situé à l'adresse suivante :

https://github.com/Cerlia/Museum-full

Pour télécharger les sources, depuis la page ci-dessus, cliquer sur le bouton <> Code, puis sur **Download ZIP** pour télécharger d'un coup tous les fichiers du projet.

Noter que, si on voulait contribuer au projet, on créerait plutôt un clone du dépôt sur sa machine avec l'outil de versioning Git.

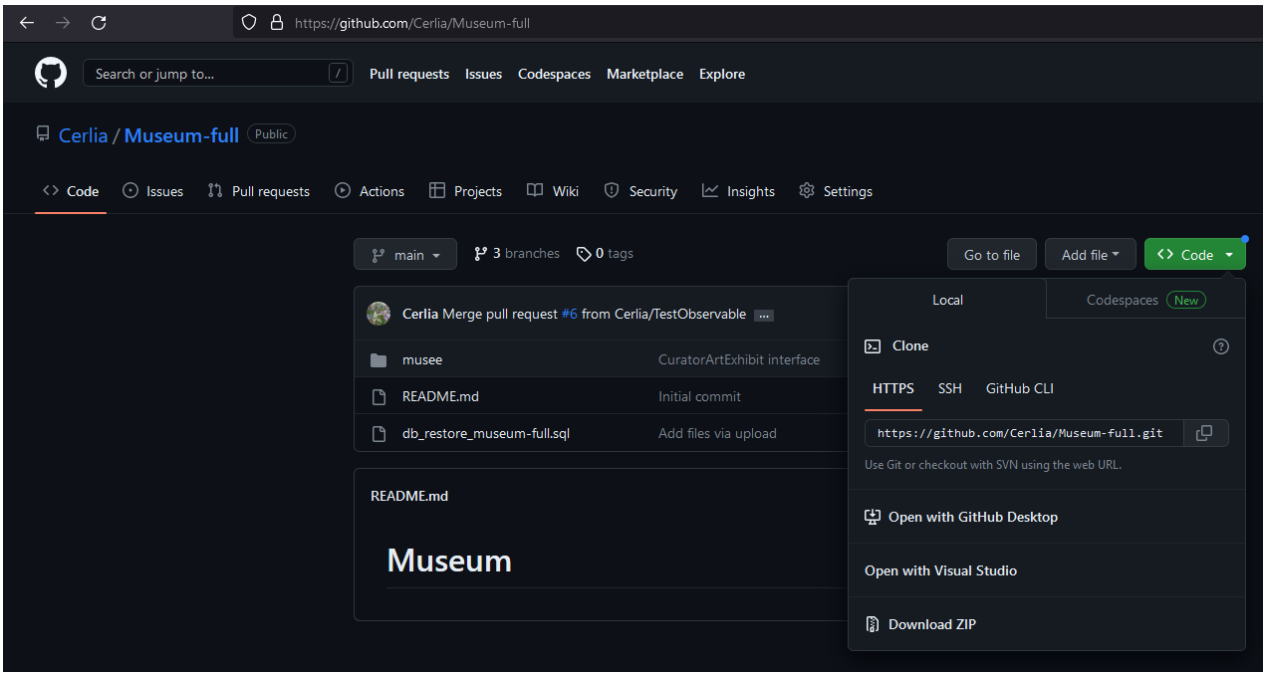

Fig. 7 - Téléchargement des fichiers du projet à partir de son dépôt sur GitHub

Dézipper l'archive à l'emplacement de son choix. Dans Eclipse, notre espace de travail n'a pas encore de projet. Eclipse suggère plusieurs manières de démarrer un projet, dont « Import Projects ». Choisir cette option, puis General, puis Projects from Folder or Archive. Sélectionner ensuite le chemin du dossier du projet, cocher le projet à importer, et choisir Finish.

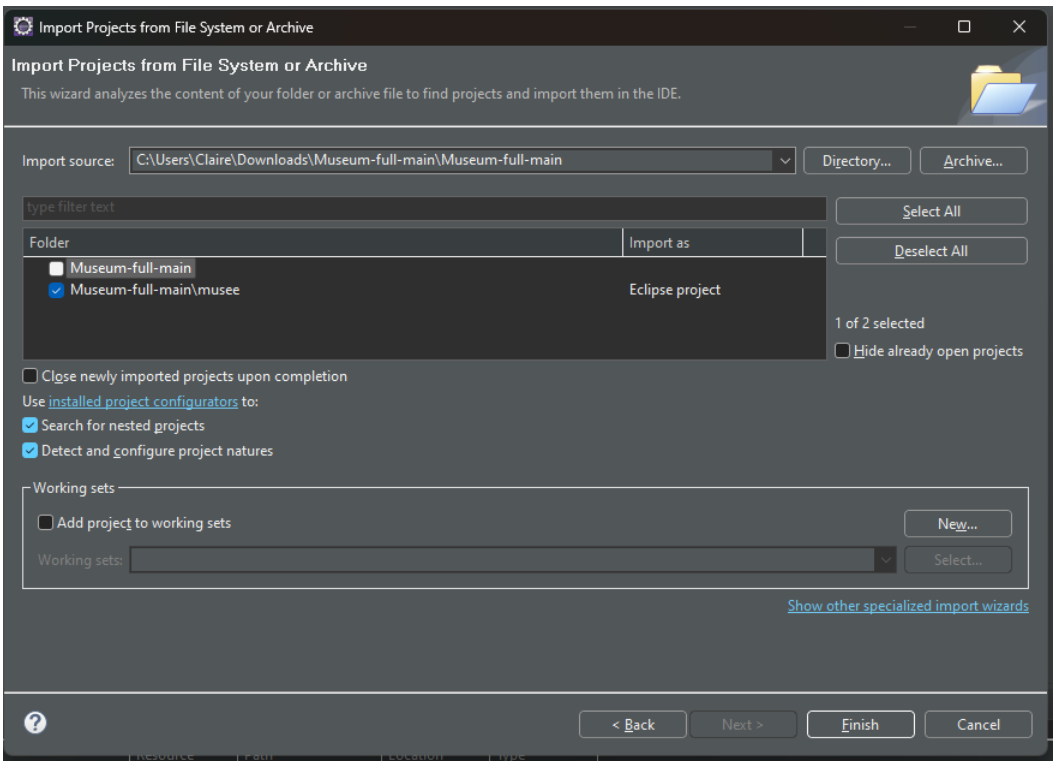

Fig. 8 - Importation dans Eclipse d'un projet existant à partir de son dossier racine

Le projet est maintenant ajouté à l'explorateur de projets d'Eclipse, dans le panneau de gauche. Eclipse signale immédiatement des problèmes avec le Build Path, c'est-à-dire le référencement des sources et bibliothèques nécessaires pour compiler le programme.

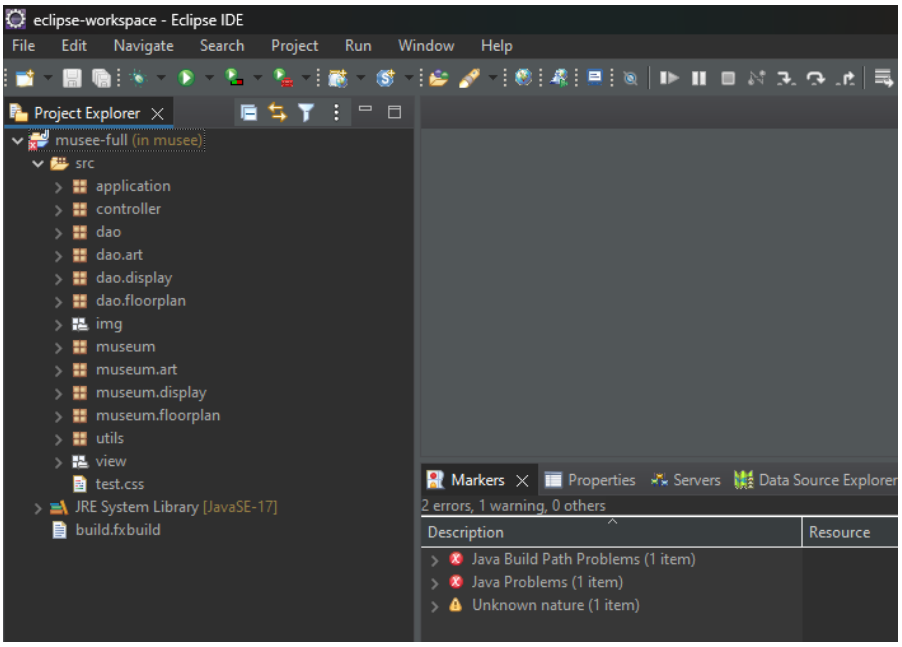

Fig. 9 - Signalement de problèmes avec le Java Build Path par Eclipse

Maintenant que le projet est importé, nous allons donc nous occuper de ce *build path*.

4. Gestion des dépendances du projet

Pour configurer le *build path* du projet, faire un clic droit sur le projet dans l'explorateur, puis choisir Build Path, et Configure Build Path…

On voit tout de suite qu'une bibliothèque est signalée manquante : sqljdbc42.jar. Celle-ci est une version de JDBC (Java DataBase Connectivity) pour SQL, dont le rôle sera d'interagir avec la base de données SQL.

