

# 阿进的写字台

博客园 首页 新随笔 联系 订阅 管理

随笔 - 49 文章 - 0 评论 - 54

## mybatis 多个接口参数的注解使用方式 (@Param)

### 1 简介

#### 1.1 单参数

在 **Mybatis** 中，很多时候，我们传入接口的参数只有一个。对应接口参数的类型有两种，一种是基本的参数类型，一种是 **JavaBean**。

例如在根据主键获取对象时，我们只需要传入一个主键的参数即可。而在插入，更新等操作时，一般会涉及到很多参数，我们就使用 **JavaBean**。

#### 1.2 多参数

但是，在实际的情况中，我们遇到类似这样的情况可能：

1. 接口需要使用的参数多于一个；
2. 接口需要使用的参数又远少于对应 **JavaBean** 的成员变量，或者需要多个 **JavaBean** 对象；
3. 或者需要使用的参数对应 **JavaBean** 没有相应的成员变量。

比如 获取一段时间产生的日志信息，日志对应的 **JavaBean** 只有一个日期，那我们使用该 **JavaBean** 就无法满足我们的要求。

又比如我们进行模糊搜索，搜索条件只有两个，但对应的 **JavaBean** 有 **50+** 个成员变量，那创建对应的 **JavaBean** 就过于浪费了。

对此，我知道的有如下几种方法

### 2 多个接口参数的两种使用方式

#### 2.1 Map 方法(不推荐)

昵称：阿进的写字台  
园龄：1年6个月  
粉丝：47  
关注：4  
[+加关注](#)

2018年12月						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

<input type="text"/>	找找看
<input type="text"/>	谷歌搜索

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

`Map` 方法的使用很简单，就是将对应的参数以 **key-value** 的方式存储，key 对应 SQL 中的参数名字，value 对应需要传入的参数值。

以获取一段时间内存储的用户为例

### 2.1.1 创建接口方法

```
/**
 * 获取一段时间内的用户
 * @param params
 * @return
 */
List<Student> selectBetweenCreatedTime(Map<String, Object> params);
```

该方法返回的是多个记录，因此使用 `List` 作为返回值。

### 2.1.2 配置对应的SQL

```
<select id="selectBetweenCreatedTime" parameterType="java.util.Map" resultMap="BaseResultMap">
  select
  <include refid="Base_Column_List" />
  from student
  where gmt_created > #{bTime, jdbcType=TIMESTAMP} and gmt_created < #{eTime,
jdbcType=TIMESTAMP}
</select>
```

id 与 之前创建的方法名一样。

### 2.1.3 调用

```
@Test
public void testSelectBtweenCreatedTimeMap() {

    Map<String, Object> params = new HashMap<>();
    Calendar bTime = Calendar.getInstance();
    // month 是从0~11, 所以9月是8
    bTime.set(2018, Calendar.AUGUST, 29);
    params.put("bTime", bTime.getTime());

    Calendar eTime = Calendar.getInstance();
```

#### 我的标签

#### 我的标签

Java(26)

mybatis(17)

mybatis使用(7)

mybatis源码(6)

iot(6)

mqtt(6)

paho(6)

Thread(5)

多线程(4)

反射(3)

更多

#### 随笔分类

C/C++(1)

Clean Code

Java(8)

Java 进阶(15)

kafka(1)

linux(1)

mqtt(6)

mybatis-工具(1)

```
eTime.set(2018,Calendar.SEPTEMBER,2);
params.put("eTime", eTime.getTime());
SqlSession sqlSession = null;
try {
    sqlSession = sqlSessionFactory.openSession();

    StudentMapper studentMapper = (StudentMapper) sqlSession.getMapper(StudentMapper.class);
    List<Student> students = studentMapper.selectBetweenCreatedTime(params);
    for (int i = 0; i < students.size(); i++) {
        System.out.println(students.get(i));
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (sqlSession != null) {
        sqlSession.close();
    }
}
}
```

调用方法很简单，传入相应的 **Map** 参数即可。此时，**Map** 中的 key 对应。因此，在此例子中传入的参数

1. 传入一个 **key** 为 **btime** 的时间，作为开始时间；
2. 传入一个 **key** 为 **etime** 的时间，作为结束时间；

## 2.2 @Param 方法(推荐)

@Param 方法就是使用注解的方式，

### 2.2.1 创建接口方法

```
/**
 * 获取指定时间内的对象
 * @param pbTime 开始时间
 * @param peTime 结束时间
 * @return
 */
```

[mybatis-使用\(8\)](#)[mybatis-源码\(10\)](#)[paho\(5\)](#)[spring-boot使用](#)[tomcat\(1\)](#)[多线程那些事\(1\)](#)[集合框架那些事\(1\)](#)

