

mybatis 代码生成器 (IDEA, Maven)及配置详解 (部分配置你应该不知道)

在使用 mybatis 过程中，当手写 **JavaBean**和**XML** 写的越来越多的时候，就越来越同意出错。这种重复性的工作，我们当然不希望做那么多。

还好，mybatis 为我们提供了强大的代码生成--**MybatisGenerator**。

通过简单的配置，我们就可以生成各种类型的实体类，Mapper接口，MapperXML文件，Example对象等。通过这些生成的文件，我们就可以方便的进行单表进行增删改查的操作。

以下的工具使用的都是 **IDEA**

1 创建代码生成器

1.1 创建Maven项目

1.1.1 菜单上选择新建项目

File | New | Project

昵称：阿进的写字台
园龄：1年6个月
粉丝：47
关注：4
[+加关注](#)

2018年12月						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

找找看

谷歌搜索

常用链接

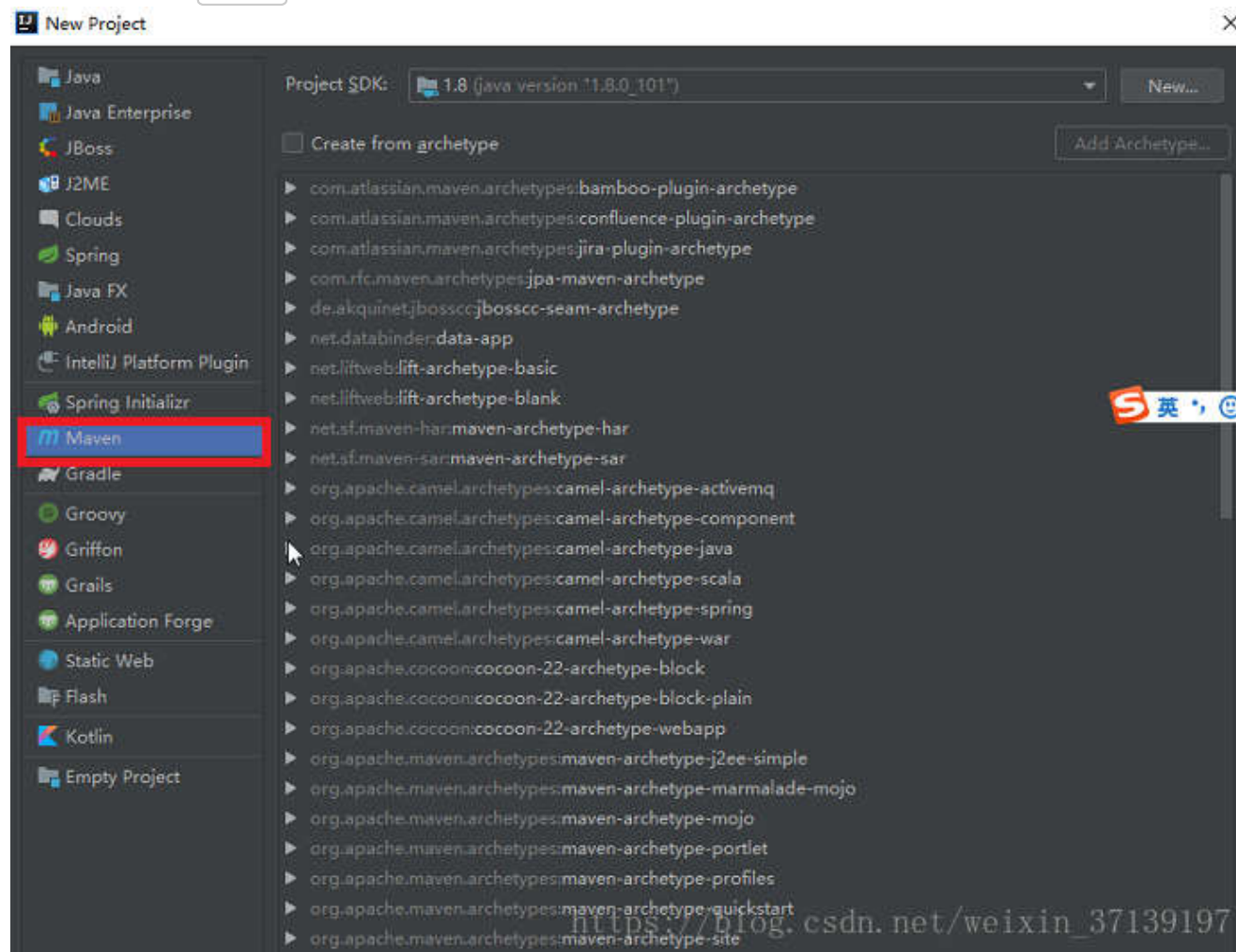
[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

