

java executor包参数处理功能

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：<https://www.wtabcd.cn/fanwen/zuowen/3e59047ccfc33e46b9c3c3394165cfa7.html>

范文网，为你加油喝彩！

sql语句中的参数赋值是有由executor包中的parameter子包完成的。

parameter子包其实只有一个parameterhandler接口，它定义了2个方法：

```
public interface parameterhandler { object getparameterobject(); void setparameters(prepared statement ps) throws sqlexception;}
```

parameterhandler接口有一个默认的实现类defaultparameterhandler，它在scripting包的子包中。

mybatis中支免费读英语的软件持进行参数设置的语句类型是preparedstatement接口及其子接口callablestatement，所以setparameters的母子性爱输入参数是preparedstatement类型。

setparameters方法的实现逻辑就是依次取出每个参数的值，然名词的意动用法后根据参数类型调用preparedstatement中的赋值方法进行赋值。

```
public class defaultparameterhandler implements parameterhandler { // 类型处理器注册表
    private final typehandlerregistry typehandlerregistry; // mappedstatement对象（包含完整的
    增删改查节点信息） private final mappedstatement mappedstatement; // 参数对象 private f
    inal object parameterobject; // boundsql对象（包含sql语句、参数、实参信息） private f
    inal boundsql boundsql; // 配置信息 private final configuration configuration; public d
    efaultparameterhandler(mappedstatement mappedstatement, object paramet赞美竹子的古诗erobjec
    t, boundsql boundsql) { this.mappedstatement = mappedstatement; this.configuration
    = mappedstatement.getconfiguration(); this.typehandlerregistry = mappedstatement.getconfigura
    tion().gettypehandlerregistry(); this.parameterobject = parameterobject; this.boundsql = b
    oundsql; } @override public object getparameterobject() { return parameterobject; } /
    *** 为语句设置参数 * @param ps 语句 * / @override public void setparameters(pre
    paredstatement ps) { errorcontext.instance().activity("setting parameters").object(mappedstateme
    nt.getparametermap().getid()); // 取出参数列表 list<parametermapping> parametermapping
    s = boundsql.getparametermappings(); if (parametermappings != null) { for (int
    i = 0; i < parametermappings.size(); i++) { parametermapping parametermapping
    = parametermappings.get(i); // parametermode.out是callablestatement的输出参数，已经
    单独注册。故忽略 if (parametermapping.getmode() != parametermode.out) {
    object value; // 取出属性名称 string propertyname = parametermapp
    ing.getProperty(); if (boundsql.hasadditionalparame湖北的大学排名ter(propertyname)) {
```

```
// 从附加参数中读取属性值 value = boundsql.getadditionalparamete  
r(propertyname); } else if (parameterobject == null) { value = nu  
ll; } else if (typehandlerregistry.hastypehandler(parameterobject.getClass())) {  
// 参数对象是基本类型，则参数对象即为参数值 value = parameterobject;  
} else { // 参数对象是复杂类型，取出参数对象的该属性值  
metaobject metaobject = configuration.newmetaobject(parameterobject); value =  
metaobject.getvalue(propertyname); } // 确定该参数的处理器 ty  
pehandler typehandler = parametermapping.gettypehandler(); jdbcType jdbcType =  
parametermapping.getjdbcType(); if (value == null && jdbcType == null) {  
jdbcType = configuration.getjdbcTypeforNull(); } try {  
// 此方法最终根据参数类型，调用java.sql.preparedStatement类中的参数赋值方法，对sql语句中  
的参数赋值 typehandler.setParameter(ps, i + 1, value, jdbcType); } c  
atch (TypeException | SQLException e) { throw new TypeException("could not  
set parameters for mapping: " + parametermapping + ". cause: " + e, e);  
} } } } }
```

到此这篇关于java executor包参数处理功能 的文章就介绍到这了,更多相关executor包参数处理功能内容请搜索www.887551.com以前的文章或继续浏览下面的相关文章希望大家以后多多支持www.887551.com！

更多作文 请访问 https://www.wtabcd.cn/fanwen/list/92_0.html

文章生成doc功能，由[范文网](http://www.fanwen.net)开发