4.1. Installation de JDBC

On peut trouver différentes versions de JDBC, et toutes ne sont pas compatibles avec ce projet. C'est pourquoi une version de JDBC est fournie dans le dépôt GitHub du projet. Le fichier sqljdbc42.jar doit simplement être placé sur le disque de l'ordinateur, à l'emplacement de son choix.

Une fois l'archive téléchargée, retourner à la fenêtre de configuration du *build path* dans Eclipse, et cliquer sur Add External JARs… Choisir le nouveau fichier sqljdbc42.jar, et valider. Le *modulepath* contient maintenant le bon fichier, et l'ancien peut être supprimé.

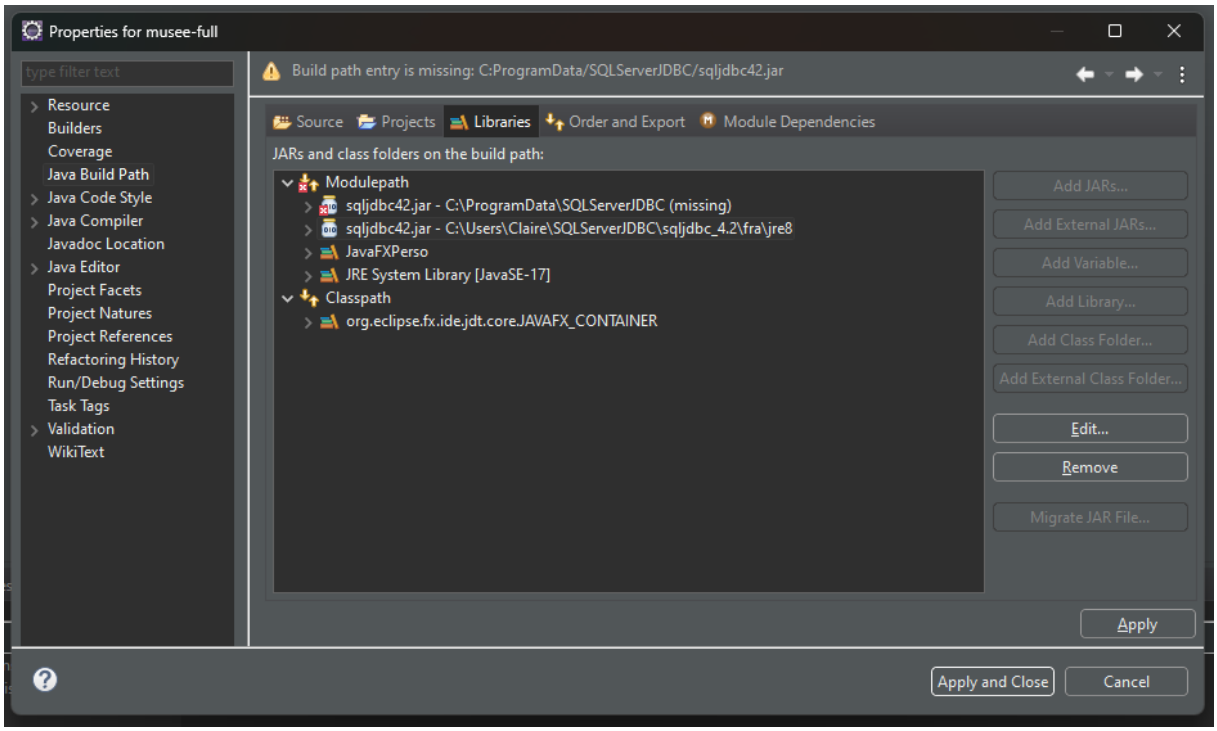

Fig. 10 - Modulepath du projet avant suppression du fichier sqljdbc42.jar manquant

Cliquer sur Apply and Close. Eclipse ne signale plus de problème avec le *build path*, mais de nombreux nouveaux problèmes sont apparus : des types ne sont pas reconnus. En effet, ce projet est un projet JavaFX, et il va nécessiter l'ajout d'une nouvelle bibliothèque spécialisée.

4.2. Installation de JavaFX

Depuis Java 8, JavaFX est la bibliothèque officielle pour les interfaces graphiques. L'installation peut se faire depuis Eclipse, dans le menu Help puis Install New Software…

Clic sur Add, et ajouter l'URL suivante : http://download.eclipse.org/efxclipse/updates-released/3.0.0/site

Lorsque cette adresse est entrée, Eclipse propose deux logiciels à installer : e(fx)clipse – install et e(fx)clipse – single components. Cocher les deux, et cliquer sur Next.

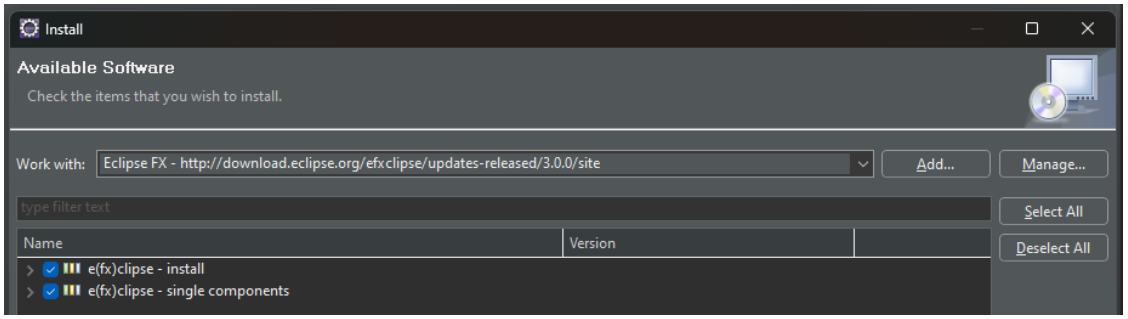

Fig. 11 - Installation de JavaFX dans Eclipse

Les écrans suivants montrent des détails et le contrat de licence. Cliquer sur Next pour terminer l'installation. Eclipse propose de redémarrer pour prendre en compte le nouveau logiciel installé. Il contient des bibliothèques qui vont permettre de développer un projet JavaFX, c'est-à-dire des classes dédiées, un éditeur CSS spécialisé, un outil d'édition de fichiers FXML…

Après redémarrage, les classes spécifiques à JavaFX dans le projet ne sont toujours pas reconnues car sa bibliothèque n'a pas encore été ajoutée au *build path* du projet. Cette bibliothèque doit être créée, et pour cela, on doit télécharger la dernière version de JavaFX sur le site **openjfx.io**.

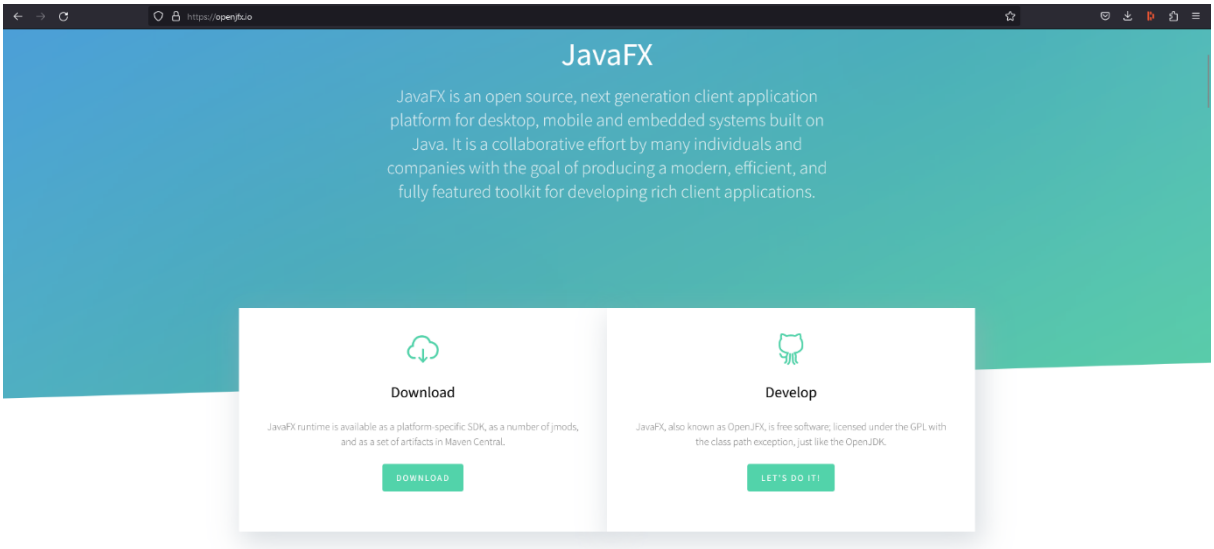

Fig. 12 - Le site openjfx.io permet d'obtenir la dernière version de JavaFX

Depuis cette page, on récupère sur gluonhq.com une archive zip (appelée dans mon cas openjfx-20_windows-x64_bin-sdk.zip) à dézipper dans un emplacement où on pourra le retrouver. Le dossier décompressé contient une petite dizaine de fichiers .jar dans son dossier lib.

