JDBC

JDBC（Java Database Connectivity）Java 数据库连接

JDK只提供一套访问的接口，具体的实现是由各大数据库厂商提供

常用的类：Connection， Statement， Resultset

jdbc案例

创建maven工程，引入JDBC 8.0的驱动包

```xml
    <!-- mysql 8.x JDBC驱动包 -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.22</version>
    </dependency>
```

创建获取Connection, 执行sql的工具类

```java
package com.hqyj.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class DbUtil {

    public static Connection getConnection() {
        Connection connection = null;
        String url ="jdbc:mysql://localhost:3306/java201102?serverTimezone=Asia/Shanghai"; //8.x驱动必须要添加时区参数
        String user = "root";
        String password = "123456";
        try {
            connection = DriverManager.getConnection(url, user, password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }

    public static ResultSet query(Connection conn, String sql) {
        ResultSet resultSet = null;
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            resultSet = statement.executeQuery();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return resultSet;
    }

}
```

测试代码

```
package com.hqyj.jdbc;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class JdbcTest {
    private Connection connection;

    @Before
    public void init() {
        connection = DbUtil.getConnection();
    }

    @Test
    public void testConnection() {
        String sql = "select * from emp";
        ResultSet rs = DbUtil.query(connection, sql);
        try {
            while(rs.next()) {// 移动到下一行，如果没有数据返回false, 反之返回true
                System.out.print(rs.getInt(1) + "\t");
                String empName = rs.getString("emp_name"); //读取列的值时，参数可以用字段名
                System.out.print(empName + "\t");
                System.out.println(rs.getInt(3));  //读取列的值时，参数也可以用序号，从1开始
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally {
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    @After
    public void destory() {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

员工表增删改查（CRUD)

员工表

```
CREATE TABLE `emp`  (
  `emp_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '员工编号',
  `emp_name` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '员工姓名',
  `emp_age` int(11) NOT NULL COMMENT '年龄',
  PRIMARY KEY (`emp_id`) USING BTREE
)
```

创建员工表对应的实体

```
package com.hqyj.jdbc.entity;

public class Emp {
    private int empId;
    private String empName;
    private int empAge;

    public Emp() {
    }
    public Emp(int empId, String empName, int empAge) {
        this.empId = empId;
        this.empName = empName;
        this.empAge = empAge;
    }


    public Emp(String empName, int empAge) {
        this.empName = empName;
        this.empAge = empAge;
    }
    public int getEmpId() {
        return empId;
    }
    public void setEmpId(int empId) {
        this.empId = empId;
    }
    public String getEmpName() {
        return empName;
    }
    public void setEmpName(String empName) {
        this.empName = empName;
    }
    public int getEmpAge() {
        return empAge;
    }
    public void setEmpAge(int empAge) {
        this.empAge = empAge;
    }
    @Override
    public String toString() {
        return "Emp [empId=" + empId + ", empName=" + empName + ", empAge=" + empAge + "]";
    }


}
```

创建DAO类

```
package com.hqyj.jdbc.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import com.hqyj.jdbc.entity.Emp;

//Dao:  data access object
public class EmpDao {

    /*
     * 分页查询员工列表
     */
    public List<Emp> selectEmpByPage(Connection conn, int page, int rows){
        List<Emp> ret = new ArrayList<>();
        String sql = "select * from emp limit ?,?";
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setInt(1, (page-1)*rows); //计算分页起始行的公式：(page-1)*rows
            statement.setInt(2, rows);//每页行数
            ResultSet rs = statement.executeQuery();
            while(rs.next()) {
                Emp emp = new Emp();
                emp.setEmpId(rs.getInt("emp_id"));
                emp.setEmpName(rs.getString("emp_name"));
                emp.setEmpAge(rs.getInt("emp_age"));
                ret.add(emp);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return ret;
    }
    /*
     * 根据ID查询员工信息。如果查询到了返回Emp对象，如果没有查到，返回null
     */
    public Emp selectEmpById(Connection conn, int empId) {
        Emp emp = null;
        String sql = "select * from emp where emp_id=? ";
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setInt(1, empId);
            ResultSet rs = statement.executeQuery();
            if(rs.next()) {
                emp = new Emp();
                emp.setEmpId(rs.getInt("emp_id"));
                emp.setEmpName(rs.getString("emp_name"));
                emp.setEmpAge(rs.getInt("emp_age"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return emp;
    }

    /*
     * 插入员工信息
     */
    public void insertEmp(Connection conn, Emp emp) {
        String sql = "insert into emp(emp_name,emp_age) values(?,?)";//预处理的
SQL，用?占位，代表要插入的值
        try {
            PreparedStatement statement = conn.prepareStatement(sql);//创建一个预处
理statement
            statement.setString(1, emp.getEmpName()); //设置参数（SQL里面的？），下标
从1开始
            statement.setInt(2, emp.getEmpAge());
            statement.executeUpdate(); //执行插入动作
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    /*
     * 修改员工信息
     */
    public void updateEmp(Connection conn, Emp emp) {
        String sql = "update emp set emp_name=?,emp_age=? where emp_id=?";
        try {
            conn.setAutoCommit(false);  //把自动提交事务设置为假（需要代码commit）
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setString(1, emp.getEmpName());
            statement.setInt(2, emp.getEmpAge());
            statement.setInt(3, emp.getEmpId());
            statement.executeUpdate();
            conn.commit(); //提交事务
//          conn.rollback();//回滚事务
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    /*
     * 根据id删除员工信息
     */
    public void deleteEmpById(Connection conn, int empId) {
        String sql = "delete from emp where emp_id=?";
        try {
            PreparedStatement statement = conn.prepareStatement(sql);
            statement.setInt(1, empId);
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

}
```

测试代码

```
package com.hqyj.jdbc;

import java.sql.Connection;
import java.util.List;

import org.junit.Before;
import org.junit.Test;

import com.hqyj.jdbc.dao.EmpDao;
import com.hqyj.jdbc.entity.Emp;

public class EmpTest {
    private Connection connection;
    private EmpDao empDao;

    @Before
    public void init() {
        connection = DbUtil.getConnection();
        empDao = new EmpDao();

    }

    @Test
    public void testInsert() {
        Emp emp = new Emp("Jack", 21);
        empDao.insertEmp(connection, emp);

    }

    @Test
    public void testUpdate() {
        Emp emp = new Emp(100002,"tom",20);
        empDao.updateEmp(connection, emp);
    }

    @Test
    public void testDeleteEmpById() {
        empDao.deleteEmpById(connection, 100004);
    }

    @Test
    public void testSelectEmpById() {
        Emp emp = empDao.selectEmpById(connection, 100002);
//      if(emp != null) {
//          emp.getEmpName();
//      }
        System.out.println(emp);
    }

    @Test
    public void testSelectEmpByPage() {
        List<Emp> list = empDao.selectEmpByPage(connection, 2, 3);//查询第二页，每页3行
        for(Emp emp : list) {
            System.out.println(emp);
        }
    }
}
```