

# 一、DBUtils介绍 apache

## 什么是dbutils，它的作用

DBUtils是java编程中的数据库操作实用工具，小巧简单实用。

DBUtils封装了对JDBC的操作，简化了JDBC操作。可以少写代码。

1. 对于数据表的读操作，他可以把结果转换成List，Array，Set等java集合，便于程序员操作；
2. 对于数据表的写操作，也变得很简单（只需写sql语句）
3. 可以使用数据源，使用JNDI，数据库连接池等技术来优化性能--重用已经构建好的数据库连接对象

## 二、DBUtils的三个核心对象

- > QueryRunner类
- > ResultSetHandler接口
- > DBUtils类

### QueryRunner类

QueryRunner中提供对sql语句操作的API.

它主要有三个方法

query() 用于执行select

update() 用于执行insert update delete

batch() 批处理

### ResultSetHandler接口

用于定义select操作后，怎样封装结果集.

### DbUtils类

它就是一个工具类, 定义了关闭资源与事务处理的方法

## 三、Dbutils快速入门

- > 导入jar包
- > 创建QueryRunner对象

- › 使用query方法执行select语句
- › 使用ResultSetHandler封装结果集
- › 使用DbUtils类释放资源

### **Dbutils快速入门实现步骤:**

创建数据库及表

```
CREATE DATABASE day14;
```

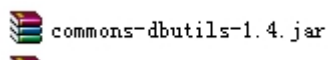
```
USE day14;
```

```
create table account(  
id int primary key auto_increment,  
name varchar(50),  
money float  
);
```

```
insert into account(name,money) values('aaa',1000);  
insert into account(name,money) values('bbb',1000);  
insert into account(name,money) values('ccc',1000);
```

```
SELECT * FROM account;
```

第一步:导入jar包



注意: c3p0与mysql驱动jar也要导入。

```
//创建QueryRunner对象  
//使用query方法执行select语句  
//使用ResultSetHandler封装结果集  
//使用DbUtils类释放资源
```

## **四、QueryRunner对象**

### **1.1 构造函数:**

- › new QueryRunner(); 它的事务可以手动控制。

也就是说此对象调用的方法（如：query、update、batch）参数中要有Connection对象。

> new QueryRunner(dataSource ds); 它的事务是自动控制的。一个sql一个事务。

此对象调用的方法（如：query、update、batch）参数中无需Connection对象。

## 1.2 方法

### 进行基本的CRUD操作：练一下

```
public class TestQueryRunner {
    @Test
    public void testInsert() throws SQLException{
        QueryRunner qr = new QueryRunner(C3P0Util.getDataSource());
        //qr.update("INSERT INTO student(NAME,birthday) VALUES(?,?)", "aaa","2015-08-31");
        qr.update("INSERT INTO student(NAME,birthday) VALUES(?,?)", "bbb",new Date());
    }
    @Test
    public void testUpdate() throws SQLException{
        QueryRunner qr = new QueryRunner(C3P0Util.getDataSource());
        qr.update("update student set name=?, birthday=? where id=?", "ccc",new Date(),1);
    }
    @Test
    public void testDelete() throws SQLException{
        QueryRunner qr = new QueryRunner(C3P0Util.getDataSource());
        qr.update("delete from student where id=?",1);
    }
    @Test
    public void testBatch() throws SQLException{
        QueryRunner qr = new QueryRunner(C3P0Util.getDataSource());
        Object[][] params = new Object[10][]; //高维表示要执行多少条sql语句
        for (int i = 0; i < params.length; i++) {
```

## 五、ResultSetHandler接口

### ResultSetHandler下的所有结果处理器

//ArrayHandler:适合取1条记录。将该条记录的每列值封装到一个数组中Object[]

//ArrayListHandler:适合取多条记录。把每条记录的每列值封装到一个数组中Object[]，把数组封装到一个List中

//ColumnListHandler:取某一列的数据。封装到List中。

//KeyedHandler:取多条记录，每一条记录封装到一个Map中，再把这个Map封装到另外一个Map中，key为指定的字段值。

//MapHandler:适合取1条记录。把当前记录的列名和列值放到一个Map中

//MapListHandler:适合取多条记录。把每条记录封装到一个Map中，再把Map封装到List中

//ScalarHandler:适合取单行单列数据

BeanHandler

BeanListHandler

## 六、DBUtils控制事务的开发

## 七、ThreadLocal

模拟ThreadLocal的设计，让大家明白他的作用。

```
public class ThreadLocal {  
    private Map<Runnable, Object> container = new HashMap<Runnable, Object>();  
    public void set(Object value) {  
        container.put(Thread.currentThread(), value); //用当前线程作为key  
    }  
    public Object get() {  
        return container.get(Thread.currentThread());  
    }  
    public void remove() {  
        container.remove(Thread.currentThread());  
    }  
}
```

总结：调用该类的get方法，永远返回当前线程放入的数据。线程局部变量。

## 八、完成案例

### 1、项目开发的做准备工作

创建数据库及表

```
create database day14;
```

```
use day14;
```

```
create table books(  
    id varchar(200) primary key,  
    name varchar(100) not null,  
    ...  
);
```

```
price double,  
pnum int,  
category varchar(50),  
description varchar(200)  
)
```

```
SELECT * FROM books;
```

```
INSERT INTO books VALUES('1001','java编程思想',98,100,'计算机','好书');
```

```
INSERT INTO books VALUES('1002','金瓶梅',10,20,'言情','好书');
```

```
INSERT INTO books VALUES('1003','金瓶梅2',20,30,'武侠','好书');
```

## 2、创建项目

### 2、1添加jar包

### 3、添加Dao、service、servlet分层代码（参见分析图）