

Tutoriel Accès à une Base de Données MySQL avec l'API JDBC java.sql

NB :

Pour les TP de JDBC j'ai deux exemple celui de mon projet perso et les exemples du site Objis mais pour ce TP je préféré présenter uniquement les tutorial d'exemple sur le site de Objis mais personnaliser.

J'ai Présenter des exemples de diagramme de mon projet perso

Le code complété est sur GitHub.

Lien GitHub TP Objis : <https://github.com/IMANA47/DemoJdbc>

Lien GitHub Projet Perso : <https://github.com/IMANA47/TP8-17-Mai-2025>

Objectif :

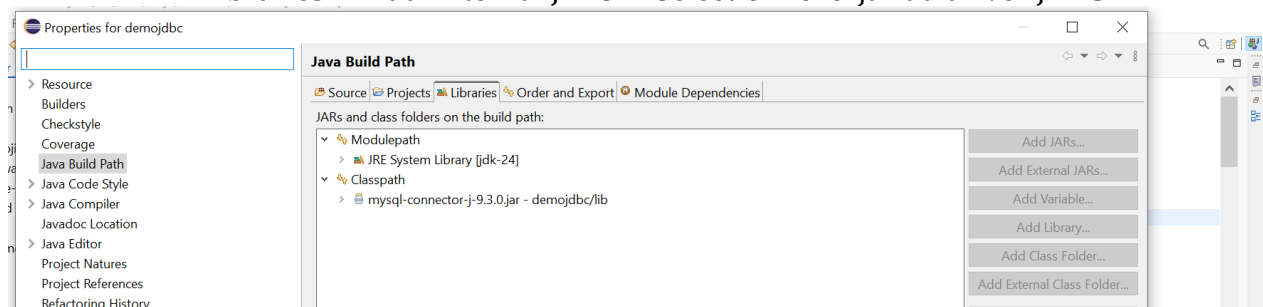
L'objectif de ce tutoriel est de comprendre et de mettre en œuvre l'API JDBC (java.sql) pour interagir avec une base de données MySQL en Java. Nous allons explorer les différentes étapes pour insérer et récupérer des données à travers une base de données, en utilisant des exemples de mon projet une petit parti.

Outils utilisé

- Eclipse IDE : Pour développer et exécuter le code Java.
- MySQL Workbench CE : Pour la gestion de ta base de données et l'exécution de requêtes SQL.
- JDBC MySQL Connector : Le driver JDBC de MySQL pour connecter Java à la base de données.

Ajout du JDBC dans Eclipse

- J'ai téléchargé le fichier .jar du driver JDBC (par exemple, mysql-connector-j-9.3.0.jar).
- Ajoute mon projet sous Eclipse :
 - Fais un clic droit sur ton projet → Build Path → Configure Build Path → Onglet Libraries → Add External JARs → Sélectionne le .jar du driver JDBC.



2 Création de la Base de Données MySQL

Création de la base de données flowatch

Cette base contiendra trois tables pour exemple :

1. Utilisateurs
2. vehicules
3. conducteurs

Script SQL pour la création et insertion des données

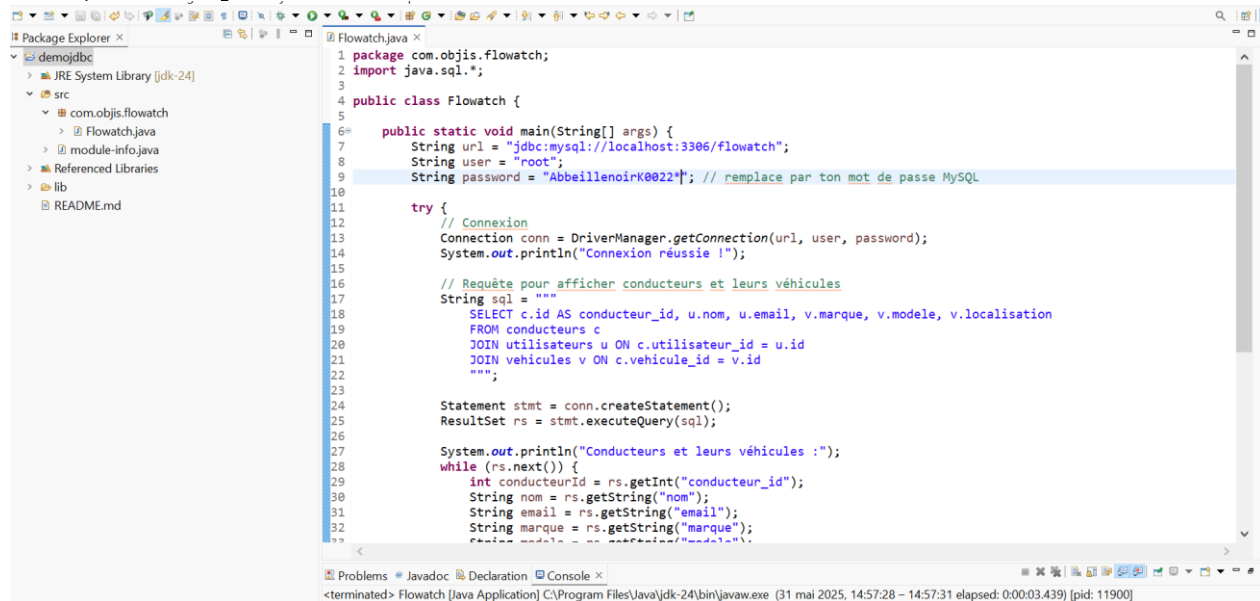
```
1 -- Crée la base
2 CREATE DATABASE IF NOT EXISTS flowatch;
3 USE flowatch;
4
5 -- Table des utilisateurs
6 CREATE TABLE IF NOT EXISTS utilisateurs (
7     id INT AUTO_INCREMENT PRIMARY KEY,
8     nom VARCHAR(50),
9     email VARCHAR(100)
10 );
11
12 -- Table des véhicules
13 CREATE TABLE IF NOT EXISTS vehicules (
14     id INT AUTO_INCREMENT PRIMARY KEY,
15     marque VARCHAR(50),
16     modele VARCHAR(50),
17     localisation VARCHAR(100)
18 );
19
20 -- Table des conducteurs (liens entre utilisateurs et véhicules)
21 CREATE TABLE IF NOT EXISTS conducteurs (
22     id INT AUTO_INCREMENT PRIMARY KEY,
23     utilisateur_id INT,
24     vehicule_id INT,
25     FOREIGN KEY (utilisateur_id) REFERENCES utilisateurs(id),
26     FOREIGN KEY (vehicule_id) REFERENCES vehicules(id)
27 );
28
29 -- Données de test
30 INSERT INTO utilisateurs (nom, email) VALUES
31 ('Oko Dani', 'okodani@gmail.com'),
32 ('Oko Dani', 'okodani@gmail.com'),
33 ('Oko Dani', 'okodani@gmail.com');
```

