# 1Dbutils

## 简介

* 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

## 三个核心对象

### 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

## dbutils入门案例

### 数据库准备

|  |
| --- |
| 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 double  );  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; |

### 查询数据案例

* **Step1创建一个项目导入jar包，配置C3P0**

|  |
| --- |
|  |
|  |

* **Step2创建QueryRunner对象**
* **Step3使用query方法执行select语句**
* **Step4 ResultSetHandler封装结果集**
* **Step5使用DbUtils类释放资源**

|  |
| --- |
|  |
|  |

### 添加删除更新案例

|  |
| --- |
| 插入 |
|  |
| 删除 |
|  |
| 更新 |
|  |

## ResultSetHandler下的处理器

### ArrayHandler

适合取1条记录。把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.**默认第一个为KEY**。

|  |
| --- |
|  |
|  |

### 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

适合取单行多列数据

|  |
| --- |
|  |

### BeanListHandler

适合多行多列

|  |
| --- |
|  |

# 2转帐案例

## 方式一 无事务

|  |
| --- |
| AccountDaoImpl.java |
|  |
| AccountServiceImpl.java |
|  |
| 测试 |
|  |

## 方式二 添加事务

|  |
| --- |
| 更改Dao代码结构，传入一个connection |
|  |
| 更改Service代码结构 |
|  |

## ThreadLocal【本地线程】

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。

|  |
| --- |
|  |

## 方式三 在Service中不显示Connection

|  |
| --- |
| 在本地线程中存Connection |
|  |
| 修改AccountDaoImpl代码 |
|  |
| 修改AccountServiceImpl代码 |
|  |

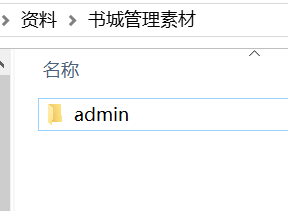
# 3综合案例-图书管理

## 3.1创建表

|  |
| --- |
| 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)  );  INSERT INTO books VALUES('1001','java编程思想',98,100,'计算机','好书');  INSERT INTO books VALUES('1002','转角遇到爱',10,20,'言情','好书');  INSERT INTO books VALUES('1003','天龙八部',20,30,'武侠','好书');  SELECT \* FROM books; |

## 3.2创建web项目

* 创建一个javaweb项目，**然后把项目编码改成utf-8**
* 把素材admin文件夹放入**WebContent**中



* **导入jstl包,因为内部的jsp页面使用了jstl标签**
* **导入beanutils和dbutils包**
* **mysql驱动 ，数据库连接池的包c3p0,dbcp**
* 启动tomcat方法http://localhost:8080/javaweb\_day14\_bookstore/admin/login/home.jsp

|  |
| --- |
|  |

## 3.3 dtree的使用

dtree是一个js库,显示一个树型结构

dtree的官网http://www.destroydrop.com/javascripts/tree/

|  |
| --- |
|  |

dtree的使用步骤

1. 在官网下载dtree.zip
2. 解压后把文夹放入项目中

## 3.4显示图书列表功能



### 书的JavaBean

|  |
| --- |
|  |

### 写BookDaoImpl

|  |
| --- |
| 先导入前面写的好C3P0工具类 |
|  |

### 写BookServiceImpl

|  |
| --- |
|  |

### 写BookListServlet

|  |
| --- |
|  |
| 修改“商品查看”的访问路径为BookListServlet |

### 修改list.jsp数据显示

|  |
| --- |
|  |

## 3.5添加图书信息功能

|  |
| --- |
|  |

### 修改add.jsp的action

enctype="application/x-www-form-urlencoded" 用于普通表单提交

enctype="multipart/form-data" 用于有文件的表单提交

|  |
| --- |
|  |

### 创建AddBooServletServlet

|  |
| --- |
| 先把表单的请求参数封装到Book中来 |
|  |

### dao添加一个插入书的方法

|  |
| --- |
|  |

### service添加一个插入书的业务

|  |
| --- |
|  |

### 修改AddBookServlet的方法

|  |
| --- |
|  |

## 3.6回显修改图书信息功能

### 在dao添加一个查找方法

|  |
| --- |
|  |

### 在service添加一个查找的业务方法

|  |
| --- |
|  |

### 修改list.jsp 编辑的代码

|  |
| --- |
|  |
|  |

### 创建一个FindBookByIdServlet

|  |
| --- |
|  |

### 显示编辑页面数据

|  |
| --- |
|  |

### 显示商品类别的默认值

|  |
| --- |
|  |

## 3.7完成修改功能

### 修改edit.jsp表单的提交路径

|  |
| --- |
|  |

### 在dao中添加一个updateBook方法

|  |
| --- |
|  |

### 在service中添加一个更新书的业务方法

|  |
| --- |
|  |

### 在编辑的表单中添加个id隐藏字段

|  |
| --- |
| **注：因为更新Book需要用到id,而id又不需要显示在页面上，所以使用隐藏字段** |
|  |

### 创建一个UpdateBookServlet

|  |
| --- |
|  |
|  |

## 3.8删除图书功能

### 修改list.jsp删除按键的访问地址

|  |
| --- |
|  |

### 添加一个DeleteBookByIdServelet

|  |
| --- |
|  |

### service中添加一个删除图书的方法

|  |
| --- |
|  |

### dao中添加一个删除图书的方法

|  |
| --- |
|  |

## 3.9删除图书前给个确认提示框

|  |
| --- |
|  |

### 修改删除按钮的href

|  |
| --- |
|  |

### 在list.jsp页面添加一段js代码

|  |
| --- |
|  |

## 3.10实现删除图书的全选和不全选

### 在list页面中添加多选框

|  |
| --- |
|  |
|  |

### js实现全选，和不全选

|  |
| --- |
|  |

## 3.11批量删除实现

### 给每一本书的复选框架添加个ID

|  |
| --- |
|  |

### 添加一个批量删除按钮，调用js方法

|  |
| --- |
|  |
|  |

### 实现批量删除的js方法

|  |
| --- |
|  |

### 添加一个DeleteAllBookServlet

|  |
| --- |
|  |

### service中添加一个删除所有书的方法

|  |
| --- |
|  |

### dao中添加一个删除所有书的方法

|  |
| --- |
|  |

## 3.12 多条件查询功能实现

|  |
| --- |
|  |

### 创建一个SearchBookServlet

|  |
| --- |
|  |

### service中添加一个findBookByCondition方法

|  |
| --- |
|  |

### dao中添加一个findBookByCondition方法

|  |
| --- |
|  |

## 3.13 实现分页

|  |
| --- |
|  |
| mysql的分页查询  limit start,length |

### 封装一个分页数据模型

|  |
| --- |
| 这里采用泛型，方便代码复用  也有人取名PageBean |
|  |

### 在Dao中添加一个分页查询方法

|  |
| --- |
|  |

### 在BookListServlet中重写分页代码

|  |
| --- |
|  |

### 测试分页

<http://localhost:8080/javaweb_day14_bookstore/BookListServlet?page=2>

<http://localhost:8080/javaweb_day14_bookstore/BookListServlet?page=1>

### 集成分页在list.jsp中

|  |
| --- |
|  |

# 4 WEB-INF

jsp页面可以放在WEB-INF中

1.如果是把jsp放在WebContent的目录下，

http://localhost:8080/day15-20180329-bookstore/lesson04.jsp

在浏览器访问的时候，可以直接访问项目名/.jsp名字

这种方式页面的数据不安全

2.如果是把jsp放在WEB-INF的目录下,这个页面的数据比较安全

如果想访问WEB-INFO下的jsp，只能通过Servlet转发