Un petit exemple JDBC

1 Préparation

- Créez dans Eclipse un projet Java standard.
- Ajoutez Maven à votre projet : **Sélectionnez votre projet** / **Bouton-droit** / **Configure** / **Convert to Maven Project**. Vous devez à cette étape donner une numéro de version à votre projet. Laissez les valeurs par défaut.
- Ajoutez au fichier `pom.xml` (créé à l'étape précédente) la dépendance ci-dessous (base de données embarquée HyperSQL) :

```
<!-- https://mvnrepository.com/artifact/org.hsqldb/hsqldb -->
<dependencies>
<dependency>
    <groupId>org.hsqldb</groupId>
    <artifactId>hsqldb</artifactId>
    <version>2.4.1</version>
    <scope>test</scope>
</dependency>
</dependencies>
```

2 Un petit exemple JDBC

Préparez la classe ci-dessous. C'est une classe de service très simple qui sauvegarde, supprime et récupère des noms associés à des numéros.

```java
package simplejdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Collection;
import java.util.LinkedList;

public class SimpleJdbc {

    final String url = "jdbc:hsqldb:file:testdb";
    final String user = "SA";
    final String password = "";

    private Connection newConnection() throws SQLException {
        return DriverManager.getConnection(url, user, password);
    }

    public void initSchema() throws SQLException {
        var query = "create table if not exists NAME (" //
                + " id integer not null," //
                + " name varchar(50) not null," //
                + " primary key (id) )";
        try (var conn = newConnection()) {
            conn.createStatement().execute(query);
        }
    }

    public void addName(int id, String name) throws SQLException {
        var query = "insert into NAME values (?,?)";
        try (var conn = newConnection()) {
            var st = conn.prepareStatement(query);
            st.setInt(1, id);
            st.setString(2, name);
            st.execute();
        }
    }

    public void deleteName(int id) throws SQLException {
        var query = "Delete From NAME where (id = ?)";
        try (var conn = newConnection()) {
            var st = conn.prepareStatement(query);
            st.setInt(1, id);
            st.execute();
        }
    }

    public String findName(int id) throws SQLException {
        var query = "Select * From NAME where (id = ?)";
        try (var conn = newConnection()) {
            var st = conn.prepareStatement(query);
            st.setInt(1, id);
            var rs = st.executeQuery();
            if (rs.next()) {
                return rs.getString("name");
            }
        }
        return null;
    }

    public Collection<String> findNames() throws SQLException {
        var query = "Select * From NAME order by name";
        var result = new LinkedList<String>();
        try (var conn = newConnection()) {
            var st = conn.createStatement();
            var rs = st.executeQuery(query);
            while (rs.next()) {
```

3 Tester votre classe

Créez maintenant la classe de test unitaire (avec **JUnit 5**) et vérifiez son fonctionnement :

```java
package simplejdbc;

import static org.junit.Assert.assertEquals;
import static org.junit.jupiter.api.Assertions.assertThrows;

import java.sql.SQLException;

import org.junit.jupiter.api.Test;

public class TestSimpleJdbc {

    @Test
    public void testSimleJdbcSample() throws SQLException {
        var dao = new SimpleJdbc();
        dao.initSchema();
        dao.deleteName(100);
        dao.deleteName(200);
        dao.addName(100, "Hello");
        dao.addName(200, "Salut");
        assertEquals("Hello", dao.findName(100));
        assertEquals("Salut", dao.findName(200));
        for (String name : dao.findNames()) {
            System.out.printf("name=%s\n", name);
        }
    }

    @Test
    public void testErrors() {
        assertThrows(SQLException.class, () -> {
            var dao = new SimpleJdbc();
            dao.initSchema();
            dao.addName(100, "Hello");
            dao.addName(100, "Salut");
        });
    }

}
```