

Доступ до баз даних з використанням JDBC

Типи JDBC-драйверів

За способом реалізації драйверів розділяються на 4 типи:

1. JDBC-ODBC міст. До цього типу належать драйвери, реалізовані поверх драйвера ODBC. Фактично всі виклики API JDBC транслюються у виклики ODBC, а також подальша обробка виконується API ODBC.
2. Native-JDBC. До другого типу належать драйвери, що використовують програмні частини, написані на інших мовах.
3. Network-JDBC. Цей тип драйвера повністю реалізується на Java, але при цьому виклики JDBC передаються в сетевий протокол (RMI, HTTP та ін.), що далі транслюється в конкретний протокол баз даних.
4. "Чистий" драйвер JDBC. Схожий до 3-го типу, реалізується повністю на Java, але виклики реалізуються з використанням протоколу баз даних, без мережевих протоколів.

Підключення драйвера MySQL

- Можна завантажити jar-файл драйвера з офіційного сайту вендора бази даних, або з центрального репозиторію Maven за адресою mvnrepository.com
- Для підключення бібліотеки jar до класичного проєкту Java в IntelliJ Idea потрібно виконати команду File=>Project Structure...=> Libraries

Класи JDBC

Інтерфейс\Клас	Опис
DriverManager	Клас для керування драйверами в Java додатку. За його допомогою можна створити підключення до БД, використовуючи потрібний драйвер.
Connection	Інтерфейс описує методи для з'єднання з БД
Statement	Інтерфейс містить набір методів для обробки запитів до БД
DatabaseMetaData	Клас описує інформацію про базу даних
ResultSetMetaData	Клас описує інформацію про результуючий набір

Класи JDBC

Інтерфейс\Клас	Опис
Statement	Інтерфейс містить набір методів для обробки запитів до БД
PreparedStatement	Інтерфейс містить набір методів для обробки параметричних запитів до БД
CallableStatement	Інтерфейс містить набір методів для обробки роботи зі Сторед-процедурами
ResultSet	Клас представляє набір даних (результат роботи оператора SELECT)

Приклад підключення

```
try {  
    Class.forName("com.microsoft.sqlserver.jdbc  
                    .SQLServerDriver");  
} catch (ClassNotFoundException e) {  
    System.out.print("Помилка завантаження драйвера");  
    return;  
}  
String connectionUrl = "jdbc:sqlserver://localhost:1433;" +  
    "databaseName=MyBase;user=sa;password=admin;";  
try {  
    Connection conn = DriverManager  
        .getConnection(connectionUrl);
```

Приклад підключення

```
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery("select * from Users");

while(rs.next()){
    System.out.print(rs.getString("Name")+"\t");
    System.out.print(rs.getString("Age")+"\t");
    System.out.println();
}
rs.close();
conn.close();
} catch (SQLException e) {
    System.out.print("Ошибка работы с данными");
    return;
}
```

Результуючий набір ResultSet

Метод	Опис
<code>boolean first()</code>	Перехід на перший запис, true якщо перехід успішний.
<code>boolean last()</code>	Переходить на останній запис, true якщо перехід успішний.
<code>boolean next()</code>	Перехід на наступний запис, true якщо перехід успішний.
<code>boolean previous()</code>	Перехід на попередній запис, true якщо перехід успішний.
<code>boolean absolute(int numRecord)</code>	Переходить на вказаний запис в результуючому наборі

Результуючий набір ResultSet

Метод	Опис
<code><Тип даних> get<Тип даних>(int index)</code> <code><Тип даних> get<Тип даних>(String name)</code>	Повертає значення поля поточного рядка, приведені до певного типу даних (<code>getString</code> , <code>getInt</code>). Має дві перегрузки, для звертання до поля за індексом (індексування починається з одиниці) і за іменем поля.
<code>void close()</code>	Закриває набір даних (звільняє ресурси зайняті для обробки набору даних). Після закриття, набір даних недоступний.

Оновлення даних в ResultSet

ResultSet createStatement(int type, int concur);

type:

TYPE_FORWARD_ONLY - вказує що курсор може переміщатися тільки вперед по результуючому набору. Виклик методів last (), first, previous () - призведе до порушення виняткової ситуації.

TYPE_SCROLL_INSENSITIVE -вказує що курсор може переміщатися довільно по результуючому набору. Зміни внесені в результуючий набір віобразяться тільки при повторному відкритті набору даних.

Оновлення даних в ResultSet

TYPE_SCROLL_SENSITIVE - вказує що курсор може переміщатися довільно по результуючому набору. Зміни внесені в результуючий набір негайно відображаються.

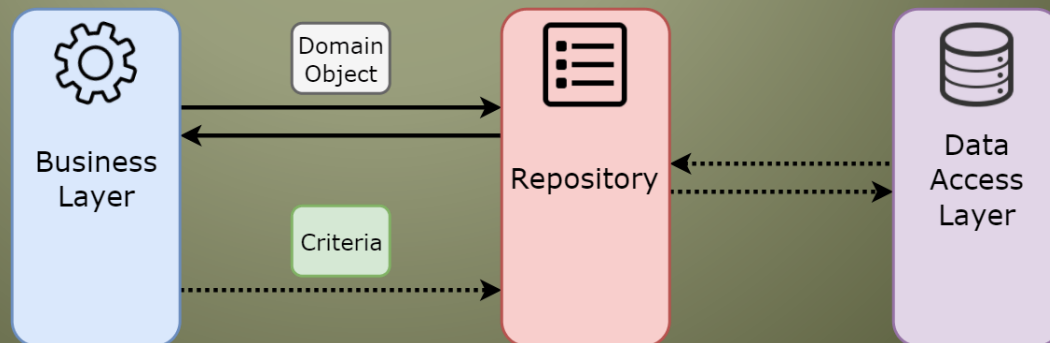
concur - вказує тип оновлення результуючого набору.

CONCUR_READ_ONLY - вказує, що набір даних не можна оновити

CONCUR_UPDATABLE - вказує, що набір даних можна оновити

Repository

Репозиторій – це фасад для доступу до бази даних. Репозиторій є посередником між шаром доступу до даних і доменним шаром, працюючи як in-memory колекція доменних об'єктів. Клієнти створюють декларативні опису запитів і передають їх в репозиторій для виконання.



Repository

