



Chapitre 4 : Accès aux bases de données avec JDBC



Introduction

- JDBC (Java DataBase Connector) est une API chargée de communiquer avec les bases de données en Java.
- Les classes et interfaces de l'API JDBC figurent dans le package `java.sql` :

```
import java.sql.*;
```
- JDBC peut être utilisé pour accéder à n'importe quelle base de données à partir de:
 - Simple application Java
 - Une servlet
 - Page JSP, ...



Travail avec une base de données

- JDBC permet de travailler avec les base de données de la même façon quelque soit leur fournisseur (Oracle, SQL Server, MySQL, PostgreSQL,...).
- Il suffit de télécharger la bibliothèque qui assure la communication entre Java et cette base de données.
- Cette bibliothèque s'appelle Driver ou Pilote ou Connecteur.
- Elle figure sur le site du fournisseur du SGBDR utilisé.
- Exemple de pilote MySQL :
 - Le connecteur MySQL pour Java se nomme comme cet exemple : "**mysql-connector-java-8.0.27.jar**"

03/12/2024

cours JEE - Dr. Abdessamad Belangour

184



Etapas d'interaction avec une BDD

1. Chargement du pilote
2. Etablissement de la connexion
3. Création des objets encapsulant les requêtes
4. Exécution des requêtes
5. Parcours des résultats dans le cas d'une requête de sélection
6. Fermeture des objets résultats, requêtes et connexion

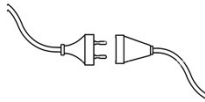
Java - Dr A. Belangour

185

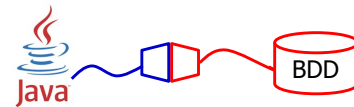


Chargement du pilote

- Pour se connecter à une base de données il faut charger son pilote.
- Le chargement se fait comme suit : `Class.forName("nom_classe_accès_bdd");`
- La documentation de la Bdd utilisée fournit le nom de la classe à utiliser.
- Exemple :
 - Dans le cas de la Bdd MySQL, ce chargement est comme suit :
`Class.forName(com.mysql.cj.jdbc.Driver)` ou `Class.forName(com.mysql.jdbc.Driver)` selon les versions
- Une fois chargée, la classe JDBC qui se nomme **DriverManager** prend en charge le driver pour communiquer avec la base de donnée.



Java - Dr A. Belangour



186



Classes de l'API JDBC

- Les classes et interfaces les plus usuelles sont les suivantes:
 - **DriverManager** (classe): charge et configure le driver de la base de données.
 - **Connection** (interface): réalise la connexion et l'authentification à la base de données.
 - **Statement** (interface): contient la requête SQL et la transmet à la base de données.
 - **PreparedStatement** (interface): représente une requête paramétrée
 - **ResultSet** (interface): représente les résultat d'une requête de sélection.

03/12/2024

cours JEE - Dr. Abdessamad Belangour

187



Etablissement de la connexion

- Pour se connecter à une base de données, il faut disposer d'un objet **Connection** créé grâce au DriverManager en lui passant :
 - l'URL de la base à accéder , Le login, Le mot de passe
- Exemples:
 - `String url="jdbc:mysql://localhost/mydb"; // exemple URL BDD MySQL`
 - `String login="root";`
 - `String password="motdepasse";`
 - `Connection con=DriverManager.getConnection(url, login, password);`

03/12/2024

cours JEE - Dr. Abdessamad Belangour

188



Exécution de requêtes SQL

- L'interface **Statement** permet d'envoyer des requêtes SQL à la base de données.
- Un objet Statement est créé grâce à un objet Connection de la façon suivante :
`Statement st = con.createStatement();`
- Il possède deux méthodes :
 - **executeUpdate()** : Insertion, suppression, mise à jour.
 - `int n= st.executeUpdate("INSERT INTO Etudiant VALUES (3452,'Taha','Ali')");`
 - **executeQuery()** : Selection.
 - `ResultSet res= st.executeQuery("SELECT * FROM Etudiant");`

