**使用jdbc访问mysql**

1.在idea中新建 marven项目

2.添加 mysql 的驱动的依赖

<dependencies>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.13</version>

<scope>runtime</scope>

</dependency>

</dependencies>

3.在src下新建package test ,并添加 mysql 帮助类 JDBCUtils

import java.sql.\*;

public class JDBCUtils {

private static Connection con;

static {

try {

//初始化MySQL的Driver类

Class.forName("com.mysql.cj.jdbc.Driver");

String url ="jdbc:mysql://192.168.125.200:3306/testdb?useSSL=false";

String user = "xl";

String password ="1010";

//连接数据库

con = DriverManager.getConnection(url, user, password);

} catch(Exception e) {

throw new RuntimeException(e + ",数据库连接失败！");

}

}

public static Connection getConnection() {

return con;

}

public static void close(Connection con, Statement state) {

if(con != null) {

try {

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(state != null) {

try {

state.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void close(Connection con, Statement state, ResultSet rs) {

if(con != null) {

try {

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(state != null) {

try {

state.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(rs != null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

4.在 mysql数据库上新建对应的 数据库和表

5.在package 下新建Main 类，并新建main方法

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class Main {

public static void main(String[] args) throws SQLException {

Connection con = JDBCUtils.getConnection();

String sql = "SELECT \* FROM student";

//创建PreparedStatement对象，并将sql语句发送到数据库

PreparedStatement pst = con.prepareStatement(sql);

//取得执行后的结果集

ResultSet rs = pst.executeQuery();

//输出sort表中第二列的所有数据

while(rs.next()) {

System.out.println(rs.getString(2));

}

JDBCUtils.close(con, pst, rs);

}

}

1. 固定jdk 的版本

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

1. 打包成jar后，自动添加清单

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-jar-plugin</artifactId>

<version>3.2.0</version>

<configuration>

<archive>

<manifest>

<addClasspath>true</addClasspath>

<mainClass>test.Main</mainClass> <!-- 此处为主入口-->

</manifest>

</archive>

</configuration>

</plugin>

</plugins>

</build>

1. 通过maven 的package命令，打包成jar包

在命令行中运行，如果提示找不到驱动，可以手工将 jdbc驱动的jar包合并到输出的jar包中