



# JAVA

## JDBC

### Charger un pilote en mémoire

# JDBC = Java DataBase Connectivity

- JDBC est une API d'accès aux systèmes de gestion de base de données relationnelles qui permet d'exécuter des requêtes SQL au sein d'un programme Java et de récupérer les résultats, ce qui représente une alternative aux solutions propriétaires. C'est de plus une tentative de standardiser l'accès aux bases de données car l'API est indépendante du SGBD choisi, pourvu que le pilote JDBC existe pour ce SGBD, et qu'il implémente les classes et interfaces de l'API JDBC.

# Structure générale

**Pour effectuer un traitement avec une base de données, il faut :**

- 1. charger un pilote en mémoire,**
- 2. établir une connexion avec la base de données,**
- 3. récupérer les informations relatives à la connexion,**
- 4. exécuter des requêtes SQL et/ou des procédures stockées,**
- 5. récupérer les informations renvoyées par la base de données (si nécessaire),**
- 6. fermer la connexion**

# Bibliothèques nécessaires

**Pour instancier les objets nécessaires au dialogue avec une base de données, il faut importer les bibliothèques suivantes :**

- `java.sql.*`;
- `sun.jdbc.odbc.*`; (pour inclure le pont JDBC-ODBC)

# Liste de quelques pilotes

- **Pour une base Oracle : `oracle.jdbc.driver.OracleDriver`**
- **Pour une base Access : `sun.jdbc.odbc.JdbcOdbcDriver`**
- **Pour une base PostgreSQL : `postgresql.Driver`**
- **Pour une base MySQL : `org.gjt.mm.mysql.Driver`**

# Charger un pilote en mémoire

Le pilote JDBC connaît les méthodes pour se connecter à votre base de données, c'est pourquoi celui-ci est essentiel.

Ce pilote est généralement disponible dans un package jar.

Le chemin doit être ajouté à votre variable d'environnement CLASSPATH pour permettre au programme de l'utiliser.

# Charger un pilote en mémoire

La première étape est de charger le pilote en utilisant la méthode *Class.forName(String driver)*.

Cette classe permet ainsi au programme de rester totalement indépendant de la base de données utilisée en conservant le nom du pilote dans un fichier de propriétés.

Voici un exemple avec le pilote de Sun utilisé pour se connecter à une base de données via ODBC (sun.jdbc.odbc.JdbcOdbcDriver) :

```
try {  
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
}  
  
catch (ClassNotFoundException e) {  
    system.err.println("Erreur de chargement de pilote");  
}
```

# Tester

