JDBC操作数据库

Jdbc基本使用流程

1. 导入连接mysql或其他数据库的jar包，java项目中，建立lib目录复制jar包进去，右键建立路径即可使用，但是在web项目中，要将其加入WEB-INF/lib目录下才能使用
2. 使用 Class.Forname(“com.mysql.jdbc.Driver”); 加载驱动
3. 建立数据库连接
   1. Connection conn = DriverManager.getConnection(url , name , password);

Url : 连接数据库地址 ，jdbc:mysql://localhost:3306/具体库名

1. 执行sql语句

使用Statement或者PreperedStatement对象进行数据库操作，推荐使用PreperedStatement对象,效率高，防注入，方便观看

1. 获取结果

获取结果用到ResultSet结果集接口

1. 关闭资源

实际案例

public class MysqlModel{

private string url = null;

private String username=null;

private String password=null;

private String driverClass = null;

private Statement stat = null;

// construct code block

{

//get config file

Properties pro = new Propreties();

//获取配置文件的内容

1，FileInputStream in = new FileInputStream("./src/db.properties"); （web项目不行）

注意：第一种获取文件内容的方式只能在java项目中使用，不能在web项目中这样写，理由如下

在java项目中， ./ 代表的路径就在本项目根目录下，所以可以获取到配置文件

在java web项目中，./代表在tomcat/bin/目录下， 这样写不能获取到配置文件，服务器报错，src目录下的配置文件会被放到WEB-INF/classes/目录下

1. InputStream in = MysqlModel.class.getResourceAsStream("/db.properties");

(web项目用这个，路径里面不要加. )

这种方式是类路径的读取方式

/ 斜杠表示classpath的根目录

在java项目中，classpath的根目录从项目的bin目录开始

在java web项目中，classpath的根目录从/WEB-INF/classes目录开始

// load config

pro.load(in);

//get config information

url = pro.getProperty("url");

username= pro.getProperty("username");

password= pro.getProperty("password");

driverclass= pro.getProperty("driverclass");

//load Driver

Class.forName(driverclass);

//get mysql connection

Connection conn = DriverManager.getConnection(url , username , password);

//crearte preparedStatement

String sql = "select \* form test where name=? and age=? and money=? ";

这里的？代表一个变量的占位符

stat= conn.prepareStatement(sql);

//设置参数

stat.setString(1, "alex");

stat.setInt(2, 14);

stat.setDouble(3, 14.11);

//run query

ResultSet res = stat.executeQuery();

//get res

while(res.next()){

String name = res.getString("name");

Int age = res.getInt("age");

Double money = res.getDouble("money");

}

//close connection

res.close();

stat.close();

conn.close();

}

}

jdbc核心api

java.sql.\* 和 javax.sql.\*

|- Driver接口： 表示java驱动程序接口。所有的具体的数据库厂商要来实现此接口。

|- connect(url, properties): 连接数据库的方法。

url: 连接数据库的URL

URL语法： jdbc协议:数据库子协议://主机:端口/数据库

user： 数据库的用户名

password： 数据库用户密码

|- DriverManager类： 驱动管理器类，用于管理所有注册的驱动程序

|-registerDriver(driver) : 注册驱动类对象

|-Connection getConnection(url,user,password); 获取连接对象

|- Connection接口： 表示java程序和数据库的连接对象。

|- Statement createStatement() ： 创建Statement对象

|- PreparedStatement prepareStatement(String sql) 创建PreparedStatement对象

|- CallableStatement prepareCall(String sql) 创建CallableStatement对象

|- Statement接口： 用于执行静态的sql语句

|- int executeUpdate(String sql) ： 执行静态的更新sql语句（DDL，DML）

|- ResultSet executeQuery(String sql) ：执行的静态的查询sql语句（DQL）

|-PreparedStatement接口：用于执行预编译sql语句

|- int executeUpdate() ： 执行预编译的更新sql语句（DDL，DML）

|-ResultSet executeQuery() ： 执行预编译的查询sql语句（DQL）

|-CallableStatement接口：用于执行存储过程的sql语句（call xxx）

|-ResultSet executeQuery() ： 调用存储过程的方法

|- ResultSet接口：用于封装查询出来的数据

|- boolean next() ： 将光标移动到下一行

|-getXX() : 获取列的值

扩展

获取sql语句中的占位符数量

int count = stat.getParameterMetaData().getParameterCount();