1.1.2 选择左侧的 Maven



由于我们只是创建一个普通的项目，此处点击 Next 即可。

1.1.3 输入 GroupId 和 ArtifactId

- 在我的项目中，

GroupId 填 com.homejim.mybatis

ArtifactId 填 mybatis-generator

我的标签

我的标签

Java(26)

mybatis(17)

mybatis使用(7)

mybatis源码(6)

iot(6)

mqtt(6)

paho(6)

Thread(5)

多线程(4)

反射(3)

更多

随笔分类

C/C++(1)

Clean Code

Java(8)

Java 进阶(15)

kafka(1)

linux(1)

mqtt(6)

mybatis-工具(1)

点击 [Next](#) 。

1.1.4 [Finish](#)

通过以上步骤，一个普通的 [Maven](#) 项目就创建好了。

1.2 配置 generator.xml

其实名字无所谓，只要跟下面的 **pom.xml** 文件中的对应上就好了。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE generatorConfiguration PUBLIC
    "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
    "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd" >
<generatorConfiguration>

    <!-- 本地数据库驱动程序jar包的全路径 -->
    <classPathEntry location="C:\Users\Administrator\.m2\repository\mysql\mysql-connector-
java\8.0.12\mysql-connector-java-8.0.12.jar"/>
    <context id="context" targetRuntime="MyBatis3">
        <commentGenerator>
            <property name="suppressAllComments" value="false"/>
            <property name="suppressDate" value="true"/>
        </commentGenerator>

        <!-- 数据库的相关配置 -->
        <jdbcConnection>
            <property name="driverClass" value="com.mysql.jdbc.Driver"/>
            <property name="connectionURL" value="jdbc:mysql://localhost:3306/mybatis"/>
            <property name="userId" value="root"/>
            <property name="password" value="jim777"/>
        </jdbcConnection>

        <jdbcTypeResolver>
            <property name="forceBigDecimals" value="false"/>
        </jdbcTypeResolver>

        <!-- 实体类生成的位置 -->
        <javaModelGenerator>
```

[mybatis-使用\(8\)](#)[mybatis-源码\(10\)](#)[paho\(5\)](#)[spring-boot使用](#)[tomcat\(1\)](#)[多线程那些事\(1\)](#)[集合框架那些事\(1\)](#)

随笔档案

[2018年12月 \(5\)](#)[2018年11月 \(5\)](#)[2018年10月 \(11\)](#)[2018年9月 \(8\)](#)[2018年8月 \(5\)](#)[2018年3月 \(3\)](#)[2018年1月 \(3\)](#)[2017年12月 \(9\)](#)

linux

积分与排名

积分 - 29546

排名 - 17320

最新评论

```
        targetPackage="com.homejim.mybatis.entity"
        targetProject=".\\src\\main\\java">
        <property name="enableSubPackages" value="false"/>
        <property name="trimStrings" value="true"/>
    </javaModelGenerator>

    <!-- *Mapper.xml 文件的位置 sqlMapGenerator-->
    <sqlMapGenerator
        targetPackage="mybatis/mapper"
        targetProject=".\\src\\main\\resources">
        <property name="enableSubPackages" value="false"/>
    </sqlMapGenerator>

    <!-- Mapper 接口文件的位置 -->
    <javaClientGenerator type="XMLMAPPER"
        targetPackage="com.homejim.mybatis.mapper"
        targetProject=".\\src\\main\\java">
        <property name="enableSubPackages" value="false"/>
    </javaClientGenerator>