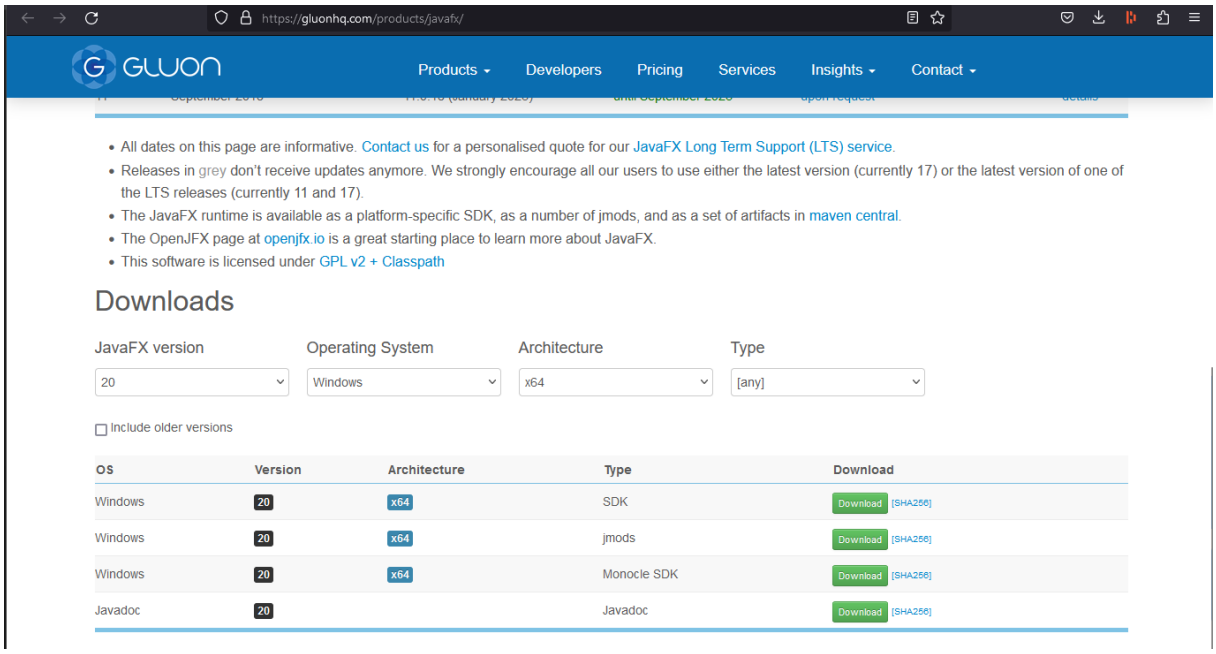

Fig. 13 - Téléchargement de la dernière version de JavaFX sur gluonhq.com

Retourner dans Eclipse, et dans la barre de menu, choisir Window > Preference > Java > Build Path > User Libraries. Cliquer sur New pour créer une nouvelle bibliothèque personnelle. On commence par lui donner un nom, en l'occurrence *JavaFXperso*. Puis on clique dessus, et sur Add External JARs pour y ajouter des bibliothèques. On choisit alors toutes les archives .jar qui se trouvent dans le dossier lib de l'archive qu'on vient de dézipper. Les archives sont ajoutées à la bibliothèque JavaFXperso, et on peut cliquer sur Apply and Close pour terminer.

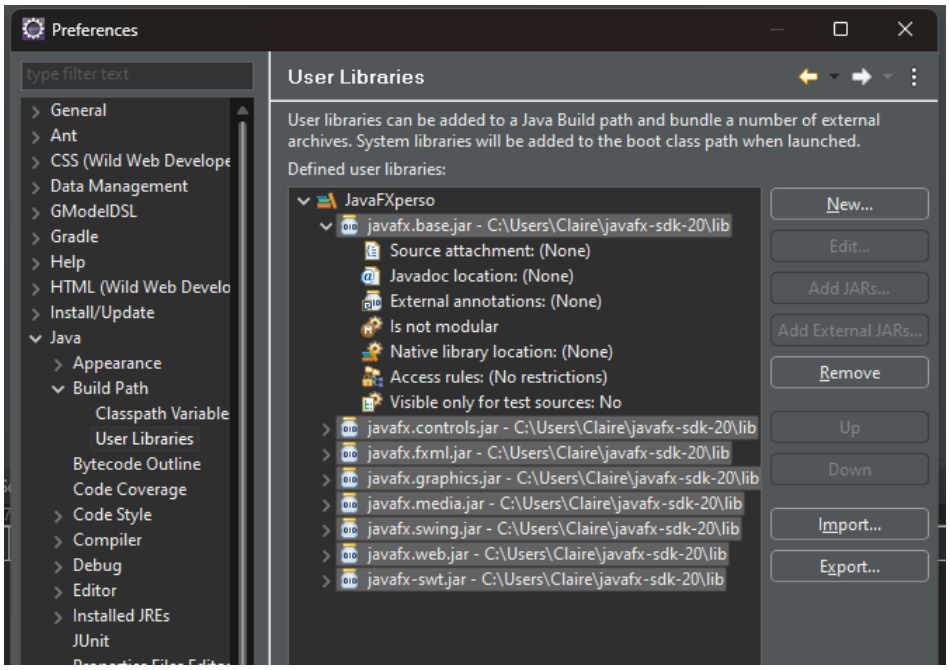

Fig. 14 - Ajout des bibliothèques JavaFX à une nouvelle bibliothèque personnelle

Les erreurs subsistent dans le projet, puisque cette nouvelle bibliothèque *JavaFXperso* n'a pas encore été ajoutée au *build path* du projet. De nouveau, faire un clic droit sur le dossier du projet dans l'explorateur, puis Build Path, Configure Build Path…, et supprimer la bibliothèque *JavaFXPerso* existante. Cliquer sur Add Library, choisir User Library, et sélectionner la nouvelle bibliothèque *JavaFXperso*.

Cette fois-ci, Eclipse ne signale plus d'erreurs de classes introuvables, les classes JavaFX sont bien reconnues et associées au projet. On peut alors tenter de lancer le projet, mais rien ne se passera car on n'a pas configuré les arguments à transmettre à la machine virtuelle Java au lancement du programme.

4.3. Paramétrage des arguments de la machine virtuelle Java

Aller dans le menu Run, puis Run Configurations…

Double-cliquer sur Java Application dans le menu de gauche pour configurer un nouveau projet. L'onglet Main définit le nom du projet (ici *musee-full*) et sa classe principale (ici *application.Main*). Passer ensuite à l'onglet Arguments. Le champ « VM arguments » est vide, et on va y ajouter les deux lignes suivantes :

```
--module-path "votre chemin vers \lib de votre dossier javafx"
--add-modules=javafx.controls,javafx.fxml
```

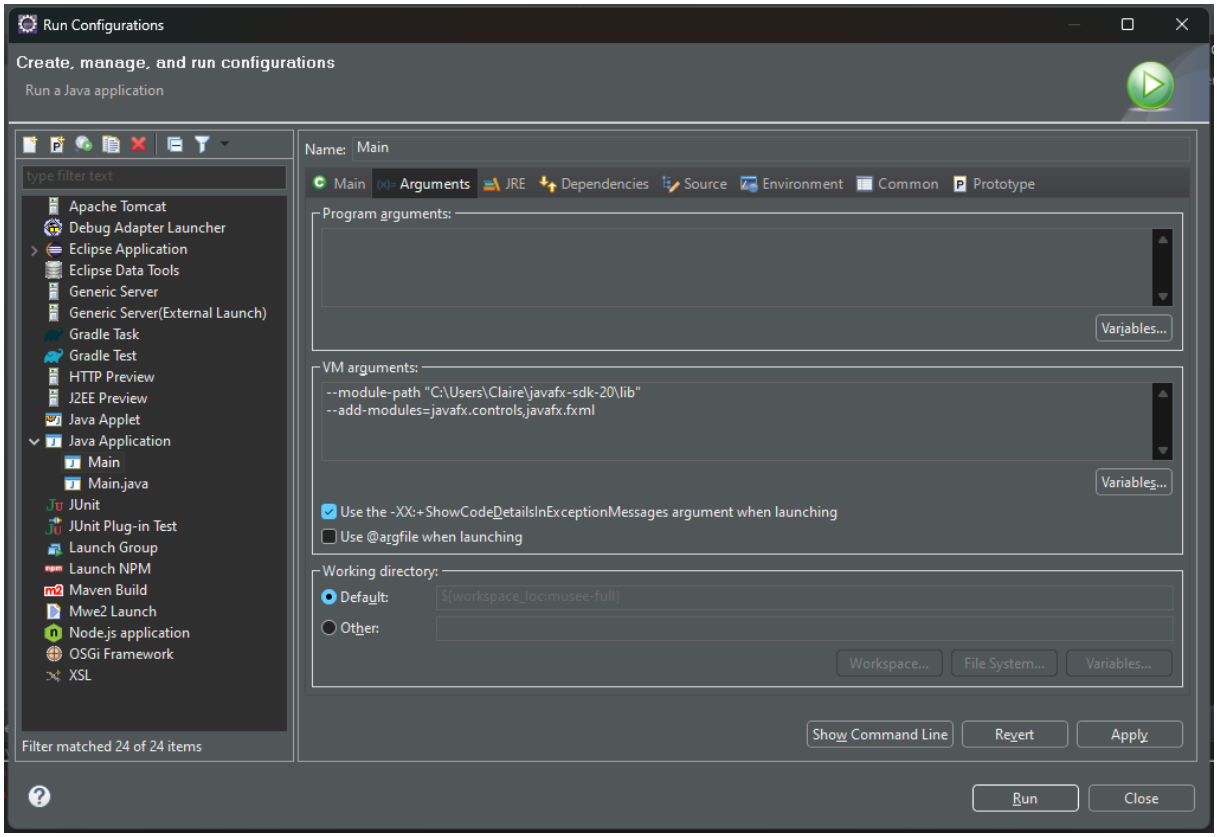

Fig. 15 - Ajout de paramètres de JVM permettant l'exécution du projet JavaFX

Cliquer sur Apply, puis Run. Cette fois-ci, l'application doit se lancer. La console va indiquer une erreur d'échec de connexion à la base de données située sur localhost. Il est temps de mettre en place la base de données.

