

JDBC



- » Présentations
- » Introduction à JDBC
- » Drivers
- » Bonnes pratiques
- » Utilisation

Simon Mielcarek

Développeur chez
Synnaxium Studio
Formateur



Synnaxium

Et vous ?

4

Présentez vous !

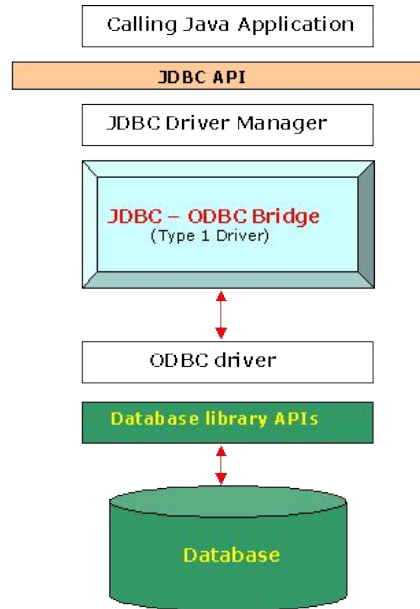


1. Introduction à JDBC

Package dans JDK : `java.sql`

Permet la connexion et les requêtes aux SGBD

Nécessite un pilote par SGBD



Plusieurs drivers disponibles :

- JDBC-ODBC

- JDBC-API

- Middleware

- Natif

Nous allons utiliser MySQL

Le driver nous est fourni directement sur le site MySQL

<https://dev.mysql.com/downloads/connector/j/>

Réalisons quelques tests

Nous utilisons gradle comme “build tool”

Ouvrir build.gradle :

Mettre à jour le groupe “dependencies”

```
implementation "mysql:mysql-connector-java:8.0.29"
```

Sur VSCode il faut “Refresh” notre projet
Clic droit sur build.gradle “Reload Project”
Redémarrer VSCode

Pour tester la connexion nous allons utiliser le code suivant :

```
Connection conn = null;
try {
    conn =
        DriverManager.getConnection("jdbc:mysql://51.68.227.19/sakila?" +
            "user=M2l&password=H3l!0M2l");

    // Do something with the Connection
} catch (SQLException ex) {
    // handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
}
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Main {

    public static void main(String[] args){

        Connection conn = null;
        try {
            conn =
                DriverManager.getConnection("jdbc:mysql://51.68.227.19/sakila?" +
                                           "user=M2I&password=H3ll0M2I");

            // Do something with the Connection
        } catch (SQLException ex) {
            // handle any errors
            System.out.println("SQLException: " + ex.getMessage());
            System.out.println("SQLState: " + ex.getSQLState());
            System.out.println("VendorError: " + ex.getErrorCode());
        }
    }
}
```

D'abord, créer un objet de type “Connection”

Et établir à connexion à la base de données.

Pour cela nous utilisons la commande :

`DriverManager.getConnection()`

<https://docs.oracle.com/en/java/javase/18/docs/api/java.sql/java/sql/DriverManager.html>

Soit utiliser uniquement l'url

`jdbc:mysql://51.68.227.19/sakila?user=M2I&password=H3ll0M2I`

Soit utiliser url, identifiant, mot de passe

`jdbc:mysql://51.68.227.19/sakila` et nos identifiants séparément

Ne pas oublier de gérer les Exceptions de type `SQLException`

D'abord, préparons la requête :

```
Statement stmt = connection.createStatement()
```

Ensuite nous exécutons notre requête :

```
ResultSet resultat = stmt.executeQuery(query)
```



```
while(resultat.next()){  
    ...  
}
```

Va déplacer le pointeur du résultat vers le résultat suivant jusqu'à ce qu'il renvoie false

```
String titre = monResultat.getString("title");  
int annee = monResultat.getInt("release_year");  
System.out.println(titre + "\t" + annee);
```

```
public static void main(String[] args) {
    Connection conn = null;
    try {
        conn =
            DriverManager.getConnection("jdbc:mysql://51.68.227.19/sakila?" +
                                       "user=M2I&password=H31l0M2I");

        Statement stmt = conn.createStatement();
        ResultSet resultat = stmt.executeQuery(sql: "SELECT * FROM film");

        while(resultat.next()){
            String titre = resultat.getString(columnLabel: "title");
            int annee = resultat.getInt(columnLabel: "release_year");
            System.out.println(titre+" - "+annee);
        }

        // Do something with the Connection
    } catch (SQLException ex) {
        // handle any errors
        System.out.println("SQLException: " + ex.getMessage());
        System.out.println("SQLState: " + ex.getSQLState());
        System.out.println("VendorError: " + ex.getErrorCode());
    }
}
```

Afin d'éviter la multiplication des lignes de codes

- Créer une classe statique permettant la configuration

- Créer des méthodes permettant de retourner des objets afin de manipuler les connexions à la base de donnée

D'abord, préparons la requête :

```
Statement stmt = connection.createStatement()
```

Ensuite nous exécutons notre requête :

```
ResultSet resultat = stmt.executeUpdate(query)
```