03/12/2024

cours JEE - Dr. Abdessamad Belangour

189



Requêtes avec paramètres

- L'interface **PreparedStatement** permet d'envoyer des requêtes SQL à la base de données en prenant des paramètres.
- Ces paramètres sont représentés par des points d'interrogation(?) et doivent être spécifiés avant l'exécution.
- Exemple :
 - **PreparedStatement** p= con.**prepareStatement**("select* from Etudiant where cne=?
And nom= ? ");
p.**setInt**(1, 3452345); // remplissage 1^{er} paramètre
p.**setString**(2, "Alaoui"); // remplissage 2^{ième} paramètre
ResultSet resultats = p.**executeQuery**(); // exécution de la requête

Java - Dr A. Belangour

190



Résultat d'une requête de sélection

- Une requête de sélection retourne un **ResultSet** qui est un ensemble d'enregistrements constitués de colonnes qui contiennent les données.
- Les principales méthodes :
 - **next()** : se déplace sur le prochain enregistrement : retourne false si la fin est atteinte. Le curseur pointe initialement juste avant le premier enregistrement.
 - **getInt(int/String)** : retourne le contenu de la colonne dont le numéro (resp. le nom) est passé en paramètre sous forme d'entier.
 - **getFloat(int/String)** : retourne le contenu de la colonne sous forme de nombre flottant.
 - **getDate(int/String)** : retourne le contenu de la colonne sous forme de date.
 - **Close()** : ferme le ResultSet

03/12/2024

cours JEE - Dr. Abdessamad Belangour

191



Résultat d'une requête de sélection

□ Exemple :

```
ResultSet res= st.executeQuery("SELECT * FROM Etudiant");
while (res.next()) {
    System.out.println("CNE= "+res.getString(1)+" Nom= " + res.getString(2)
                      +" Prénom= "+res.getString(3));
}
res.close();
```



Enoncé TP

- Création de la base de données :
 - Utiliser MYSQL pour créer une base de données nommée « etudiantsDB » ayant une table ETUDIANT composée des champs :
 - CNE (Auto_increment)
 - NOM
 - PRENOM
 - Remarque : Ne pas oublier de télécharger le connecteur MYSQL pour java et de le référencer dans votre projet.



Enoncé TP

- Création de l'application Web :
 - L'application web démarre avec une page d'accueil « index.jsp » qui se connecte à la base de données et affiche sous forme d'un tableau HTML le « cne », « nom » et « prénom » d'un étudiant ainsi que des liens d'ajout, de suppression de modification et de recherche d'un étudiant

Liste des Etudiants Inscrits :

CNE	NOM	PRENOM	operations	
14	Samiri	Hassan	supprimer	modifier
15	Tantawi	Taha	supprimer	modifier
16	Youssefi	Youssef	supprimer	modifier
17	Tahiri	Adil	supprimer	modifier

[Ajouter Etudiant](#)

Rechercher Etudiant :



Enoncé TP

- La page de la recherche affiche les résultats sous forme d'un tableau HTML et affiche un lien de retour à la page d'accueil.

Liste des Etudiants Inscrits :

CNE	NOM	PRENOM	operations	
1	Samiri	Hassan	supprimer	modifier
2	Tantawi	Taha	supprimer	modifier
3	Youssefi	Youssef	supprimer	modifier
4	Tahiri	Adil	supprimer	modifier

[Ajouter Etudiant](#)

Rechercher Etudiant :



Resultats de la recherche:

CNE	NOM	PRENOM	operations	
2	Tantawi	Taha	supprimer	modifier
4	Tahiri	Adil	supprimer	modifier

[retour à la page d'accueil](#)

- L'insertion, la modification et la suppression est confiée à des Servlets
- Le reste est confié à des pages JSP



Enoncé TP

- Le projet est réalisé avec **Netbeans** et dispose de la structure ci-dessous.

