1. 在maven中添加mybatis-generator插件

前提：需要导入mysql-connector-java；mybatis-spring-boot-starter（我用的是springboot）

插件：

<plugin>

<groupId>org.mybatis.generator</groupId>

<artifactId>mybatis-generator-maven-plugin</artifactId>

<version>1.3.5</version>

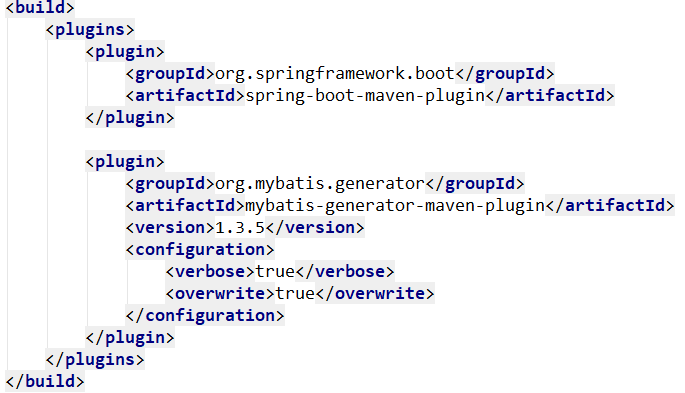
<configuration>

<verbose>true</verbose>

<overwrite>true</overwrite>

</configuration>

</plugin>



2. 配置数据属性：新建generatorConfig.properties

jdbc.driverLocation=D:/javatool/mavenLib/mysql/mysql-connector-java/8.0.13/mysql-connector-java-8.0.13.jar

jdbc.driverClass=com.mysql.cj.jdbc.Driver

jdbc.connectionURL=jdbc:mysql://localhost:3306/test?serverTimezone=UTC&characterEncoding=utf8

jdbc.username=root

jdbc.password=root

#dao接口地址

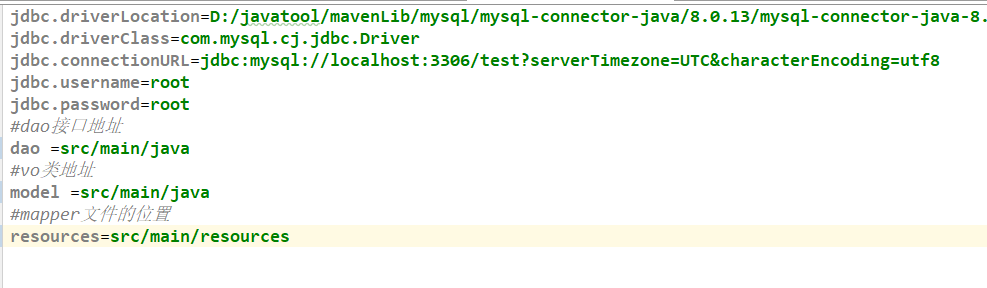
dao =src/main/java

#vo类地址

model =src/main/java

#mapper文件的位置

resources=src/main/resources



3. 编写配置文件：generatorConfig.xml



<?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">

<!-- 配置生成器 -->

<generatorConfiguration>

<!--执行generator插件生成文件的命令： call mvn mybatis-generator:generate -e -->

<!-- 引入配置文件 -->

<properties resource="generatorConfig.properties"/>

<!--classPathEntry:数据库的JDBC驱动,换成你自己的驱动位置 可选 -->

<classPathEntry location="${jdbc.driverLocation}" />

<!-- 一个数据库一个context -->

<!--defaultModelType="flat" 大数据字段，不分表 -->

<context id="MysqlTables" targetRuntime="MyBatis3Simple" defaultModelType="flat">

<!-- 自动识别数据库关键字，默认false，如果设置为true，根据SqlReservedWords中定义的关键字列表；

一般保留默认值，遇到数据库关键字（Java关键字），使用columnOverride覆盖 -->

<property name="autoDelimitKeywords" value="true" />

<!-- 生成的Java文件的编码 -->

<property name="javaFileEncoding" value="utf-8" />

<!-- beginningDelimiter和endingDelimiter：指明数据库的用于标记数据库对象名的符号，比如ORACLE就是双引号，MYSQL默认是`反引号； -->

<property name="beginningDelimiter" value="`" />

<property name="endingDelimiter" value="`" />

<!-- 格式化java代码 -->

<property name="javaFormatter" value="org.mybatis.generator.api.dom.DefaultJavaFormatter"/>

<!-- 格式化XML代码 -->

<property name="xmlFormatter" value="org.mybatis.generator.api.dom.DefaultXmlFormatter"/>

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

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

<!-- 注释 -->

<commentGenerator >

<property name="suppressAllComments" value="true"/><!-- 是否取消注释 -->

<property name="suppressDate" value="false" /> <!-- 是否生成注释代时间戳-->

</commentGenerator>

<!-- jdbc连接 -->

<jdbcConnection

driverClass="${jdbc.driverClass}"

connectionURL="${jdbc.connectionURL}"

userId="${jdbc.username}"

password="${jdbc.password}"/>

<!-- 类型转换 -->

<javaTypeResolver>

<!-- 是否使用bigDecimal， false可自动转化以下类型（Long, Integer, Short, etc.） -->

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

</javaTypeResolver>

<!-- 生成实体类地址 -->

<javaModelGenerator targetPackage="cn.commodity.os.vo" targetProject="${model}" >

<property name="enableSubPackages" value="false"/>

<property name="trimStrings" value="true"/>

</javaModelGenerator>

<!-- 生成mapxml文件 -->

<sqlMapGenerator targetPackage="mapper" targetProject="${resources}" >

<property name="enableSubPackages" value="false" />

</sqlMapGenerator>

<!-- 生成mapxml对应client，也就是接口dao -->

<javaClientGenerator targetPackage="cn.commodity.os.dao" targetProject="${dao}" type="XMLMAPPER" >

<property name="enableSubPackages" value="false" />

</javaClientGenerator>

<!-- table可以有多个,每个数据库中的表都可以写一个table，tableName表示要匹配的数据库表,也可以在tableName属性中通过使用%通配符来匹配所有数据库表,只有匹配的表才会自动生成文件 -->

<table tableName="commodity" enableCountByExample="true" enableUpdateByExample="true" enableDeleteByExample="true" enableSelectByExample="true" selectByExampleQueryId="true">

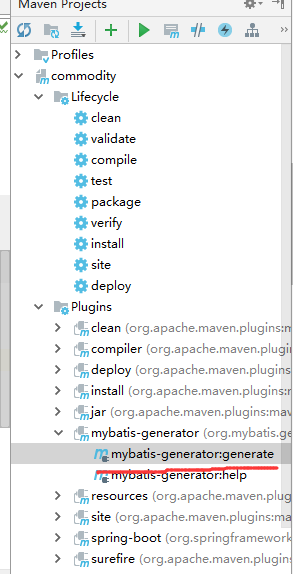
<!-- 数据库表主键 -->

<generatedKey column="id" sqlStatement="Mysql" identity="true" />

</table>

</context>

</generatorConfiguration>4. 点击插件生成代码



5. 自动生成的代码

