

Explications liées au programme DemoTest10JdbcDao

<pre><dependency> <groupId>org.apache.derby</groupId> <artifactId>derbyclient</artifactId> </dependency></pre>	Fichier Pom → il faut ajouter les dépendances à Derby
--	---

<pre>spring.datasource.url=jdbc:derby://localhost:1527/BDEmp2020 spring.datasource.username=app spring.datasource.password=app spring.datasource.driver-class-name= org.apache.derby.jdbc.ClientDriver</pre>	Fichier application.properties : Permet de configurer la BD utilisée
--	--

<pre>@Data @NoArgsConstructor @AllArgsConstructor public class Employe implements Serializable { private Integer nemp; private double bareme; private String loc; private String nom; private String prenom; private Character sexe;}</pre>	La classe Employe utilisée n'est pas une Entity JPA
---	---

<pre>public class Dao { public static List<Employe> findAll(Connection con) { List<Employe> list=new ArrayList<>(); try { Statement stmt=con.createStatement(); ResultSet rst=stmt.executeQuery("Select * from Employe"); while(rst.next()) {Employe E=new Employe(rst.getInt("Nemp"), rst.getString("Nom"), rst.getString("Prenom"), rst.getString("Sexe").charAt(0), rst.getDouble("Bareme"), rst.getString("Loc")); list.add(E); } } catch (SQLException ex) { Logger.getLogger(Dao.class.getName()).log(Level.SEVERE, null, ex); } return list; } }</pre>	Exemple d'utilisation d'une classe DAO pour un méthode seulement
--	--

Différents accès à la BD via JDBC :

@Component

public class AccesDB implements CommandLineRunner {

@Autowired JdbcTemplate jdbc; // Utilisation d'un objet JdbcTemplate fournit par Spring

@Override

public void run(String... strings) throws Exception {

// Utilisation d'un objet JdbcTemplate fournit par Spring et de la BD

// déclarée dans application.properties

SqlRowSet rs=jdbc.queryForRowSet("select * from Employe");

while(rs.next()) {System.out.println(rs.getString("Nom")+rs.getString("Prenom"));}

DataSource ds=jdbc.getDataSource();

Connection con2=null;

con2=ds.getConnection();

for(Employe e:Dao.findAll(con2))

System.out.println(e.toString());

// Utilisation plus classique de JDBC !

Connection con=null;

Statement stmt=null;

try {Class.forName("org.apache.derby.jdbc.ClientDriver"); } catch (ClassNotFoundException ex) {}

try { con =

DriverManager.getConnection("jdbc:derby://localhost:1527/BDEmp2020;

user=app;password=app");

} catch (SQLException ex) { }

stmt=con.createStatement();

ResultSet rst=stmt.executeQuery("Select * from Employe");

while(rst.next())

{System.out.println(rst.getString("Nom")+rst.getString("Prenom"));}

}}