

Подключение к проекту базы данных (jdbc) через Maven

1. Создаём проект с использованием maven и прописываем как зависимость драйвер jdbc:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apach
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>untitled5</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
    <dependency>
      <groupId>org.postgresql</groupId>
      <artifactId>postgresql</artifactId>
      <version>42.3.3</version>
    </dependency>
  </dependencies>
</project>
```

А именно:

```
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.3.3</version>
</dependency>
```

Найти зависимость можно просто прописав в браузере «jdbc postgresql maven» или перейдя по этой ссылке:

<https://mvnrepository.com/artifact/org.postgresql/postgresql>

и выбрав последнюю версию.

Зависимости подключённого к проекту артефакта maven найдёт сам, сам составит список всех библиотек, необходимых проекту, включая зависимости зависимостей и зависимости зависимостей зависимостей, сам скачает их на машину, на которой происходит сборка и сам добавит их в classpath.

2. Создадим базу данных в системе управления базами данных:

```
[denis@denis-hpelitebookfolio9470m ~]$ sudo -iu postgres
[sudo] пароль для denis:
[postgres@denis-hpelitebookfolio9470m ~]$ createdb template
[postgres@denis-hpelitebookfolio9470m ~]$ psql -d template
psql (13.6)
Введите "help", чтобы получить справку.

template=#
```

3. Создадим таблицу employee:

```
template=# create table employee (
id_emp integer,
id_branch integer,
lastname varchar(30),
firstname varchar(30),
date_emp date,
salary integer,
commision numeric(4,2)
);
CREATE TABLE
```

4. Внесём в неё данные:

```
template=#
insert into employee
values (1, 1, 'Igor', 'Stezea', to_date('01/02/2012','dd/mm/yyyy'), 5600, 2.3);
INSERT 0 1
template=#
insert into employee
values (2, 1, 'Andrei', 'Cartea', to_date('12/04/2013','dd/mm/yyyy'), 5300, 1.9);
INSERT 0 1
```

5. Вставим следующий код в Main.java для проверки работоспособности:

```
import java.sql.*;

public class Main {
    public static void main(String[] args) {
        Connection connection = null;
        try {
            connection =
DriverManager.getConnection("jdbc:postgresql://192.168.43.184/template", "postgres",
"1961");
            System.out.println("\nConnected to the PostgreSQL server successfully.\n");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```

Statement statement = connection.createStatement();
String selectEmployee = "SELECT " +
    "id_emp, " +
    "id_branch, " +
    "rpad(lastname, 15), " +
    "rpad(firstname, 15), " +
    "date_emp, " +
    "salary, " +
    "commision " +
    "FROM employee;";
ResultSet resultSet = statement.executeQuery(selectEmployee);
while(resultSet.next()) {
    int id_emp = resultSet.getInt(1);
    int id_branch = resultSet.getInt(2);
    String firstname = resultSet.getString(3);
    String lastname = resultSet.getString(4);
    Date date = resultSet.getDate(5);
    int salary = resultSet.getInt(6);
    double commision = resultSet.getDouble(7);
    System.out.println(id_emp + "\t| " + id_branch + "\t| " + firstname + " | " + lastname + " | " + date + " | " + salary + " | " + commision);
}
connection.close();
} catch (SQLException e) {
    System.out.println(e.getMessage());
}
}
}

```

Результат:

```

Connected to the PostgreSQL server successfully.

1 | 1 | Igor          | Stezea          | 2012-02-01 | 5600 | 2.3
2 | 1 | Andrei        | Cartea          | 2013-04-12 | 5300 | 1.9

Process finished with exit code 0

```

Maven успешно подключил зависимость, данные прочтены без ошибок.

P.S.

Если вы не понимаете код, найдите мой прошедший документ «jdbc-java.pdf», там всё описано в подробностях.