### 随笔档案

[2018年12月 \(5\)](#)[2018年11月 \(5\)](#)[2018年10月 \(11\)](#)[2018年9月 \(8\)](#)[2018年8月 \(5\)](#)[2018年3月 \(3\)](#)[2018年1月 \(3\)](#)[2017年12月 \(9\)](#)

### linux

### 积分与排名

积分 - 29332

排名 - 17422

### 最新评论

```
List<Student> selectBetweenCreateTimeAnno(@Param("bTime")Date pbTime, @Param("eTime")Date peTime);
```

@Param("bTime") 就是告诉 mybatis , 参数 pbTime 在 SQL 语句中用 bTime 作为 key。

也就是说, mybatis 帮我们完成了调用时, 类似 params.put("bTime", pbTime) 这个过程。

### 2.2.2 配置 SQL 语句

```
<select id="selectBetweenCreateTimeAnno" resultMap="BaseResultMap">
  select
  <include refid="Base_Column_List" />
  from student
  where gmt_created > # {bTime, jdbcType=TIMESTAMP} and gmt_created < # {eTime,
jdbcType=TIMESTAMP}
</select>
```

此处的 bTime 对应\*\* @Param("bTime")\*\* 中的 bTime, 需要完全一致。

eTime 也是一样。

### 2.2.3 调用

在调用时, 不需要创建 `Map` 了, 只需要按参数提示传入对应的实际参数即可。

```
@Test
public void testSelectBtweenCreateTimeAnno() {

    Map<String, Object> params = new HashMap<>();
    Calendar bTime = Calendar.getInstance();
    // month 是从0~11, 所以9月是8
    bTime.set(2018, Calendar.AUGUST, 29);

    Calendar eTime = Calendar.getInstance();
    eTime.set(2018, Calendar.SEPTEMBER, 2);

    SqlSession sqlSession = null;
    try {
        sqlSession = sqlSessionFactory.openSession();
```

1. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

@灯塔下的守望者得到肯定, 很开心, 谢谢...

--阿进的写字台

2. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

@Java深入浅出这个判断很简单的。if ((name.startsWith("get") && name.length() > 3) || (name.startsW.....

--阿进的写字台

3. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

支持, 非常好

--灯塔下的守望者

4. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

怎么没写怎么判断 setter和getter?

--Java深入浅出

5. Re:HashMap是如何工作的

mark

--Java深入浅出

### 阅读排行榜

1. Paho -物联网 MQTT C Client的实现和详解(9890)

```
StudentMapper studentMapper = (StudentMapper) sqlSession.getMapper(StudentMapper.class);
List<Student> students = studentMapper.selectBetweenCreateTimeAnno(bTime.getTime(),
eTime.getTime());
for (int i = 0; i < students.size(); i++) {
    System.out.println(students.get(i));
}
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (sqlSession != null) {
        sqlSession.close();
    }
}
}
```

## 3 @Param 的优势

**Map 方式**的缺点就是需要手动创建 `Map`，并对 SQL 中的参数进行赋值。其缺点：

1. 手动创建 `Map` 这个过程很不简洁，看着很繁琐。
2. 手动对参数进行赋值，很容易出错。比如本来是要 `params.put("bTime", bTime)` 可能会不小心写成 `params.put("bime", bTime)`，但是这个时候编译器并不会提示。

相比于 `Map` 方式，使用 **@Param** 时，我们在使用上就像调用方法一样，传入对应的实际参数即可。调用时基本不会出错。

## 4 Github

相应代码，可以访问 [我的Github-helloMybatis](#)

## 一起学 mybatis

你想不想来学习 mybatis? 学习其使用和源码呢? 那么，在博客园关注我吧!!

我自己打算把这个源码系列更新完毕，同时会更新相应的注释。快去 star 吧!!

