

mybatis-plus-offcn

1. 框架字段映射模糊点
2. 关于plus根据主键增删改问题
3. lombok注意点
4. 关于plus实体类注解解释
5. tableId的主键类型，和雪花算法的原理
6. plus中的主键回填应用场景
7. plus中的分页实现容易忽略问题

1. 框架字段映射模糊点

- mybatis-plus支持，只做增强 mybatis应该也支持

```
public class Employee implements Serializable {  
    private Long empId;  
    private String empName;  
    private String empGender;  
    private Integer age;  
    private String email;  
    public Employee() {  
    }  
    public Employee(Long empId, String empName, String empGender, Integer age, String email) {  
    }  
}
```

1 自动开启驼峰映射，表名首字母映射成大写

2 今后按照规则写

1 empId也可以成功，没区分大小写

1 plus可以，mybatis也可以只做增强

| emp_id | emp_name | emp_gender | age | email |
|--------|----------|------------|-----|-------|
|--------|----------|------------|-----|-------|

2. 关于plus根据主键增删改问题

```
@Test  
void test03() {  
    Employee employee = employeeMapper.selectById(1367686308726788098L);  
    System.out.println(employee);  
}
```

1 实体类加注解即可

1 切记按照主键查询，指定主键

- 一定要使用TableId指定主键字段

3. lombok注意点

@Data等于set get toString不包括Allargs和Noargs

4.关于plus实体类注解解释

```
//TableField作用：如果数据库表中的字段名和属性名不一致可以使用该注解去设置对应关系
@TableField("emp_id")
private Long empId;
private String empName;
private String empGender;
private Integer age;
private String email;
//去指定当前属性在数据库中不存在
@TableField(exist = false)
private String home;
```

5.tableId的主键类型，和雪花算法的原理

| 值 | 描述 |
|-------------|--|
| AUTO | 数据库 ID 自增: 前提条件， 数据库必须支持自增。 oracle 不支持自增。 |
| NONE | 无状态,该类型为未设置主键类型（主键策略如果没有设置，默认是雪花算法） I |
| INPUT | insert 前自行 set 主键值 |
| ASSIGN_ID | 分配 ID(主键类型为 Number(Long 和 Integer)或 String)(since 3.3.0), 使用接口 IdentifierGenerator 的方法 nextId(默认实现类为 DefaultIdentifierGenerator 雪花算法) 2 long类型 |
| ASSIGN_UUID | 分配 UUID, 主键类型为 String(since 3.3.0), 使用接口 IdentifierGenerator 的方法 nextUUID 1 字符串类型 |

6. plus中的主键回填应用场景

public class Employees implements Serializable { 订单表 订单明细表

1. 时间 金额

@TableField("emp_id")
@TableId(type = IdType.AUTO)

private Long empId;
private String empName;
private String empGender;
private Integer age;

生成数据关联关系的场景，创建订单成功，添加明细表，需要关联订单表，需要获取订单id 主键回填操作

```
void test01() {
    Employees employee=new Employees( empName: "张飞", empGender: "男", empAge: 30);
    employeeMapper.insert(employee);
    System.out.println("主键: "+employee.getEmpId());
}
```

7. plus中的分页实现容易忽略问题

1. 加入分页拦截器

```
@Configuration
public class MybatisPlusConfig {
    @Bean
    public MybatisPlusInterceptor mybatisPlusInterceptor() {
        MybatisPlusInterceptor interceptor = new MybatisPlusInterceptor();
        interceptor.addInnerInterceptor(new PaginationInnerInterceptor(DbType.MYSQL));
        return interceptor;
    }
}
```

1. 分页如果没有加拦截器会导致分页失效