

DbUtils 入门

一、DbUtils 简介

Commons DbUtils 是 Apache 组织提供的一个对 JDBC 进行简单封装的开源工具类库，使用它能够简化 JDBC 应用程序的开发，同时也不会影响程序的性能。

JDBC 的操作复杂、且非常容易出错。因此，DbUtils 可以让我们从这些复杂繁琐的工作中解放出来，是我们更加的专注于数据库的相关操作。

二、使用 DbUtils 的优势

- 完善的封装，使我们可以编写更加简洁的代码，是我们可以专注于数据库相关操作。
- 自动将结果集封装为 JavaBean，而不需要在通过手动封装对象，转换类型。
- 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、核心类

- DbUtils
 - 数据库操作基本类，提供了数据库连接相关操作的静态方法。
 - ◆ public static void close(Connection conn)
 - 关闭数据库连接
 - ◆ public static void close(ResultSet rs)
 - 关闭结果集
 - ◆ public static void close(Statement stmt)
 - 关闭 Statement
 - ◆ public static void commitAndClose(Connection conn)
 - 提交是否并关闭连接
- QueryRunner
 - 用于执行数据库操作的类，主要使用的类
 - ◆ batch()
 - 批量操作数据库
 - ◆ insert()
 - 向数据库中插入一条数据，可以返回插入数据的自增 ID
 - ◆ insertBatch()
 - 向数据库中批量插入多条数据
 - ◆ query()
 - 从数据库中查询数据
 - ◆ update()
 - 修改数据库中的内容，返回修改的条数
 - 该类中给我们提供了大量的重载方法来调用，我们可以根据不同

的需要调用不同的方法。

- **ResultSetHandler**
 - **ResultSetHandler** 是一个接口，我们操作数据库时，最麻烦的问题就是从数据库中查询到的数据封装为对象，这是一个非常繁琐，非常容易出错的操作。这个接口就是帮我们解决这个问题。而更好的是他已经给我们提供了足够多的实现类，而不用我们再去自己实现。
 - **ArrayHandler**
 - ◆ 将查询到的数据封装为一个 **Object** 数组
 - **ArrayListHandler**
 - ◆ 将查询到的数据封装为一个 **List<Object[]>**
 - **BeanHandler**
 - ◆ 返回一个对象
 - **BeanListHandler**
 - ◆ 返回对象的集合（**List**）
 - **BeanMapHandler**
 - ◆ 返回一个 **map**，**map** 的 **key** 为第一个数据，值为一行数据
 - **ColumnListHandler**
 - ◆ 返回第一列的数据
 - **KeyedHandler**
 - ◆ 返回一个 **map**，**map** 的 **key** 为第一个数据，值为一行数据
 - **MapHandler**
 - ◆ 将查询到的数据封装为一个 **Map<String,Object>**
 - **MapListHandler**
 - ◆ 将查询到数据封装为一个 **List< Map<String,Object>>**
 - **ScalarHandler**
 - ◆ 返回一个单独的结果

四、 示例代码

1. 环境准备

在数据库中创建一个表 **user**

```
CREATE TABLE user (  
    id INT(11) AUTO_INCREMENT PRIMARY KEY,  
    `name` VARCHAR(20),  
    age INT(11)  
)
```

2. 创建 User 用于封装数据

```
public class User {  
    private Integer id;  
    private String name;  
    private Integer age;  
    //省略部分代码
```

```
}
```

3. 创建 TestDB 测试类

a) 获取数据库连接的静态方法

```
public static Connection getConnection() {  
    Connection conn = null;  
    try {  
        Class.forName("com.mysql.jdbc.Driver");  
        conn =  
        DriverManager.getConnection("jdbc:mysql:///test","root","  
root");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return conn;  
}
```

4. 插入数据方法

```
public void save() {  
    QueryRunner runner = new QueryRunner();  
    Connection conn = getConnection();  
    String sql = "INSERT INTO user (name,age)  
values(?,?)";  
    try {  
        Long id = runner.insert(conn, sql, new  
ScalarHandler<Long>(), "sunwukong", 20 );  
        System.out.println(id);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally{  
        DbUtils.closeQuietly(conn);  
    }  
}
```

5. 查询一个对象

```
public void queryOne() {  
    QueryRunner runner = new QueryRunner();  
    Connection conn = getConnection();  
    String sql = "SELECT id , name , age FROM user";  
    try {  
        User user = runner.query(conn, sql, new  
BeanHandler<User>(User.class));  
        System.out.println(user);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

```
    } finally{  
        DbUtils.closeQuietly(conn);  
    }  
}
```

6. 查询多个对象

```
public void queryMore(){  
    QueryRunner runner = new QueryRunner();  
    Connection conn = getConnection();  
    String sql = "SELECT id , name , age FROM user";  
    try {  
        List<User> list = runner.query(conn, sql, new  
BeanListHandler<User>(User.class));  
        System.out.println(list);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally{  
        DbUtils.closeQuietly(conn);  
    }  
}
```

7. 修改

```
public void update(){  
    QueryRunner runner = new QueryRunner();  
    Connection conn = getConnection();  
    String sql = "UPDATE user set name=? , age=? WHERE id=?";  
    try {  
        runner.update(conn, sql, "admin2",12,1);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally{  
        DbUtils.closeQuietly(conn);  
    }  
}
```

五、 总结

经过上面的代码不难发现，DbUtils 的确是简化了我们的开发。但是可以注意到的是 DbUtils 的最大优势并不是修改和插入，因为即使原生的 JDBC 修改和插入也并不是多繁琐。DbUtils 最大的优势是查询，DbUtils 为我们提供了大量的结果集处理器，使我们可以很方便的将查询到结果转换为各种对象，极大的简化了我们的开发，所以掌握 DbUtils 最重要的就是要熟悉它为我们提供的众多结果集。