

一、概述

1.1、目的

1.2、背景

1.3、参考资料

1.4、术语

二、整体设计

2.1、

三、功能描述

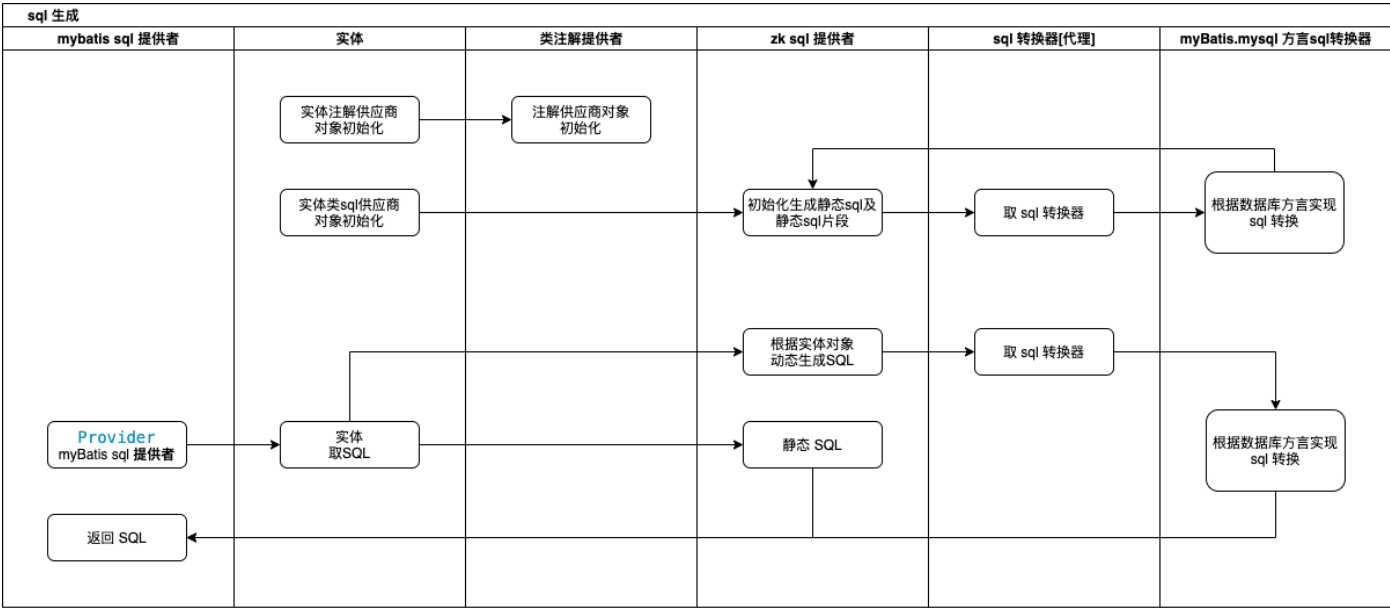
3.1 通用功能包

- 排序枚举类型
- 排序类
- json Java 数据类型及 ARJson 类型处理器

3.2 sql 生成

注解、注解提供器、查询条件、实体基础类、zk sql 提供者、sql 转换器、myBatis sql 提供者

- 整体设计思路如下图：



3.2.1 注解

- 类注解
含：表名、表别名、排序字段数组
- 字段注解

3.2.2 注解提供者

角色类的注解

3.2.3 查询条件

- 查询逻辑
com.zk.db.commons.ZKDBQueryLogic
枚举类型，定义 and 和 or 两种查询逻辑；
- 查询类型
com.zk.db.commons.ZKDBQueryType
定义常见的查询类型：=、!=、>、>=、<、<=、in not、in like、not like、is null、is not null
- 查询条件接口
com.zk.db.commons.ZKDBQueryCondition
com.zk.db.commons.ZKDBQueryConditionWhere

3.2.3.1 myBatis 查询条件实现类

- com.zk.db.mybatis.commons.ZKDBQueryConditionCol
字段列的查询条件，可生成实际的查询条件 sql；
- com.zk.db.mybatis.commons.ZKDBQueryConditionSql
字段列的查询条件，直接定义查询条件 sql