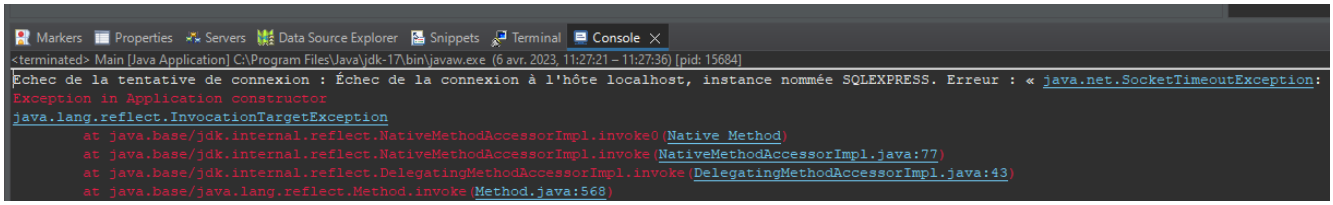

Fig. 16 - Erreur dans Eclipse : Echec de la connexion à l'hôte localhost

5. Mise en place de la base de données

5.1. Installation de SQL Server et Microsoft SQL Management Studio

Pour la base de données de ce projet, le formateur nous a recommandé SQL Server, téléchargeable sur le site de Microsoft. Plusieurs versions existent, nous choisissons la version gratuite **SQL Server 2022 Express**.

Lancer l'exécutable d'installation téléchargé (de type *SQL2022-SSEI-Expr.exe*), et choisir l'installation personnalisée. Après un certain temps, la fenêtre Centre d'installation SQL Server s'affiche, qui permet de gérer les installations.

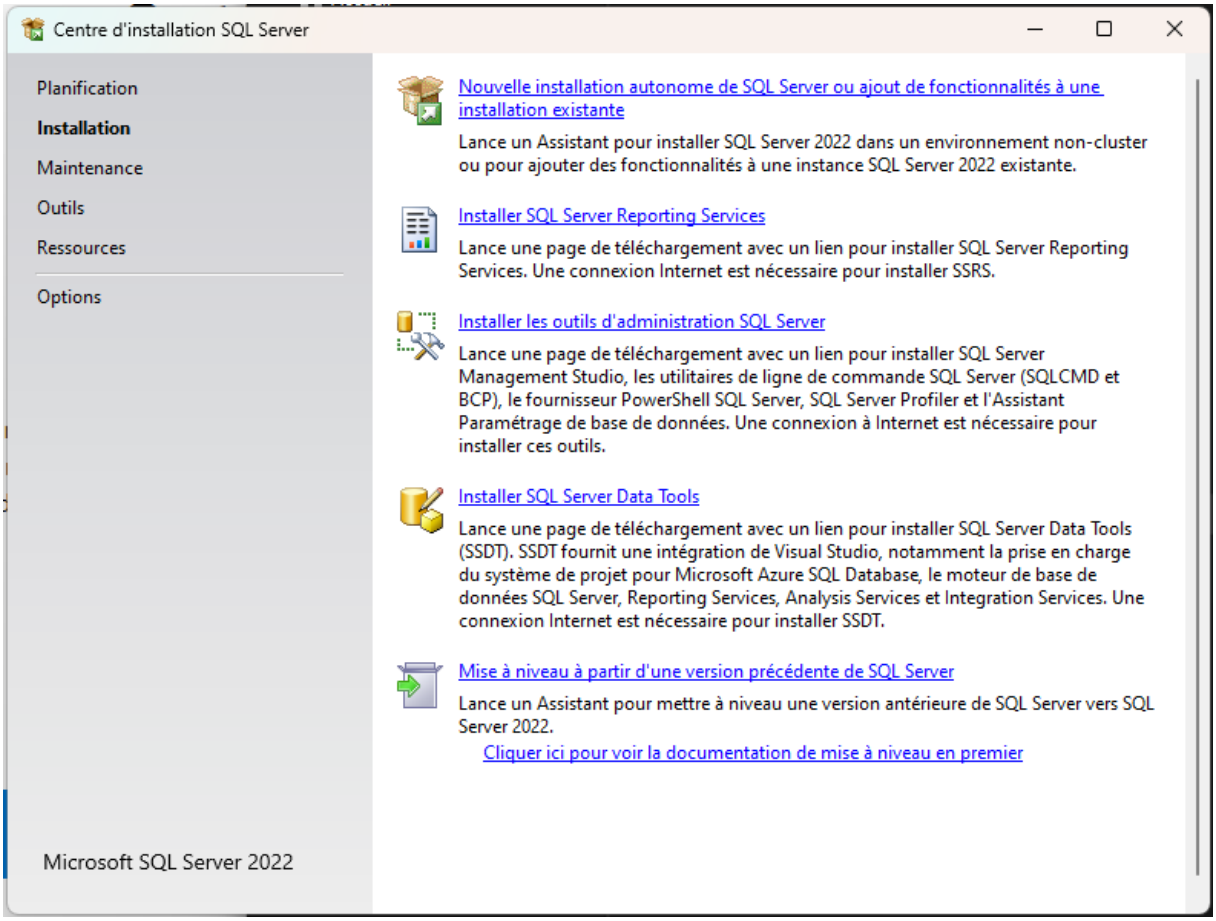

Fig. 17 - Centre d'installation SQL Server

Deux éléments vont nous intéresser dans cette liste. Cliquer d'abord sur le premier (« Nouvelle installation autonome… ») pour installer SQL Server en lui-même.

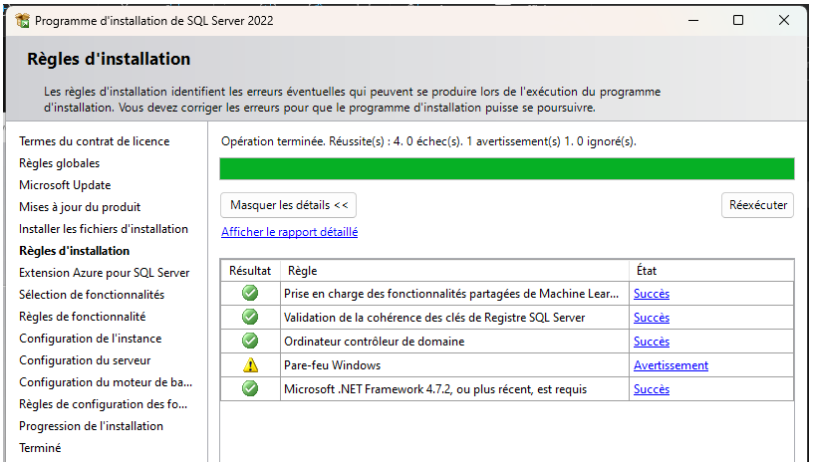

Fig. 18 - Résultat de l'installation de SQL Server

On peut au passage remarquer un avertissement concernant le pare-feu Windows, avertissement dont on se préoccupera un peu plus tard, au point 6.

L'installation de SQL Server se poursuit. Dans Sélection de fonctionnalités, vérifier que « Services Moteur de base de données » est bien coché. L'instance de SQL Server doit ensuite être nommée, par défaut « SQLExpress ». Enfin, on configure le moteur de base de données. Pour l'instant, on garde les paramètres par défaut (Mode d'authentification Windows uniquement).

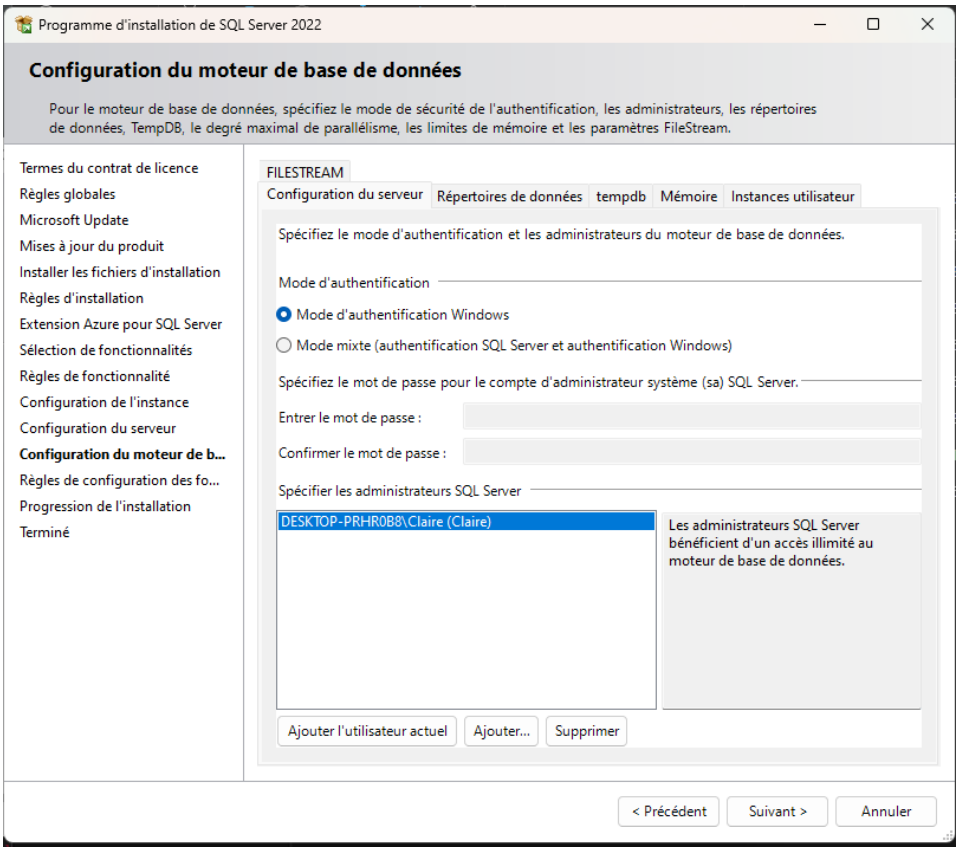

Fig. 19 - Configuration du moteur de base de données SQL Server

Poursuivre l'installation jusqu'au bout pour terminer l'installation de SQL Server.