[mybatis最新源码和注释](#)

2. 运行时动态库: not found 及介绍-linux 的-Wl,-rpath命令(3508)

3. JDBC详解系列（一）之流程(3348)

4. Paho - MQTT C Client的实现(3036)

5. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(2198)

### 评论排行榜

1. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(14)

2. mybatis 多个接口参数的注解使用方式 (@Param)(7)

3. mybatis源码-解析配置文件（二）之解析的流程(3)

4. mybatis源码-原来resultMap解析完是这样(3)

5. Java设计模式-建造者(Builder)模式(2)

### 推荐排行榜

1. MyBatis动态SQL（认真看看，以后写SQL就爽多了）(15)

2. HashMap是如何工作的(5)

3. 扒一扒：Java 中的枚举(4)

4. mybatis源码-原来resultMap解析完是这样(4)

5. mybatis源码-Mapper解析之SQL 语句节点解析（一条语句对应一个MappedStatement)(2)

homejim / mybatis-cn

Watch 2 Unstar 16 Fork 4

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

mybatis源码的中文注释以及mybatis的使用和源码解析 Edit

mybatis mybatis-sources mybatis-chinese Manage topics

83 commits 1 branch 0 releases 1 contributor View license

Branch: master New pull request Create new file Upload files Find file Clone or download

作者: [阿进的写字台](#)

出处: <https://www.cnblogs.com/homejim/>

本文版权归作者和博客园共有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出原文连接, 否则保留追究法律责任的权利。

分类: [mybatis-使用](#)

标签: [Java](#), [mybatis](#), [mybatis使用](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

阿进的写字台

[关注 - 4](#)

[粉丝 - 47](#)

[+加关注](#)

1

0

« 上一篇: [mybatis源码-解析配置文件 \(四-1\) 之配置文件Mapper解析\(cache\)](#)

» 下一篇: [mybatis 代码生成器 \(IDEA, Maven\)及配置详解 \(部分配置你应该不知道\)](#)

posted @ 2018-10-09 10:36 阿进的写字台 阅读(460) 评论(7) 编辑 收藏

#1楼 2018-10-09 11:28 Java使徒

“又学了一招

支持(0) 反对(0)”

---

#2楼[楼主] 2018-10-09 11:34 阿进的写字台

“

@ Java使徒

有帮助那就最好啦， Github中有代码哦， 有问题欢迎提出

支持(0) 反对(0)”

---

#3楼 2018-11-16 09:12 大浪淘沙2018

“

在多条件查询的场景， 查询条件数量不定， 使用Map配合动态sql更灵活一点， 使用@Param方式反而不好处理吧

支持(0) 反对(0)”

---

#4楼[楼主] 2018-11-16 09:45 阿进的写字台

“

@ 大浪淘沙2018

@Param 的底层就是通过 Map 实现的， 所以是一样的。

只不过在可读性上面来看， 更建议用 @Param， Map 的话使用上可能会出错。@Param 通过函数参数的方式让 IDE 帮我们检查。

支持(0) 反对(0)”

---

#5楼[楼主] 2018-11-16 09:50 阿进的写字台

“

@ 大浪淘沙2018

个人觉得通过 @Param + 动态SQL的方式比 Map + 动态SQL 更好

支持(0) 反对(0)”

---

#6楼 2018-11-17 08:15 大浪淘沙2018

“

@ 阿进的写字台

使用@Param， 类似于开发语言的强类型定义， 可以让框架检查参数是否有误， 确实可以回避map内容不定带来的问题， 不过在实际业务系统中， 一张业务单据字段很多， 如果查询字段5个以内， 用@Param还好， 但超出了5个， 比如十几个， 则会导致函数参数个数过多， 以及调用不便的麻烦

支持(0) 反对(0)”

---

#7楼[楼主] 2018-11-19 10:45 阿进的写字台

@ 大浪淘沙2018

如果参数数量过多，我个人会定义一个对象，不使用Map。。在controller的时候直接就转化为该对象了。。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【活动】华为云12.12会员节 云产品1折起 满额送Mate20 点击抢购

【推荐】服务器100%基准CPU性能，1核1G首年168元，限时特惠!



腾讯云

## 腾讯云AMD云服务器

节省IT成本30%  
1核1G AMD机型**0.57元/天**起

[立即抢购](#)

#### 相关博文:

- 关于mybatis的@Param注解和参数
- @Param注解在Mybatis中的使用 以及传递参数的三种方式
- java @param参数注解
- Mybatis使用@Param
- Mybatis中@param注解

#### 最新新闻:

- 王兴: 20年到 C, 20年到 B



- 硅谷是个什么谷（第二十五章）： 更好的世界
  - 戴威的董事会股东的签字权 谁搞垮了ofo的资金和前途？
  - 携程有忧，建章难退
  - 史玉柱重出江湖三年： 属于他的时代，远去了
- » 更多新闻...

Copyright ©2018 阿进的写字台