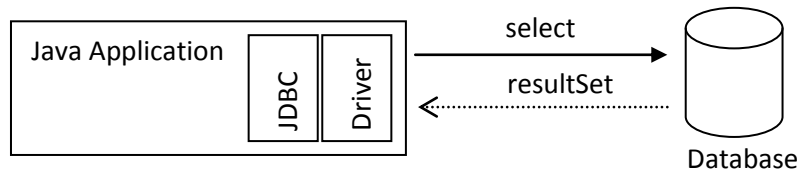


Tutoriel JDBC premiers pas

Lecture de Données dans une Base de Données



JDBC définit un ensemble d'interfaces pour connecter une application à une base de données SQL. JDBC est inclus dans Java SE dans les packages `java.sql` et `javax.sql` :

- JDBC 3.0 dans Java SE 5 et Java SE 6
- JDBC 4.1 dans Java SE 7

L'utilisation de JDBC se fait en 2 étapes :

1. Etablissement de la connexion à la base de données : à l'image d'une ouverture de fichier, il faut ouvrir une connexion cliente vers le serveur de base de données. Pour réaliser cette phase un peu technique, on vous fournit la classe `fr.eisti.sql.DatabaseConnector` à utiliser comme suit :

```
// à faire une fois au démarrage de l'application pour préciser la base //
// utilisée, le logon et le mot de passe
DatabaseConnector.setOracleXeLocalParameters("monlogin", "monpassword");
// à faire autant de fois que nécessaire (si une connexion est déjà
// ouverte, c'est celle-ci qui est retournée)
java.sql.Connection connection = DatabaseConnector.getConnection();
// à faire à la fin de l'application
DatabaseConnector.closeConnection();
```

2. Utilisation de la connexion pour faire une requête (select) :

```
Statement stmt = null;
String query = "select NOM, PRENOM, AGE from PERSONNE";
try {
    stmt = connection.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    while (rs.next()) {
        String nom = rs.getString("NOM");
        // ou par numéro de champs : nom = rs.getString(1);
        String prenom = rs.getString("PRENOM");
        // ou par numéro de champs : prenom = rs.getString(2);
        int age = rs.getInt("AGE");
        // ou par numéro de champs : age = rs.getInt(3);
        Personne p = new Personne(nom, prenom, age);
        // utiliser p ...
        System.out.println("une personne extraite de la BDD " + p);
    }
} catch (SQLException e) {
    // traiter l'exception
} finally {
    if (stmt != null) { stmt.close(); }
}
```

En Java SE 7 la structure du code devient :

```
String query = "select NOM, PRENOM, AGE from PERSONNE";
try (Statement stmt = connection.createStatement()) {
    ResultSet rs = stmt.executeQuery(query);
    // traitement du résultat ...
} catch (SQLException e) {
    // traiter l'exception
}
// stmt est automatiquement fermé si besoin (exception ou pas)
```

Chaque éditeur de SGBD fournit un driver sous la forme d'un jar contenant les implémentations des interfaces de JDBC capables d'interagir avec son produit. On a donc un driver différent pour Oracle, MySQL, SQL Server, etc. Pour une base Oracle (XE) 10g le driver est ojdbc14.jar, pour une base Oracle (XE) 11 le driver est ojdbc6.jar. Ces drivers sont disponibles sur votre PC dans le répertoire d'installation d'Oracle.

Bibliographie :

1. Un tutoriel Oracle sur JDBC : <http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>
2. Un autre sur la structure try-with-resources de Java SE 7 :
<http://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html>
3. Drivers JDBC et URL de connexion pour différents SGBD : <http://wiki.netbeans.org/DatabasesAndDrivers>