

代码生成器配置

基本配置

dataSource

- 类型: `DataSourceConfig`
- 默认值: `null`

数据源配置，通过该配置，指定需要生成代码的具体数据库，具体请查看 [数据源配置](#)

strategy

- 类型: `StrategyConfig`
- 默认值: `null`

数据库表配置，通过该配置，可指定需要生成哪些表或者排除哪些表，具体请查看 [数据库表配置](#)

packageInfo

- 类型: `PackageConfig`
- 默认值: `null`

包名配置，通过该配置，指定生成代码的包路径，具体请查看 [包名配置](#)

template

- 类型: `TemplateConfig`
- 默认值: `null`

模板配置，可自定义代码生成的模板，实现个性化操作，具体请查看 [模板配置](#)

globalConfig

- 类型: `GlobalConfig`
- 默认值: `null`

全局策略配置，具体请查看 [全局策略配置](#)

injectionConfig

- 类型: `InjectionConfig`
- 默认值: `null`

注入配置，通过该配置，可注入自定义参数等操作以实现个性化操作，具体请查看 [注入配置](#)

数据源 `dataSourceConfig` 配置

dbQuery

- 数据库信息查询类
- 默认由 `DbType` 类型决定选择对应数据库内置实现

实现 `IDbQuery` 接口自定义数据库查询 `SQL` 语句 定制化返回自己需要的内容

DbType

- 数据库类型
- 该类内置了常用的数据库类型【必须】

schemaName

- 数据库 schema name
- 例如 `PostgreSQL` 可指定为 `public`

typeConvert

≡ MyBatis-Plus

实现 `ITypeConvert` 接口自定义数据库 字段类型 转换为自己需要的 `java` 类型，内置转换类型无法满足可实现 `IColumnType` 接口自定义

url

- 驱动连接的URL

driverName

- 驱动名称

username

- 数据库连接用户名

password

- 数据库连接密码

数据库表配置

isCapitalMode

是否大写命名

skipView

是否跳过视图

naming

数据库表映射到实体的命名策略

columnNameNaming

数据库表字段映射到实体的命名策略, 未指定按照 naming 执行

tablePrefix

表前缀

fieldPrefix

字段前缀

superEntityClass

自定义继承的Entity类全称, 带包名

superEntityColumns

自定义基础的Entity类, 公共字段

superMapperClass

自定义继承的Mapper类全称, 带包名

superServiceClass

自定义继承的Service类全称, 带包名

superServiceImplClass

自定义继承的ServiceImpl类全称, 带包名

superControllerClass

enableSqlFilter (since 3.3.1)

默认激活进行sql模糊表名匹配

关闭之后likeTable与notLikeTable将失效，include和exclude将使用内存过滤

如果有sql语法兼容性问题的话，请手动设置为false

已知无法使用：MyCat中间件，[支持情况传送门](#)🔗

include

需要包含的表名，当enableSqlFilter为false时，允许正则表达式（与exclude二选一配置）

likeTable

自3.3.0起，模糊匹配表名（与notLikeTable二选一配置）

exclude

需要排除的表名，当enableSqlFilter为false时，允许正则表达式

notLikeTable

自3.3.0起，模糊排除表名

entityColumnConstant

【实体】是否生成字段常量（默认 false）

entityBuilderModel

【实体】是否为构建者模型（默认 false），自3.3.2开始更名为 **chainModel**

chainModel (since 3.3.2)

【实体】是否为链式模型（默认 false）

entityLombokModel

【实体】是否为lombok模型（默认 false）

3.3.2以下版本默认生成了链式模型，3.3.2以后，默认不生成，如有需要，请开启 **chainModel**

entityBooleanColumnRemovesPrefix

Boolean类型字段是否移除is前缀（默认 false）

restControllerStyle

生成 `@RestController` 控制器

controllerMappingHyphenStyle

驼峰转连字符

entityTableFieldAnnotationEnable

是否生成实体时，生成字段注解

versionFieldName

乐观锁属性名称

logicDeleteFieldName

逻辑删除属性名称

tableFillList

表填充字段

包名配置

parent

父包名。如果为空，将下面子包名必须写全部，否则就只需写子包名

moduleName

父包模块名

entity

Entity包名

service

Service包名

serviceImpl

Service Impl包名

mapper

Mapper包名

xml

Mapper XML包名

controller

Controller包名

pathInfo

路径配置信息

模板配置

entity

Java 实体类模板

entityKt

Kotin 实体类模板

service

Service 类模板

serviceImpl

Service impl 实现类模板

mapper

mapper 模板

xml

mapper xml 模板

controller

controller 控制器模板

全局策略 globalConfig 配置

outputDir

- 生成文件的输出目录
- 默认值: `D` 盘根目录

fileOverride

- 是否覆盖已有文件
- 默认值: `false`

open

- 是否打开输出目录
- 默认值: `true`

enableCache

- 是否在xml中添加二级缓存配置
- 默认值: `false`

author

- 开发人员
- 默认值: `null`

kotlin

swagger2

- 开启 swagger2 模式
- 默认值: `false`

activeRecord

- 开启 ActiveRecord 模式
- 默认值: `false`

baseResultMap

- 开启 BaseResultMap
- 默认值: `false`

baseColumnList

- 开启 baseColumnList
- 默认值: `false`

dateType

- 时间类型对应策略
- 默认值: `TIME_PACK`

注意事项:

如下配置 `%s` 为占位符

entityName

- 实体命名方式

mapperName

- mapper 命名方式
- 默认值: `null` 例如: `%sDao` 生成 `UserDao`

xmlName

- Mapper xml 命名方式
- 默认值: `null` 例如: `%sDao` 生成 `UserDao.xml`

serviceName

- service 命名方式
- 默认值: `null` 例如: `%sBusiness` 生成 `UserBusiness`

serviceImplName

- service impl 命名方式
- 默认值: `null` 例如: `%sBusinessImpl` 生成 `UserBusinessImpl`

controllerName

- controller 命名方式
- 默认值: `null` 例如: `%sAction` 生成 `UserAction`

idType

- 指定生成的主键的ID类型
- 默认值: `null`

注入 injectionConfig 配置

map

fileOutConfigList

- 自定义输出文件
- 配置 `FileOutConfig` 指定模板文件、输出文件达到自定义文件生成目的

fileCreate

- 自定义判断是否创建文件
- 实现 `IFileCreate` 接口

该配置用于判断某个类是否需要覆盖创建，当然你可以自己实现差异算法 `merge` 文件

initMap

- 注入自定义 Map 对象(注意需要setMap放进去)

在 [GitHub](#) 上编辑此页 [↗](#)

上次更新: 2020/10/7 下午5:41:16

[← 使用配置](#)

赞助商

