# 1. BeanUtils组件

**使用BenUtils组件：**

1. 引入commons-beanutils-1.8.3.jar核心包

2. 引入日志支持包: commons-logging-1.1.3.jar

**基本用法:**

方法1： 对象属性的拷贝

BeanUtils.copyProperty(admin, "userName", "jack");

BeanUtils.setProperty(admin, "age", 18);

方法2： 对象的拷贝

BeanUtils.copyProperties(newAdmin, admin);

方法3： map数据拷贝到javabean中

【注意：map中的key要与javabean的属性名称一致】

BeanUtils.populate(adminMap, map);

**日期类型的拷贝**

// 注册日期类型转换器：1， 自定义的方式

ConvertUtils.register(new Converter() {

// 转换的内部实现方法，需要重写

@Override

public Object convert(Class type, Object value) {

// 判断

if (type != Date.class) {

return null;

}

if (value == null || "".equals(value.toString().trim())) {

return null;

}

try {

// 字符串转换为日期

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

return sdf.parse(value.toString());

} catch (ParseException e) {

throw new RuntimeException(e);

}

}

},Date.class);

// 注册日期类型转换器：2， 使用组件提供的转换器工具类

ConvertUtils.register(new DateLocaleConverter(), Date.class);

# 2、DBUtils组件

|-- DbUtils 关闭资源、加载驱动（全部为静态方法）

closeQuietly(Connection conn, Statement stmt, ResultSet rs)；quietly表示是安全的已经捕捉异常了。

其他包括了con、 stmt 、rs 分别单独关闭的方法

|-- QueryRunner 组件的核心工具类：定义了所有的与数据库操作的方法(查询、更新)

Int update(Connection conn, String sql); 执行更新带一个占位符的sql

Int update(Connection conn, String sql, Object param); 执行更新带一个占位符的sql

Int update(Connection conn, String sql, Object… param); 执行更新带多个占位符的sql

Int[] batch(Connection conn, String sql, Object[][] params) 批处理

T query(Connection conn ,String sql, ResultSetHandler<T> rsh, Object... params) 查询方法

Int update( String sql, Object param);

Int update( String sql, Object… param);

Int[] batch( String sql, Object[][] params)

注意： 如果调用DbUtils组件的操作数据库方法，没有传入连接对象，那么在实例化QueryRunner对象的时候需要传入数据源对象： QueryRunner qr = new QueryRunner(ds);

DbUtils提供的封装结果的一些对象：

1） BeanHandler: 查询返回单个对象

2） BeanListHandler: 查询返回list集合，集合元素是指定的对象

3) ArrayHandler, 查询返回结果记录的第一行，封装对对象数组, 即返回：Object[]

4) ArrayListHandler, 把查询的每一行都封装为对象数组，再添加到list集合中

5) ScalarHandler 查询返回结果记录的第一行的第一列 (在聚合函数统计的时候用)

6) MapHandler 查询返回结果的第一条记录封装为map