    <!-- 相关表的配置 -->

    <table tableName="blog" />
</context>
</generatorConfiguration>
```

需要改一些内容:

1. 本地数据库驱动程序jar包的全路径 (**必须要改**)。
2. 数据库的相关配置 (**必须要改**)
3. 相关表的配置 (**必须要改**)
4. 实体类生成存放的位置。
5. MapperXML 生成文件存放的位置。
6. Mapper 接口存放的位置。

如果不知道怎么改, 请看后面的**配置详解**。

1. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

@灯塔下的守望者得到肯定, 很开心, 谢谢...

--阿进的写字台

2. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

@Java深入浅出这个判断很简单的。if ((name.startsWith("get") && name.length() > 3) || (name.startsW.....

--阿进的写字台

3. Re:MyBatis动态SQL (认真看看, 以后写SQL就爽多了)

支持, 非常好

--灯塔下的守望者

4. Re:mybatis源码- 反射模块一 (跟着MyBatis学反射): 类级别信息的封装

怎么没写怎么判断 setter和getter?

--Java深入浅出

5. Re:HashMap是如何工作的

mark

--Java深入浅出

阅读排行榜

1. Paho -物联网 MQTT C Client的实现和详解(9888)

1.3 配置 pom.xml

在原基础上添加一些内容。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.homejim.mybatis</groupId>
    <artifactId>mybatis-generator</artifactId>
    <version>1.0-SNAPSHOT</version>
    <!--在原基础上添加 这些就好了-->
    <build>
        <finalName>mybatis-generator</finalName>
        <plugins>
            <plugin>
                <groupId>org.mybatis.generator</groupId>
                <artifactId>mybatis-generator-maven-plugin</artifactId>
                <version>1.3.7</version>
                <configuration>
                    <!--此处需要注意, 文件与上面的文件匹配-->
                    <configurationFile>src/main/resources/generator.xml</configurationFile>
                    <verbose>true</verbose>
                    <overwrite>true</overwrite>
                </configuration>
                <executions>
                    <execution>
                        <id>Generate MyBatis Artifacts</id>
                        <goals>
                            <goal>generate</goal>
                        </goals>
                    </execution>
                </executions>
                <dependencies>
                    <dependency>
                        <groupId>org.mybatis.generator</groupId>
```

2. 运行时动态库: not found 及介绍-linux的-Wl,-rpath命令(3506)

3. JDBC详解系列 (一) 之流程(3348)

4. Paho - MQTT C Client的实现(3031)

5. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (2198)

评论排行榜

1. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (14)

2. mybatis 多个接口参数的注解使用方式 (@Param)(7)

3. mybatis源码-解析配置文件 (二) 之解析的流程(3)

4. mybatis源码-原来resultMap解析完是这样(3)

5. Java设计模式-建造者(Builder)模式(2)

推荐排行榜

1. MyBatis动态SQL (认真看看, 以后写SQL就爽多了) (15)

2. HashMap是如何工作的(5)

3. 扒一扒: Java 中的枚举(4)

4. mybatis源码-原来resultMap解析完是这样(4)

5. mybatis源码-Mapper解析之SQL 语句节点解析 (一条语句对应一个MappedStatement)(2)

```
        <artifactId>mybatis-generator-core</artifactId>
        <version>1.3.7</version>
    </dependency>
</dependencies>
</plugin>
</plugins>
</build>
<!--到此结束-->
</project>
```

需要注意的是 **configurationFile** 中的文件指的是 **generator.xml**。因此路径写的是该文件的相对路径，名称也跟该文件相同。