3 Connexion JDBC et Insertion de Données

Les etapes pour insérer des données avec JDBC :

1. Charger le driver JDBC : Utiliser `Class.forName("com.mysql.cj.jdbc.Driver");` pour charger le driver JDBC.
2. Établir une connexion à la base de données : Utiliser `DriverManager.getConnection(url, user, password);`.
3. Préparer la requête SQL : Créer un `PreparedStatement` pour insérer des données.
4. Exécuter la requête SQL : Utiliser `executeUpdate()` pour insérer les données.
5. Fermer la connexion : Il faut toujours fermer la connexion pour éviter les fuites de mémoire.

Code Java exemple :



```
1 package com.objis.flowatch;
2 import java.sql.*;
3
4 public class Flowatch {
5
6     public static void main(String[] args) {
7         String url = "jdbc:mysql://localhost:3306/flowatch";
8         String user = "root";
9         String password = "AbbeilenoirK0022*"; // remplace par ton mot de passe MySQL
10
11         try {
12             // Connexion
13             Connection conn = DriverManager.getConnection(url, user, password);
14             System.out.println("Connexion réussie !");
15
16             // Requête pour afficher conducteurs et leurs véhicules
17             String sql = """
18             SELECT c.id AS conducteur_id, u.nom, u.email, v.marque, v.modele, v.localisation
19             FROM conducteurs c
20             JOIN utilisateurs u ON c.utilisateur_id = u.id
21             JOIN vehicules v ON c.vehicule_id = v.id
22             """;
23
24             Statement stmt = conn.createStatement();
25             ResultSet rs = stmt.executeQuery(sql);
26
27             System.out.println("Conducteurs et leurs véhicules :");
28             while (rs.next()) {
29                 int conducteurId = rs.getInt("conducteur_id");
30                 String nom = rs.getString("nom");
31                 String email = rs.getString("email");
32                 String marque = rs.getString("marque");
33                 String modele = rs.getString("modele");
34             }
35         } catch (SQLException e) {
36             e.printStackTrace();
37         }
38     }
39 }
```

4 Pour la récupération de Données avec JDBC

Étapes clés pour récupérer des données avec JDBC :

1. Il faut établir une connexion à la base de données : j'utilise `DriverManager.getConnection(url, user, password);`.
2. Créer une requête SQL : Utiliser `SELECT` pour récupérer des données de la base.
3. Créer un Statement ou PreparedStatement : Préparer l'exécution de la requête.
4. Exécuter la requête : Utiliser `executeQuery()` pour obtenir un `ResultSet`.
5. Traiter les résultats : Lire les données du `ResultSet`.
6. Fermer la connexion : Il faut toujours fermer la connexion pour éviter les fuites de mémoire.

5 Résultat



```
<terminated> Flowatch [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (31 mai 2025, 14:57:28 – 14:57:31 elapsed: 0:00:03.439) [pid: 11900]
Connexion réussie !
Conducteurs et leurs véhicules :
Conducteur #1 - Oko Dani (okodani@gmail.com)
Véhicule : Toyota Corolla - Localisation : Dolisie
-----
Conducteur #2 - Patrick Mbiandou (mbiandoupatrick@gmail.com)
Véhicule : Bus Model 3 - Localisation : Point Noire
-----
Connexion fermée.
```

Conclusion

Ce tutoriel montre comment intégrer Java avec MySQL à travers l'exemple flowatch .