Depuis le centre d'installation SQL Server (figure 17), choisir maintenant « Installer les outils d'administration SQL Server ». Cela ouvre une page dans le navigateur, depuis laquelle on va télécharger **Microsoft SQL Server Management Studio**. Cet outil va permettre d'administrer graphiquement nos bases de données SQL Server.

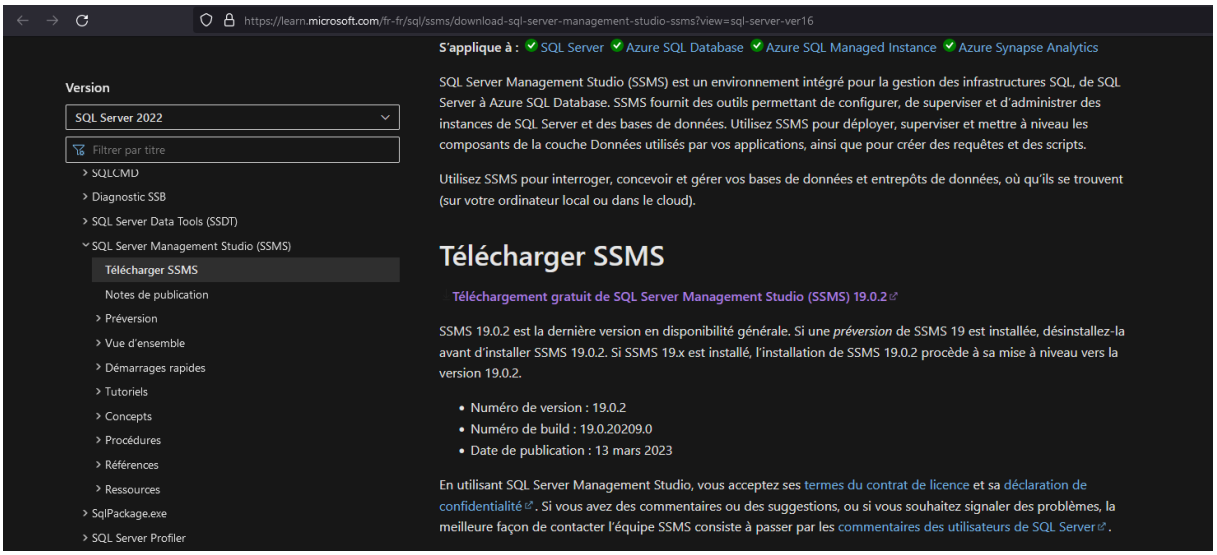

Fig. 20 - Extrait de la page du site Microsoft dédiée au téléchargement de SQL Server Management Studio

Télécharger l'exécutable (de type *SSMS-Setup-ENU.exe*), puis procéder à l'installation dans le répertoire de son choix ou celui défini par défaut.

Une fois l'installation terminée, lancer le programme (on peut par exemple chercher « ssms » dans la barre de recherche de Windows 11). On peut avoir à choisir, au premier lancement, le type de serveur. Dans le champ Server type, sélectionner Database Engine. Parcourir ensuite les serveurs existants pour sélectionner SQLEXPRESS.

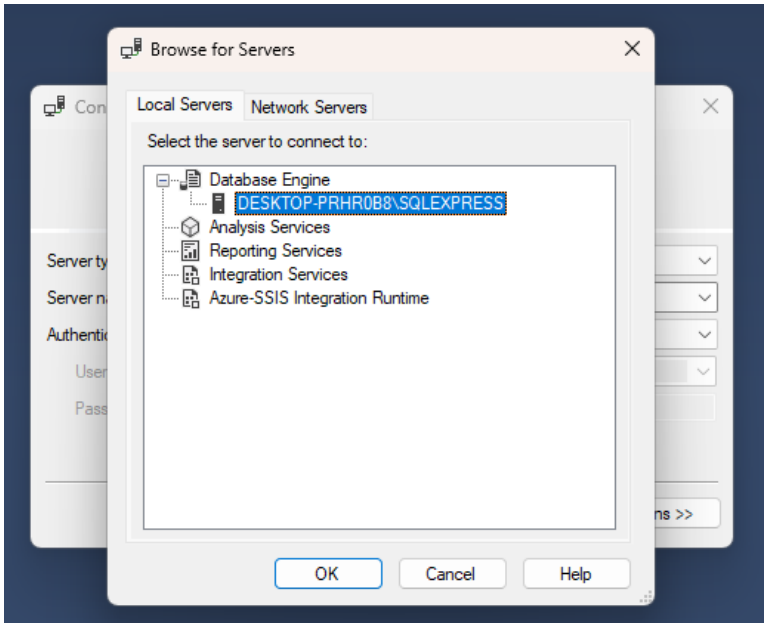

Fig. 21 - Fenêtre de choix du serveur dans SQL Server Management Studio

Il suffit ensuite de s'authentifier avec Windows (il n'y a en principe rien de spécial à faire), et on doit avoir accès à l'interface de gestion des bases de données.

5.2. Restauration de la base de données du projet

Le fichier de restauration de la base de données, appelé *db-restore-museum-full.bacpac*, se trouve dans le dépôt du projet sur GitHub et a donc dû être téléchargé dans l'archive zip à l'étape 3.

Pour créer la base de données dans SQL Server Management Studio, aller dans l'explorateur de serveur, à gauche de l'écran. Sous le nom du serveur, faire un clic droit sur Databases, puis Import Data-tier Application… Choisir le fichier bacpac téléchargé, puis renommer la base de données, idéalement « museum-full ».

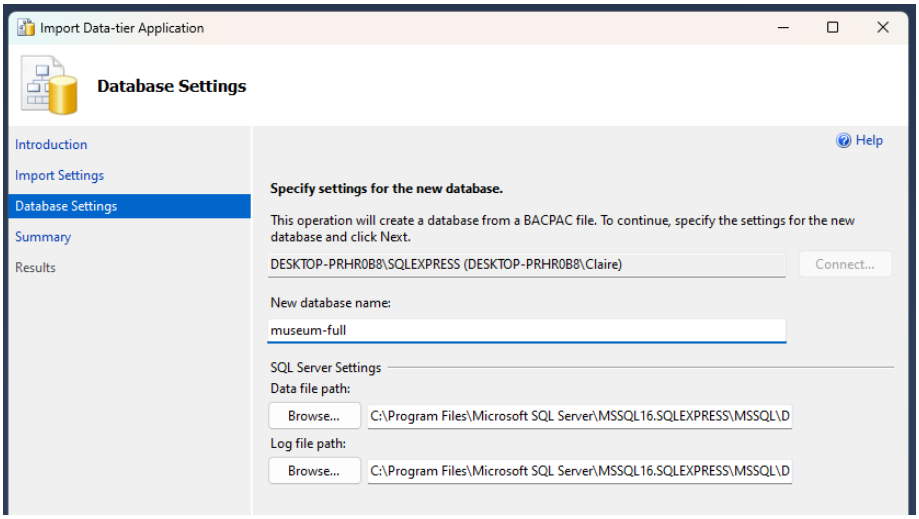

Fig. 22 - Importation d'une base de données dans SQL Server Management Studio : renommage de la base de données

La base de données est alors importée, les tables sont créées avec succès.

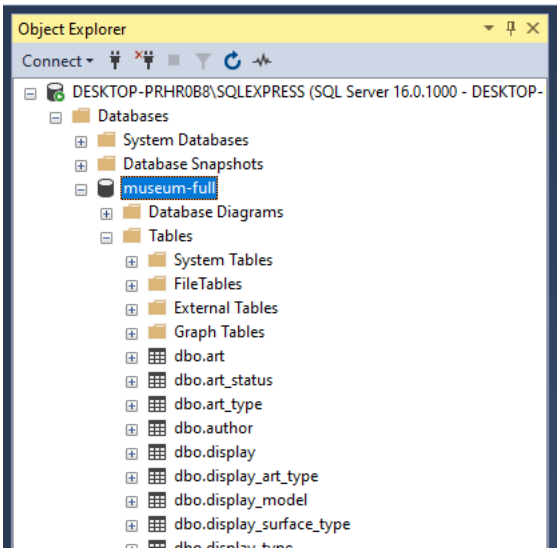

Fig. 23 - Création de la base de données museum-full effectuée

6. Connexion à la base de données

Pour connecter le projet Java à la base de données qui vient d'être créée, plusieurs choses sont à vérifier ou mettre en place :
- configurer la classe du projet chargée de la connexion,
- côté base de données, créer des identifiants permettant d'avoir accès à la base de données,
- divers petits paramétrages : autoriser TCP/IP et paramétrer le lancement du service SQL Server.