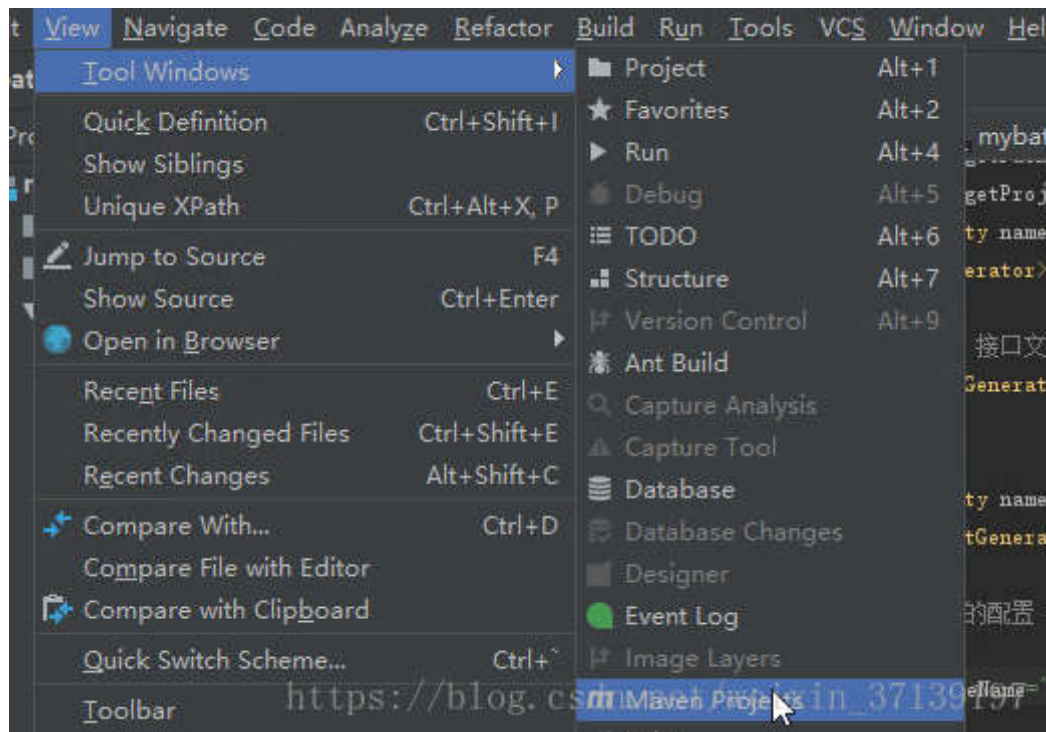
到此，**mybatis-generator** 就可以使用啦。

1.4 使用及测试

1.4.1 打开 Maven Projects 视图

在 IDEA 上，打开：

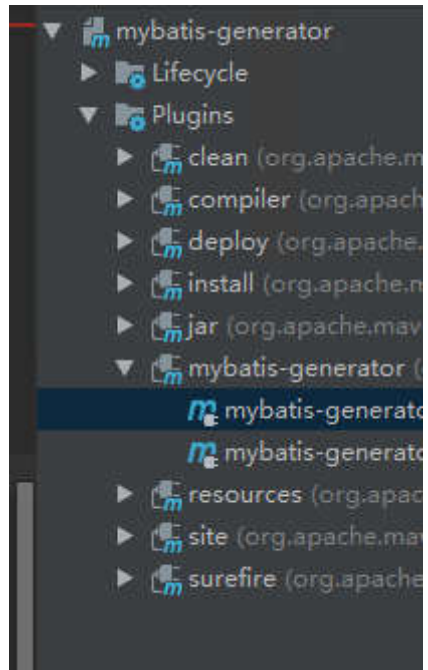
View | Tools | Windwos | Maven Projects



1.4.2 Maven Projects 中双击 mybatis-generator

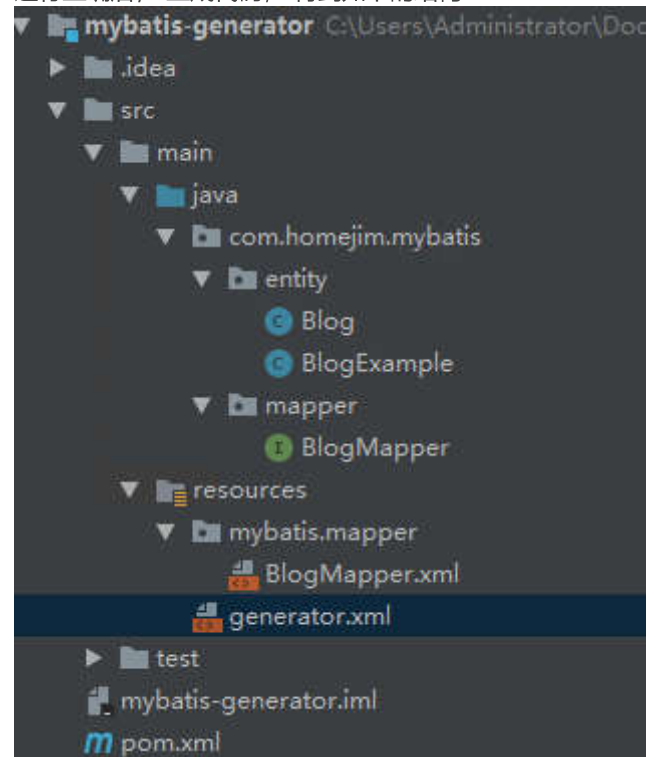
在右侧此时可以看到 Maven Projects 了。找到 mybatis-generator 插件。

