



链滴

# JAVA 基础复习之 JDBC (配置动态数据源)

作者: [FILWLL](#)

原文链接: <https://ld246.com/article/1548059293919>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 复习原因

在项目的开发当中，之前数据库连接信息都是写死在配置文件当中。但是突然接到一个需求：获取外部数据源信息，然后将某些数据通过Echarts绘制成折线图展示出来（ps：数据源需要用户动设定）

有点懵，因为之前都是写死在配置文件当中的，然后在网上也找了一些资料，发现都不是自己想要。  
最后想到，既然是简单的获取数据进行展示，直接使用JDBC连接数据库获取就可以了啊。没有必要搞什么骚操作。

## JDBC

一直使用的JPA或者mybatis进行数据库连接，很少使用JDBC进行数据库连接。因此也复习一下JDBC，毕竟无论是JPA还是Mybatis都是对JDBC的一个封装。

## 什么是JDBC

JDBC(Java DataBase Connectivity,java数据库连接)是一种用于执行SQL语句的Java API，可以为多关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序（来源于百度百科）

## 如何使用JDBC

第一步 加载驱动

第二步 连接数据库

第三步 创建sql

第四步 通过Statement类执行sql

第五步 获取到执行结果集ResultSet然后进行业务操作

第六步 关闭数据库连接

完整代码如下：

```
String ClassName = "com.mysql.jdbc.Driver";
String url = "jdbc:mysql://127.0.0.1:3306/test";
String username = "root";
String password = "root";

//第一步：加载驱动
try { //加载MySQL的驱动类 将驱动注册到DriverManager当中
    Class.forName(ClassName);
} catch (ClassNotFoundException e) {
    System.out.println("加载驱动失败！ 请检查驱动名称");
    e.printStackTrace();
}

Connection con = null;
Statement statement = null;
```

```

String sql = null;
ResultSet resultSet = null;
try {
    //第二步：获取连接
    con = DriverManager.getConnection(url, username, password);
    //第三步：创建sql
    sql = "SELECT username,password FROM test1 ";
    //第四步：获取statement类
    statement = con.createStatement();
    //第五步：获取到执行后的结果集resultSet
    resultSet = statement.executeQuery(sql);
    while (resultSet.next()){
        //通过结果集的操作方法进行数据的获取 这里可以进行实际的业务操作，例如存到一个
        //应的实体类，返回给前端
        //这里是获取的
        System.out.println(resultSet.getString(1));
        System.out.println(resultSet.getString(2));
    }
} catch (SQLException e) {
    System.out.println("数据库连接失败！ 请检查数据库连接信息");
    e.printStackTrace();
}finally {
    //先关闭结果集
    try {
        resultSet.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    //然后关闭Statement对象
    try {
        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    //最后关闭连接
    try {
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

到这里一个简单的JDBC连接就搞定了。

## 动态配置数据源

从上面JDBC连接当中可以看到，需要关注的参数是：ClassName url username password sql 这5个参数。因此我们只需要让用户输入这5个参数的相关配置信息，然后我们后台封装一个工具类，就可简单的实现动态数据源的数据展示了。

### 第一步 创建一个配置类用于接受用户指定的数据源信息并存入数据库。

配置类代码如下：

```
@Id
@GeneratedValue(strategy= GenerationType.AUTO)
private int id;//主键.

private String driverClassName; //驱动名称

private String url;//url 例如127.0.0.1

private String iport; //端口 例如： 3306

private String username;//用户名称

private String password;//用户密码

private String databaseName; //数据库名称

private String tableName; //表名称

private String xData; //x轴数据列名

private String yData; //y轴数据列名
```

**第二步 将配置类传入到工具类，然后工具类执行数据获取操作，将取到的数据存入到一个实体类然后返回给前端进行渲染生成折线图就可以了。**

工具类就直接修改上面的jdbc代码当中的5个参数，将获取到的数据进行封装成实体类返回给前端就可以了。

**重点：**由于列名是动态的，返回的结果集获取数据时，直接通过列的顺序来获取就行了。（列是从左右编号的，并且从列1开始）当然也可以通过指定别名来获取数据。

**难点 主要是处理动态数据的接收问题。**

## 参考资料

JDBC详解：<https://www.cnblogs.com/erbing/p/5805727.html>