6.1. Connexion à la base de données depuis l'application

Dans Eclipse, ouvrir le fichier src/dao/Connect.java. C'est une classe singleton qui s'occupe de la connexion à la base de données. On y spécifie dans des constantes, une fois pour tout le projet, l'identifiant (ID) et le mode de passe (MDP) de l'utilisateur de la base de données, en l'occurrence le couple « museum / admin ». Ces informations de connexion sont évidemment peu sécurisées et ne devraient jamais être utilisées dans un cas réel.

Connect.java

```
1  package dao;
2
3  import java.sql.Connection;
9
10 public class Connect {
11
12     private static Connection connection = null;
13
14     private static final String ID = "museum";
15     private static final String MDP = "admin";
16
17     /**
18      * Patron de conception Singleton
19      * @return l'instance unique de connexion
20      */
21     public static Connection getInstance() {
22         if (connection==null) {
23             try {
24                 SQLServerDataSource ds = new SQLServerDataSource();
25                 ds.setUser(ID);
26                 ds.setPassword(MDP);
27                 ds.setServerName("localhost\\SQLEXPRESS");
28                 ds.setDatabaseName("museum-full");
29                 connection = ds.getConnection();
30                 System.out.println("connecté");
31             }
32             catch (SQLException e){
33                 System.out.println("Echec de la tentative de connexion : " + e.getMessage() + e.getStackTrace()) ;
34             }
35         }
36         return connection;
37     }
38
39     private Connect() {
40         super();
41     }
```

Fig. 24 - Classe Connect.java de connexion à la base de données

Noter aussi le nom du serveur (« localhost\\SQLEXPRESS ») qui doit normalement correspondre à l'instance créée précédemment, et celui de la base de données (« museum-full »), qui doit absolument correspondre au nom choisi lors de l'importation de la base.

Nous allons maintenant vérifier que les informations de connexion de la classe Connect.java correspondent aux paramètres de la base de données.

6.2. Paramétrage dans SQL Server Management Studio

Revenir dans SQL Server Management Studio. Lors de la création de la base de données à partir du fichier .bacpac, un utilisateur nommé *museum* a dû être créé automatiquement. Cependant, ses paramètres ne sont probablement pas corrects.

Pour paramétrer un utilisateur de la base de données, aller dans l'explorateur de serveur, et sous le serveur concerné trouver Security, Logins, puis faire un clic droit sur *museum* et choisir Properties pour ouvrir les paramètres.

Dans l'onglet General, vérifier que le mode de connexion sélectionné est bien SQL Server Authentication. Entrer et confirmer le mot de passe (« admin ») et décocher Enforce password policy.

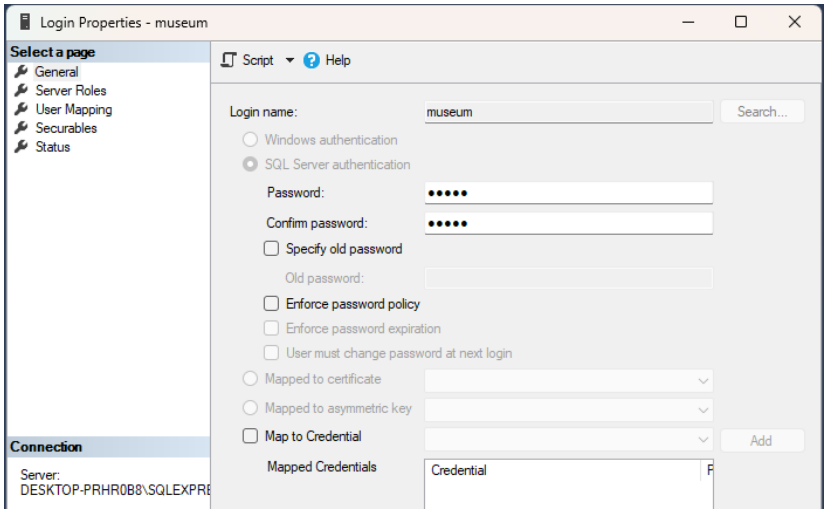

Fig. 25 - Propriétés de l'identifiant 'museum', onglet General

Dans l'onglet User Mapping, associer la bases de données *museum-full* à l'utilisateur. Il doit déjà avoir le role *public*, il faut lui ajouter le rôle *db_owner*.

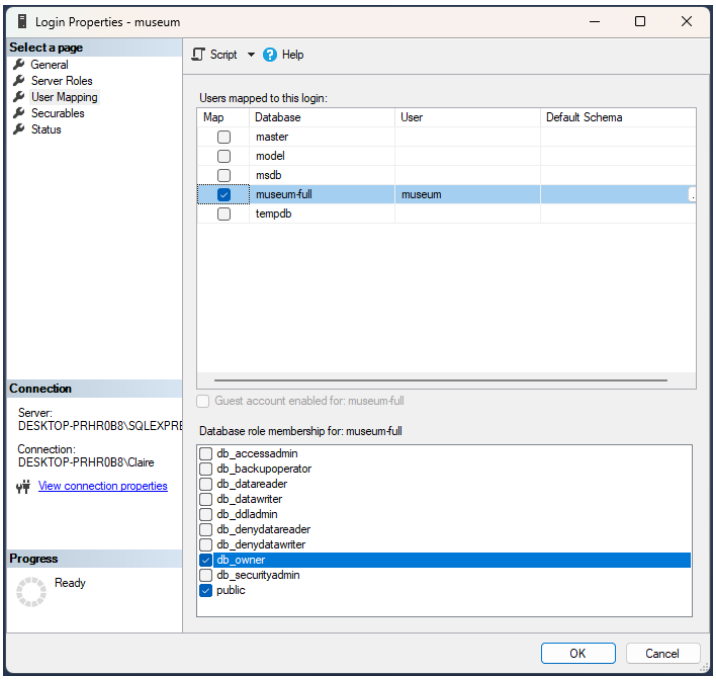

Fig. 26 - Propriétés de l'identifiant 'museum', onglet User Mapping

Il reste à autoriser l'accès à la base de données par une application. Dans l'explorateur, faire un clic droit sur le nom du serveur, puis Security. Cocher **SQL Server and Windows Authentication mode** pour activer l'authentification autre que Windows.

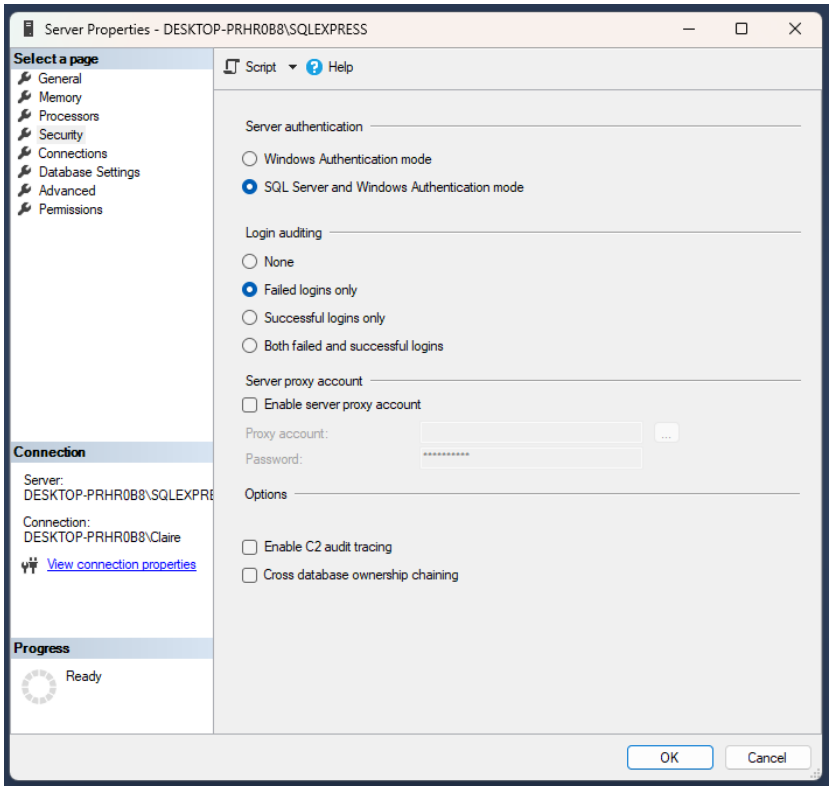

Fig. 27 - Paramètres de sécurité du serveur

SSMS nous avertit qu'il faut redémarrer le serveur. Faire un clic droit sur le nom du serveur et sélectionner Restart. Le lancement de l'application doit en principe encore échouer.

6.3. Paramétrage additionnel

Dans la barre de recherche d'application de Windows, taper Services. Trouver SQL Server dans la liste, et choisir le démarrage manuel. Faire la même chose pour SQL Server Browser, puis redémarrer ces deux services manuellement.