mybatis-generator | Plugins | mybatis-generator | mybatis-generator



1.4.3 双击运行

运行正确后, 生成代码, 得到如下的结构



2 XML 配置详解

仅仅是上面那么简单的使用还不够爽。那么我们就可以通过更改 generator.xml 配置文件的方式进行生成的配置。

2.1 优先

推荐查看官方的文档。

英文不错的: [官方网站](#)。

中文翻译版: [翻译版网站](#)

2.2 官网没有的

2.2.1 property 标签

该标签在官网中只是说用来指定元素的属性, 至于怎么用没有详细的讲解。

2.2.1.1 分隔符相关

```
<property name="autoDelimitKeywords" value="true"/>
<property name="beginningDelimiter" value="`"/>
<property name="endingDelimiter" value="`"/>
```

以上的配置对应的是 **mysql**，当数据库中的字段和数据库的关键字一样时，就会使用分隔符。

比如我们的数据列是 `delete`，按以上的配置后，在它出现的地方，就变成 ``delete``。

2.2.1.2 编码

默认是使用当前的系统环境的编码，可以配置为 GBK 或 UTF-8。

```
<property name="javaFileEncoding" value="UTF-8"/>
```

我想项目为 UTF-8，如果指定生成 GBK，则自动生成的中文就是乱码。

2.2.1.3 格式化

```
<!--格式化生成的 Java 代码-->
<property name="javaFormatter" value="org.mybatis.generator.api.dom.DefaultJavaFormatter"/>
<!--格式化生成的 XML-->
<property name="xmlFormatter" value="org.mybatis.generator.api.dom.DefaultXmlFormatter"/>
```

这些显然都是可以自定义实现的。

2.2.2 plugins 标签

plugins 标签用来扩展或修改代码生成器生成的代码。

在生成的 XML 中，是没有 **<cache>** 这个标签的。该标签是配置缓存的。

如果我们想生成这个标签，那么可以在 **plugins** 中进行配置。

```
<plugin type="org.mybatis.generator.plugins.CachePlugin" >
    <property name="cache_eviction" value="LRU"/>
</plugin>
```

```
<cache eviction="LRU">
  <!--
    WARNING - @mbg.generated
    This element is automatically generated by MyBatis Generator. do not modify.
  -->
</cache>
```

比如你想生成的 **JavaBean** 中自行实现 `Serializable` 接口。

```
<plugin type="org.mybatis.generator.plugins.SerializablePlugin" />
```

```
public class Student implements Serializable {
  /**
   * This field was generated by MyBatis Generator.
   * This field corresponds to the database column student.student_id
   */
}
```

还能自定义插件。

这些插件都蛮有用的，感觉后续可以专门开一篇文章来讲解。

2.2.3 commentGenerator 标签

看名称，就知道是用来生成注释用的。

默认配置：

```
<commentGenerator >
  <property name="suppressAllComments" value="false"/>
  <property name="suppressDate" value="false"/>
  <property name="addRemarkComments" value="false"/>
</commentGenerator>
```

`suppressAllComments`: 阻止生成注释，默认值是`false`。

`suppressDate`: 阻止生成的注释包含时间戳，默认为`false`。

`addRemarkComments`: 注释中添加数据库的注释，默认为 `false`。

还有一个就是我们可以通过 **type** 属性指定我们自定义的注解实现类，生成我们自己想要的注解。自定义的实现类需要实现

```
org.mybatis.generator.api.CommentGenerator
```

