

MybatisPlus拓展自定义BaseMapper

=====

前言

--

`Mybatis-plus`提供的`BaseMapper`中已经有频繁要使用的增删改查方法，比如`selectById`、`insert`等，但是有时候业务经常要用到某个模板sql，BaseMapper中又没有，MybatisPlus提供了Plugin入口，我们可以自定义BaseMapper来实现。比如，在某些table中有唯一约束键，当insert时如果唯一键冲突会抛错，如果恰好此时我们又不想处理这个错误，那我们希望使用`insert ignore into ...`的语法，但是BaseMapper没有提供这个sql的模板方法，此时需要我们去实现。

编写SQL模板

首先我们创建class `InsertIgnore`，定义方法名称和对应生成SQL的模板，这里我使用kotlin编写，大家可以转成对应的java class

...

```
import com.baomidou.mybatisplus.annotation.IdType
import com.baomidou.mybatisplus.core.injector.AbstractMethod
import com.baomidou.mybatisplus.core.metadata.TableInfo
import com.baomidou.mybatisplus.core.metadata.TableInfoHelper
import com.baomidou.mybatisplus.core.toolkit.StringPool
import com.baomidou.mybatisplus.core.toolkit.StringUtils
import com.baomidou.mybatisplus.core.toolkit.sql.SqlScriptUtils
import org.apache.ibatis.executor.keygen.Jdbc3KeyGenerator
import org.apache.ibatis.executor.keygen.KeyGenerator
import org.apache.ibatis.executor.keygen.NoKeyGenerator
import org.apache.ibatis.mapping.MappedStatement
```

```
/**
 * 实现insert ignore into sql模板
 */
class InsertIgnore : AbstractMethod() {

    companion object {
        const val METHOD_NAME = "insertIgnore"
```

```

const val SQL_TEMPLATE = """
    <script>
        INSERT IGNORE INTO %s %s VALUES %s
    </script>
    """
}

override fun injectMappedStatement(
    mapperClass: Class<*>?,
    modelClass: Class<*>?,
    tableInfo: TableInfo?
): MappedStatement {
    var keyGenerator: KeyGenerator = NoKeyGenerator()
    val columnScript = SqlScriptUtils.convertTrim(
        tableInfo!!.allInsertSqlColumnMaybelf,
        StringPool.LEFT_BRACKET, StringPool.RIGHT_BRACKET, null,
        StringPool.COMMA
    )
    val valuesScript = SqlScriptUtils.convertTrim(
        tableInfo.getAllInsertSqlPropertyMaybelf(null),
        StringPool.LEFT_BRACKET, StringPool.RIGHT_BRACKET, null,
        StringPool.COMMA
    )
    var keyProperty: String? = null
    var keyColumn: String? = null
    // 表包含主键处理逻辑,如果不包含主键当普通字段处理
    if (StringUtils.isNotBlank(tableInfo.keyProperty)) {
        if (tableInfo.idType == IdType.AUTO) {
            /** 自增主键 */
            keyGenerator = Jdbc3KeyGenerator()
            keyProperty = tableInfo.keyProperty
            keyColumn = tableInfo.keyColumn
        } else {
            if (null != tableInfo.keySequence) {
                keyGenerator =
                TableInfoHelper.genKeyGenerator(METHOD_NAME, tableInfo,
                builderAssistant)
                keyProperty = tableInfo.keyProperty
                keyColumn = tableInfo.keyColumn
            }
        }
    }
    val sql = String.format(SQL_TEMPLATE, tableInfo.tableName,
        columnScript, valuesScript)
    val sqlSource = languageDriver.createSqlSource(configuration, sql,
        modelClass)
    return addInsertMappedStatement(
        mapperClass,

```

```

        modelClass,
        METHOD_NAME,
        sqlSource,
        keyGenerator,
        keyProperty,
        keyColumn
    )
}
}
...

```

自定义Mapper

在自定义Mapper接口中添加insertIgnore方法

```

...
import com.baomidou.mybatisplus.core.mapper.BaseMapper

interface UltraBaseMapper<T> : BaseMapper<T> {

    /**
     * 插入一条数据，如果插入报错(比如唯一约束冲突) 则忽略
     */
    fun insertIgnore(entity: T): Int
}
...

```

业务Mapper原来需要继承BaseMapper，改为继承UltraBaseMapper

```

...
interface UserMapper : UltraBaseMapper<User>
...

```

这样UserMapper就拥有了insertIgnore方法，此时还缺少一项配置，那就是将我们的InsertIgnore这个class加入到MybatisPlus的插件中

配置插件

```

...
import com.baomidou.mybatisplus.core.injector.AbstractMethod
import com.baomidou.mybatisplus.core.injector.DefaultSqlInjector

class UltraSqlInjector : DefaultSqlInjector() {

    override fun getMethodList(mapperClass: Class<*>?):
MutableList<AbstractMethod> {
        val methodList = super.getMethodList(mapperClass)
        methodList.add(InsertIgnore())
        return methodList
    }
}
...

```

Spring配置

```

...
import org.springframework.context.annotation.Bean
import org.springframework.context.annotation.Configuration

@Configuration
class MybatisPlusConfig {

    @Bean
    fun ultraSqlInjector(): UltraSqlInjector {
        return UltraSqlInjector()
    }
}
...

```

总结

--

可以通过继承mybatis-plus提供的AbstractMethod来实现自定义SQL模板，比如本文中的示例InsertIgnore，我还实现了`InsertBatch`、`InsertIgnoreBatch`、`SelectByIdForUpdate`等，大家可以参考我的[Github仓库](<http://cxyroad.com/>
"https://github.com/AOKI123/mybatis-plus-ext")

原文链接: <https://juejin.cn/post/7364698043911454731>