- com.zk.db.mybatis.commons.ZKDBQueryConditionIf
对 ZKDBQueryConditionCol 对象进行封装，生成查询条件时，包裹 <if> 的脚本逻辑判断
- com.zk.db.mybatis.commons.ZKDBQueryConditionWhereTrim
查询条件 ZKDBQueryConditionWhere 的实现类，会去掉两端的空格后，将查询条件转换为具体的查询 sql;

3.2.4 实体基础类

3.2.4.1 类组成

- 包含一个静态 sql 提供者

3.2.4.2 生成 sql 方法

- sql 片段: 逻辑删除 sql 片段
- 取查询条件组对象
这里是查询条件的组合，自定义的查询条件组合，未转换成 sql 的查询条件；方便自定义查询方式；
- sql 片段: 取查询结果映射 sql 片段；自定义别名

3.2.5 zk sql 提供者

- 包含一个类注解提供者对象

3.2.5.1 取表信息

- 取表名
- 取表别名
- sql 片段: 取默认排序 sql 片段

3.2.5.2 保存的静态 sql

提供者初始化时：设置类注解提供者对象；并将静态 sql 生成保存

- sql 片段: 查询目标集团表；带表别名
- sql 片段: 取默认排序sql片段；带表别名
- sql 片段: 取查询结果映射 sql 片段；带表别名
- sql 片段: 取查询结果映射 sql 片段；不带表别名
- sql 片段: 主键查询条件；带表别名
- sql 片段: 列表查询，不含排序部分；带表别名
- sql 片段: 查询条件；带表别名
- 插入 sql
- 修改 sql
- 根据主键查询 sql

- 逻辑删除 sql
- 物理删除 sql

3.2.5.3 动态生成 sql

通过 sql 转换器和实体类，实时转化

- sql 片段: 取查询结果映射；自定义别名
- sql 片段: 取查询条件；自定义别名
- sql 片段: 生成排序 sql；可自定义别名，也可指定是否需要默认排序片段
重置了三个方法，详见方法主解
- sql 片段: 取主键查询 sql；自定义别名

3.2.5.4 其他sql生成过程数据对象

- 取查询条件组对象

3.2.6 sql 转换器

接口：com.zk.db.commons.ZKSqlConvert

3.2.6.1 mysql 下 myBatis 实现：

com.zk.db.mybatis.mysql.ZKMySQLSqlConvert

这里会实现 插入、修改、查询、等 sql 转换；是转换为具体 sql 的实现类；

3.2.7 myBatis sql 提供者

myBatis dao 层的注解 sql 提供者使用，定义相关 sql 的方法，通过转换、组合、取类的静态 sql 提供需要的 完成 sql

3.3 动态数据源

3.3.1 spring

- 自定义数据源：com.zk.db.dynamic.spring.dataSource.ZKDynamicDataSource 数据源中可以定义多个数据源；如 读、写数据源；
- 自定义事务管理：com.zk.db.dynamic.spring.transaction.ZKDynamicTransactionManager
重写 org.springframework.jdbc.datasource.DataSourceTransactionManager 类 doGegin 方法，根据事务的只读或可写，设置当前事务需要使用的数据源；

3.3.2 myBatis

- 动态数据源插件：com.zk.db.dynamic.mybatis.ZKDynamicDataSourcePlugin
通过 sql 关键字判断是否需要写数据库，来设置当前使用的数据源是读数据源还是写数据源。

3.4 myBatis

3.4.1 插件

- like 参数替换【不建议使用，建议在sql 定义时，添加这个处理；】
com.zk.db.mybatis.plugins.ZKSqlLikeParameter
- 分页
com.zk.db.mybatis.plugins.ZKSqlPage
- sql 运行效率
com.zk.db.mybatis.plugins.ZKSqlRunCost
- 动态数据源插件
com.zk.db.dynamic.mybatis.ZKDynamicDataSourcePlugin

3.4.2 dao 层扫描注解

com.zk.db.annotation.ZKMyBatisDao

spring 根据这个注解来加载 dao 层 bean;

3.4.3 类型处理器

- com.zk.db.mybatis.typeHandler.ZKJsonArrayTypeHandler
- com.zk.db.mybatis.typeHandler.ZKJsonTypeHandler