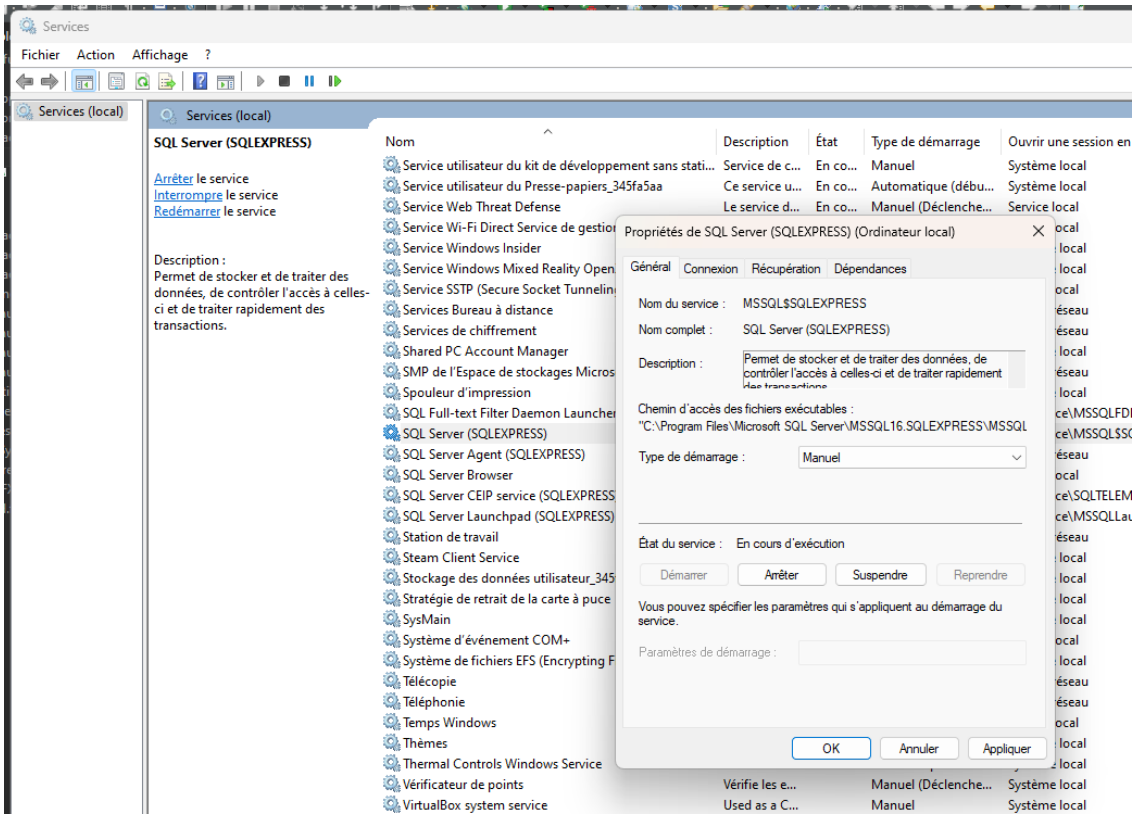

Fig. 28 - Modification du type de démarrage de SQL Server dans les services de Windows

Essayer à nouveau de lancer l'application : on peut obtenir une nouvelle erreur : « Le serveur SQLEXPRESS n'est pas configuré pour écouter avec TCP/IP ».

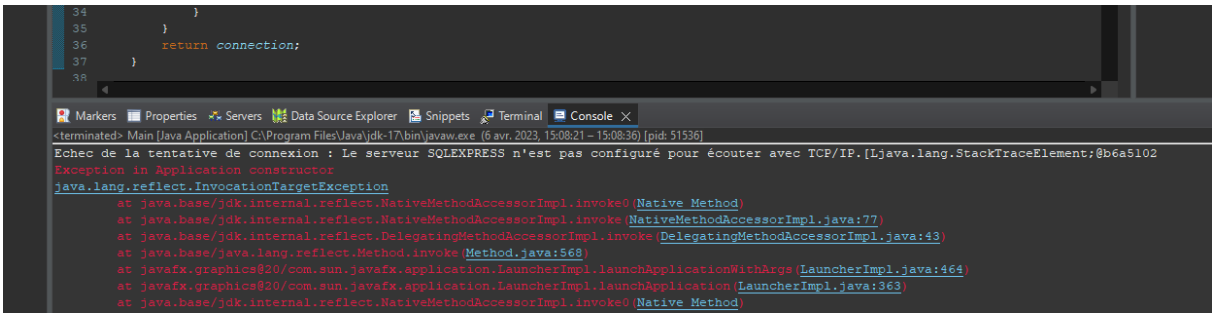

Fig. 29 - Erreur de connexion : « Le serveur SQLEXPRESS n'est pas configuré pour écouter avec TCP/IP »

Au point 5.1, figure 18, on avait vu un avertissement concernant le pare-feu de Windows. En effet, le protocole TCP/IP n'est pas activé par défaut pour SQLEXPRESS, ce qui rend le service inaccessible.

Pour l'activer, le plus simple est de passer par le gestionnaire de configuration SQL Server. Il peut être difficile à trouver. Un raccourci peut exister dans le dossier d'installation de Microsoft SQL Server, mais le fichier permettant d'ouvrir la console doit s'appeler SQLServerManager16.msc (les chiffres finaux pouvant varier suivant la version), situé dans C:\WINDOWS\SysWOW64.

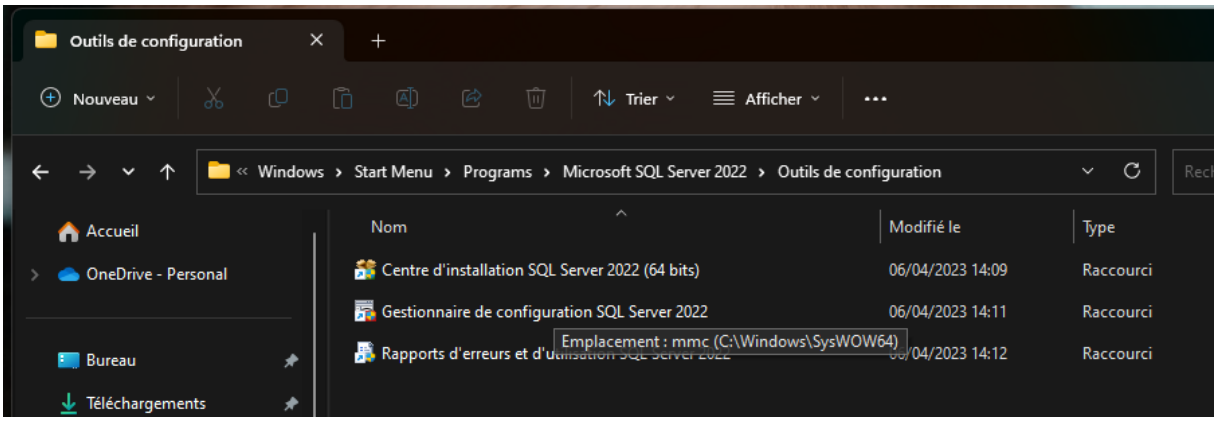

Fig. 30 - Raccourci vers la console Gestionnaire de configuration SQL Server

Dans la console Sql Server Configuration Manager, trouver le menu Protocoles pour SQLEXPRESS. Le protocole TCP/IP est désactivé par défaut, il faut donc l'activer. On doit ensuite redémarrer le service, donc fermer ce gestionnaire de configuration, retourner dans les services de Windows, et redémarrer manuellement SQL Server.

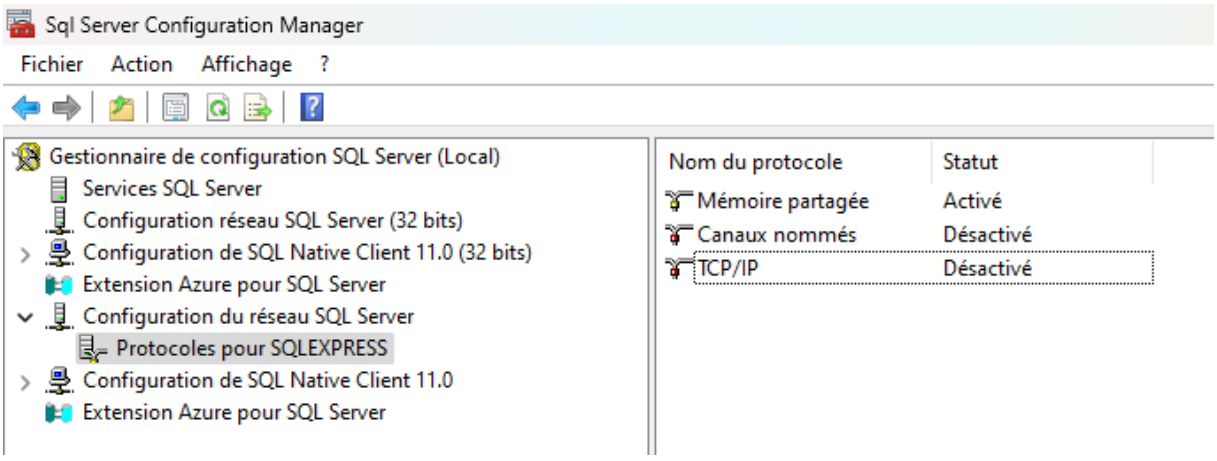

Fig. 31 - Activation du protocole TCP/IP dans Sql Server Configuration Manager

Dans Eclipse, essayer à nouveau d'exécuter le programme *musee-full*. Le programme doit maintenant se lancer correctement et pouvoir accéder à la base de données.