2.2.4 待续

待添加。

Github

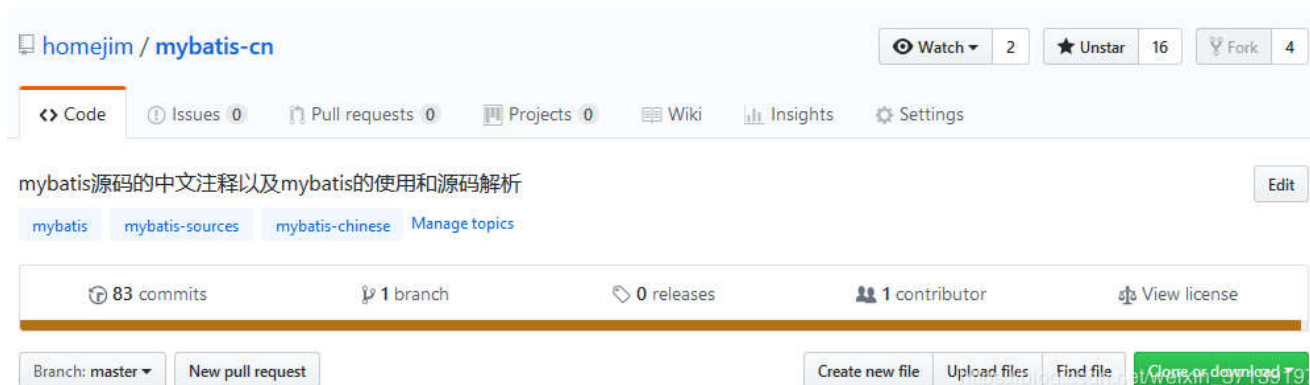
mybatis-examples 项目中的 [mybatis-generator](#) 即本文使用到的所有代码。

一起学 mybatis

你不想来学习 mybatis? 学习其使用和源码呢? 那么，在博客园关注我吧!!

我自己打算把这个源码系列更新完毕，同时会更新相应的注释。快去 star 吧!!

[mybatis最新源码和注释](#)



作者: [阿进的写字台](#)

出处: <https://www.cnblogs.com/homejim/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

好文要顶

关注我

收藏该文



[阿进的写字台](#)[关注 - 4](#)[粉丝 - 47](#)

1

0

[+加关注](#)[« 上一篇: mybatis 多个接口参数的注解使用方式 \(@Param\)](#)[» 下一篇: mybatis-高级结果映射之一对一](#)

posted @ 2018-10-13 12:00 阿进的写字台 阅读(422) 评论(2) 编辑 收藏

#1楼 2018-10-14 17:02 阿苍老师

“ 请问生成后都多了一个Example是干什么用的? ?

支持(0) 反对(0)”

#2楼[楼主] 2018-10-14 17:28 阿进的写字台

“ @ 阿苍老师

Example类用来处理MBG动态查询功能, 你可以看看selectByExample, deleteByExample等方法的XML。

具体介绍可以看这个: <http://mbg.cndocs.ml/generatedobjects/exampleClassUsage.html>。

你先看, 没明白我可以给你写一个示例

支持(0) 反对(0)”

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【活动】华为云12.12会员节 云产品1折起 满额送Mate20 点击抢购

【推荐】服务器100%基准CPU性能, 1核1G首年168元, 限时特惠!



相关博文:

- [mybatis 代码生成器配置](#)
- [读书笔记-你不知道的JS中-函数生成器](#)
- [你不知道的Spring配置文件](#)
- [mybatis代码生成器配置文件详解](#)
- [我知道你不知道我知道你不知道我知道你不知道](#)

最新新闻:

- [戴威的董事会股东的签字权 谁搞垮了ofo的资金和前途?](#)
 - [携程有忧, 建章难退](#)
 - [史玉柱重出江湖三年: 属于他的时代, 远去了](#)
 - [苹果回应在德国被禁售: 对判决结果很失望 将上诉](#)
 - [Facebook隐私问题再度发酵: 议员呼吁国会介入](#)
- » [更多新闻...](#)

Copyright ©2018 阿进的写字台