

最佳实践

Lambda表达式

Lambda 表达式查询与更新是 MybatisPlus 与 Java 8 深度结合值得称赞的特性之一，在项目中作为技术规范 **强制使用**

查询示例

```
1  /**
2   * lambdaQuery示例，方法在service中
3   */
4  public List<Student> findStutentByName(String name){
5      // 方式1: 使用lambdaQuery
6      Wrapper<Student> query = Wrappers.<Student>lambdaQuery().like(Student::getName, name);
7      return list(query);
8
9      // *方式2: 使用ChainWrapper，代码更加简洁，推荐该方式
10     return lambdaQuery().like(Student::getName, name).list();
11 }
```

更新示例

```
1  /**
2   * lambdaUpdate示例，方法在service中
3   */
4  public void updateStudent(int studentId){
5      // 方式1: 使用lambdaUpdate，先写查询条件，后写设值语句
6      Wrapper<Student> update= Wrappers.<Student>lambdaUpdate().eq(Student::getId, studentId)
7          .set(Student::getName, "张三")
8          .set(Student::getAge, 20);
9      update(update);
10
11     // *方式2: 使用ChainWrapper，代码更加简洁，推荐该方式
12     lambdaUpdate().eq(Student::getId, studentId)
13         .set(Student::getName, "张三")
14         .set(Student::getAge, 20)
15         .update();
16 }
17
```

使用函数式接口映射表的列字段，在源码编写期间可以 IDE 字段检查功能，将 bug 暴露由运行

期提前至编译期，且后期如果需要修改字段名，可利用ide的重构功能方便批量修改

空值处理

空值引发的问题具体分为两类：

- 如果where条件字段值为 null，是否拼接到条件中；
- 如果update字段值为 null，是否添加到更新语句中；

主键索引查询

主键索引查询期望返回值至多只有一条数据，因此当属性为空时，应该查不到任何对象。

```
1 // 主键索引查询错误范例
2 lambdaQuery().eq(Objects.nonNull(name), Bean::getId, id).list();
3 // 主键索引查询正确范例
4 lambdaQuery().eq(Bean::getId, id).list();
```

当参数为空时，错误范例会构造空 where 查询条件，造成全表查询，背离了业务逻辑；正确范例业务正确，查询返回空对象或者空集合。

非主键索引查询

一般来讲根据条件查询，期望的查询结果是不确定大小的集合类型，当条件属性为空时，应该进行忽略

```
1 // 普通查询错误范例
2 lambdaQuery().eq(Bean::getName, name).list();
3 // 普通查询正确范例
4 lambdaQuery().eq(Objects.nonNull(name), Bean::getName, name).list();
```

集合属性空值 (in)

像如下示例，如果集合idList为空或者空集合，生成的sql是 `in ()`，执行会报错。

```
1 lambdaQuery().in(Bean::getId, idList);
```

所以需要根据业务场景处理空集合逻辑：

如果集合属性为空时不想查到任何数据

```
1 if(CollectionUtils.isEmpty(idList)){
2     // 直接返回
3     return Collections.emptyList();
4 }
5 lambdaQuery().in(Bean::getId, idList).list();
```

如果集合属性为空时忽略该条件

```
1 lambdaQuery().in(CollectionUtils.isNotEmpty(idList), Bean::getId, idList).list();
```

扩展功能

乐观锁

//使用方式 @廖高玉

逻辑删除

@廖高玉



马士兵教育
www.mashibing.com

