

# SERVICES WEB – SOAP

## CONNEXION DB

AUDITOIRE : 3 EME ANNEE LI

---

### 1. Connexion à la base de données avec Java

L'interface permettant d'accéder aux bases de données relationnelles à partir de Java est *Java Database Connectivity (JDBC)*. Via JDBC, vous créez une connexion à la base de données, émettez des requêtes sur la base de données, mettez à jour et recevez les résultats.

JDBC fournit une interface qui vous permet d'effectuer des opérations SQL indépendamment de l'instance de la base de données utilisée. Pour utiliser JDBC, vous avez besoin de l'implémentation spécifique du pilote JDBC pour la base de données.

Pour connecter une application Java à la base de données MySQL, nous devons suivre les 5 étapes suivantes.

### 2. Pilote MySQL JDBC

Pour vous connecter à MySQL à partir de Java, vous devez utiliser le pilote JDBC à partir de MySQL. Le pilote JDBC MySQL est appelé *MySQL Connector / J*. Vous trouverez le dernier pilote JDBC MySQL à l'URL suivante: <http://dev.mysql.com/downloads/connector/j>.

### 3. Exercice

Dans ce TP, nous allons utiliser le projet réalisé dans le **TP 1** et nous allons créer une base de données puis connecter cette base avec notre projet java Web.

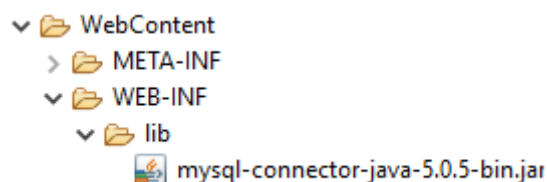
Nous utilisons **MySQL** comme base de données. Nous avons donc besoin de connaître les informations suivantes sur la base de données MySQL:

- a) **Classe de pilote** : La classe de pilote pour la base de données MySQL est **com.mysql.jdbc.Driver**
- b) **URL de connexion** : L'URL de connexion pour la base de données MySQL est jdbc : "[jdbc:mysql://localhost/tptest](#)" où **jdbc** (Java Database Connectivity) est l'API (Application Programming Interface), **mysql** (SGBD), **localhost** est le nom du serveur sur lequel mysql est en cours d'exécution, nous pouvons également utiliser l'adresse IP et **tptest** est le nom de la base de données. Nous

pouvons utiliser n'importe quelle base de données. Vous pouvez remplacer le **tptest** par le nom de votre base de données.

- c) **Nom d'utilisateur:** Le nom d'utilisateur par défaut pour la base de données MySQL est **root**.
- d) **Mot de passe:** C'est le mot de passe donné par l'utilisateur lors de l'installation de la base de données MySQL (par défaut **"** ).

1. Commençons par créer une base de données MySQL nommée **tptest** puis une table dans cette base nommée **Person** contenant les mêmes attributs de la classe **Person**.
2. Pour connecter une application java à la base de données mysql, le fichier **mysqlconnector.jar** doit être chargé.  
Collez le fichier **mysqlconnector.jar** sous le dossier **lib** :



3. Ensuite, créez un nouveau package nommé **com.poly.info.db**. Dans ce package, créez une classe nommée **ConnexionDB** pour se connecter à la

```
1 package com.poly.info.db;
2
3 import java.sql.Connection;
4
5
6
7 public class ConnexionDB {
8
9     String url="jdbc:mysql://localhost/tptest";
10    String login="root";
11    String password="";
12
13    static Connection cn; // l'objet connection
14
15    private ConnexionDB() {
16        super();
17        try {
18            Class.forName("com.mysql.jdbc.Driver");
19
20            cn = DriverManager.getConnection(url, login, password);
21        }
22        catch (Exception e) {
23
24            System.out.println(e.getMessage());
25
26        }
27    }
28
29
30    public static Connection getConnexion () {
31        if (cn==null) {
32            new ConnexionDB();
33        }
34
35        return cn;
36    }
37 }
```

4. Modifiez la classe **PersonServiceImpl**, en créant un objet **cn** de la classe **ConnexionDB** pour vous connecter à la base de données MySQL et effectuer les requêtes d'insertion et de suppression (les mêmes classes du TP1).

- Voici la Requête de la classe **addPerson** :

```
public class PersonServiceImpl implements PersonService {

    Connection cn = ConnexionDB.getConnexion();
    Statement st = null;

    @Override
    public boolean addPerson(Person p) {
        String sql = "INSERT INTO `person`(`Name`, `Age`) VALUES ('" + p.getName() + "','" + p.getAge() + ")";
        try {
            st = (Statement) cn.createStatement();
            st.executeUpdate(sql);
            System.out.println("Ajout avec succès");
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Erreur add");

            return false;
        }
    }
}
```

- Voici la Requête de la classe **deletePerson** :

```
@Override
public boolean deletePerson(int id) {

    String sql = "DELETE FROM `person` WHERE id=" + id;

    try {
        st = cn.createStatement();
        st.executeUpdate(sql);

        return true;
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();

        System.out.println(e.getMessage());
        return false;
    }
}
```

- Voici la Requête de la classe **getPerson** :

```

@Override
public Person getPerson(int id) {
    // return persons.get(id);
    Person person = null;

    String sql = "SELECT `ID`, `Name`, `Age` FROM `person` WHERE id=" + id;

    try {
        st = cn.createStatement();
        ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            person = new Person();
            person.setId(rs.getInt("id"));
            person.setAge(rs.getInt("age"));
            person.setName(rs.getString("name"));
        }

        return person;
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();

        System.out.println(e.getMessage());
        return null;
    }
}

```

- Voici la Requête de la classe **getPersonByName** :

```

@Override
public Person getPersonByName(String name) {
    // return persons.get(id);
    Person person = null;

    String sql = "SELECT `ID`, `Name`, `Age` FROM `person` WHERE name='" + name + "'";

    try {
        st = cn.createStatement();
        ResultSet rs = st.executeQuery(sql);

        while (rs.next()) {
            person = new Person();
            person.setId(rs.getInt("id"));
            person.setAge(rs.getInt("age"));
            person.setName(rs.getString("name"));
        }

        return person;
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();

        System.out.println(e.getMessage());
        return null;
    }
}

```

- Voici la Requête de la classe **getAllPerson** :

```
public List<Person> getAllPersons() {  
    java.util.List<Person> persons = new ArrayList<>();  
    String sql = "SELECT * FROM `person`";  
    try {  
        st = (Statement) cn.createStatement();  
        ResultSet rs = (ResultSet) st.executeQuery(sql);  
        while (rs.next()) {  
            Person person = new Person();  
            person.setId(rs.getInt("id"));  
            person.setName(rs.getString("Name"));  
            person.setAge(rs.getInt("Age"));  
            persons.add(person);  
        }  
        return persons;  
    } catch (Exception e) {  
        e.printStackTrace();  
        System.out.println(e.getMessage());  
        return null;  
    }  
}
```