7. Optionnel : installation de SceneBuilder

Il est tout à fait possible de développer un projet JavaFX sans outil de visualisation des interfaces graphiques créées. Cependant, on peut aussi utiliser un logiciel tel que SceneBuilder, qui permet de créer les interfaces de manière visuelle, et non uniquement textuelle.

Comme JavaFX, SceneBuilder peut être téléchargé sur le site www.gluonhq.com.

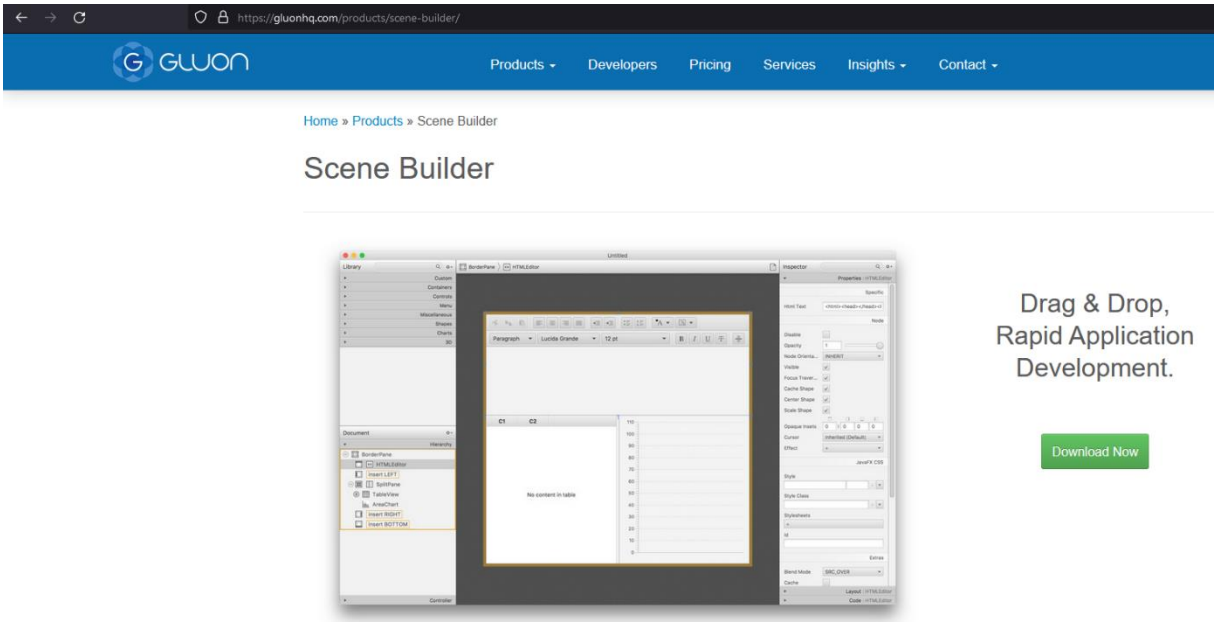

Fig. 32 - Téléchargement de SceneBuilder sur le site gluonhq.com

Télécharger la version correspondante à sa machine, puis installer le programme en notant bien le répertoire d'installation.

Dans Eclipse, aller dans le menu Window, puis Preferences. Dans le menu de gauche, trouver et cliquer sur JavaFX, et renseigner le chemin de l'exécutable de SceneBuilder.

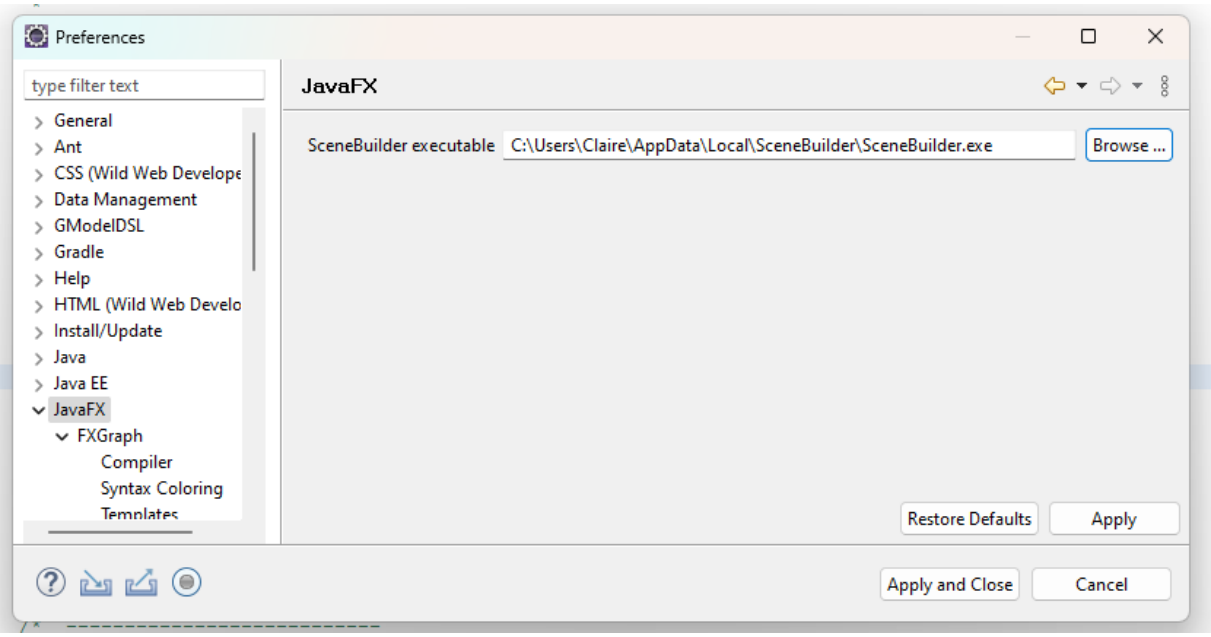

Fig. 33 - Renseignement du chemin de l'exécutable de SceneBuilder dans Eclipse

Une fois les paramètres sauvegardés, faire un clic droit sur un des fichiers .fxml du dossier view du projet, et cliquer sur Open with SceneBuilder. Celui-ci doit s'ouvrir et afficher l'interface de modification du fichier.

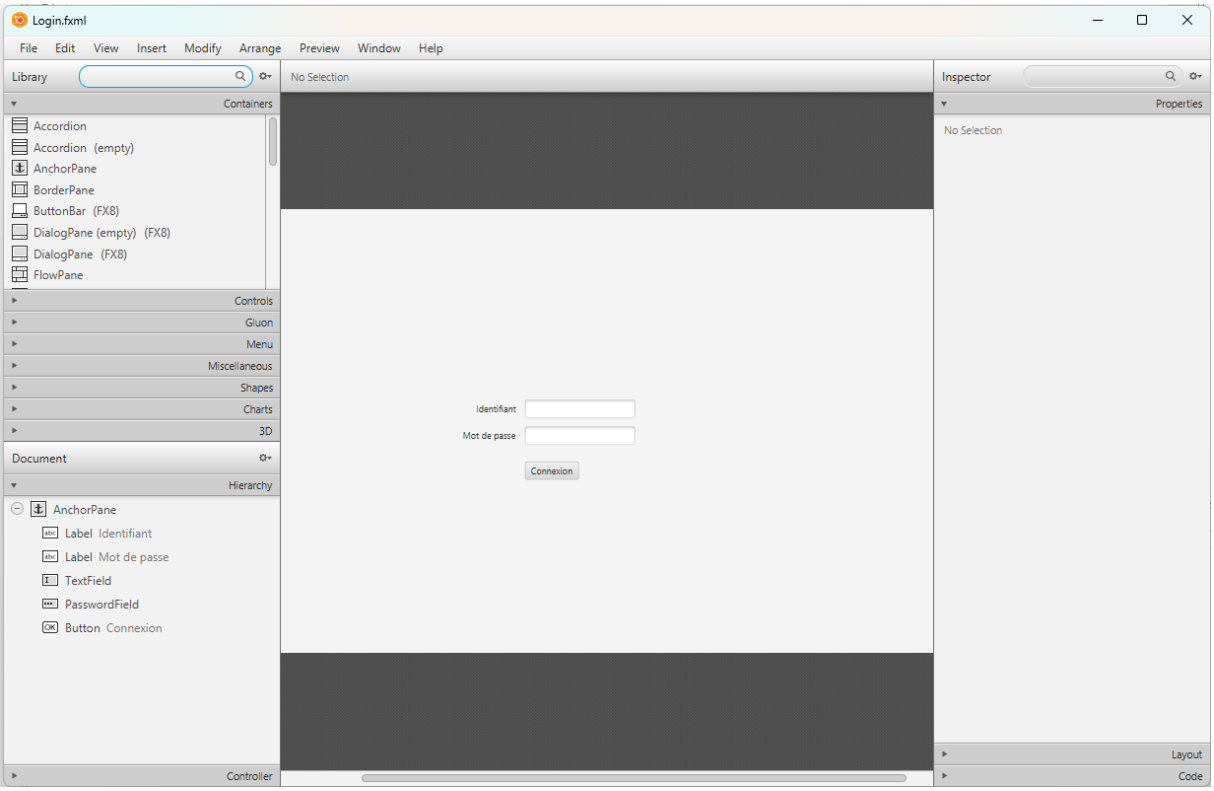

Fig. 34 - Fichier Login.fxml ouvert avec SceneBuilder