新建一个maven项目名：mybatis-generator

第一种形式：启动类逆向工程的形式

1、启动类：

```java
package com.sinosoft.cms.api;

import org.mybatis.generator.api.MyBatisGenerator;
import org.mybatis.generator.config.Configuration;
import org.mybatis.generator.config.xml.ConfigurationParser;
import org.mybatis.generator.internal.DefaultShellCallback;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

/**
 * @program: sale-server
 * @description:
 * @author: MR.wjw
 * @create:
 */
public class MybatisGeneratorApp {
 public static void main(String[] args) throws Exception {
 try {
 MybatisGeneratorApp generatorSqlmap = new MybatisGeneratorApp();
 generatorSqlmap.generator();
 } catch (Exception e) {
 e.printStackTrace();
 }

 }

 public void generator() throws Exception {

 List<String> warnings = new ArrayList<String>();
 boolean overwrite = true;
 //指定 逆向工程配置文件全路径
 File configFile = new File("D:\\GitLocal\\mybatis-generator\\src\\main\\resources\\mybatis-generator.xml");
 ConfigurationParser cp = new ConfigurationParser(warnings);
 Configuration config = cp.parseConfiguration(configFile);
 DefaultShellCallback callback = new DefaultShellCallback(overwrite);
 MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,
 callback, warnings);
 myBatisGenerator.generate(null);
 System.out.println("==========================");

 }
}
```

2、配置文件：mybatis-generator.xml 这是MySQL逆向工程

```
<?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>
 <context id="mysql" targetRuntime="MyBatis3">
 <!-- 数据库链接信息 -->
 <jdbcConnection driverClass="com.mysql.jdbc.Driver"
  connectionURL="jdbc:mysql://localhost:3306/mysql"
  userId="root"
  password="root">
  </jdbcConnection>
  <javaTypeResolver >
  <!-- 不需要将数据的小数类型，生成为BigDecimal -->
  <property name="forceBigDecimals" value="false" />
  </javaTypeResolver>
  <!-- 生成数据库表对应的pojo类 -->
  <javaModelGenerator targetPackage="com.sinosoft.cms.dao"
  targetProject="src/main/java">
  <property name="enableSubPackages" value="false" />
  <property name="trimStrings" value="true" />
  </javaModelGenerator>
  <!-- 生成mapper -->
  <sqlMapGenerator targetPackage="com.sinosoft.cms.po"
  targetProject="src/main/java">
  <property name="enableSubPackages" value="false" />
  </sqlMapGenerator>
  <!-- 生成mapper.xml -->
  <javaClientGenerator type="XMLMAPPER"
  targetPackage="com.sinosoft.cms.dao"
  targetProject="src/main/java">
  <property name="enableSubPackages" value="false" />
  </javaClientGenerator>
  <!-- 指定数据库的表 与数据库表字段一一对应-->
  <table schema="" tableName="student"
  domainObjectName="StudentPo" mapperName="StudentDao"></table>

  </context>
 </generatorConfiguration>
```

3、依赖：

```
<dependencies>
 <!-- mybatis-generator逆向工程 -->
 <dependency>
 <groupId>org.mybatis.generator</groupId>
 <version>1.3.5</version>
 <artifactId>mybatis-generator-core</artifactId>
 </dependency>
 <!-- mysql驱动 -->
 <dependency>
 <groupId>mysql</groupId>
 <artifactId>mysql-connector-java</artifactId>
 <version>5.1.47</version>
 </dependency>
 <!-- mybatis -->
 <dependency>
 <groupId>org.mybatis</groupId>
 <artifactId>mybatis</artifactId>
 <version>3.4.6</version>
 </dependency>
</dependencies>
```

1、启动类(跟上面一样)

```
package com.sinosoft.cms.api;

import org.mybatis.generator.api.MyBatisGenerator;
import org.mybatis.generator.config.Configuration;
import org.mybatis.generator.config.xml.ConfigurationParser;
import org.mybatis.generator.internal.DefaultShellCallback;

import java.io.File;
import java.util.ArrayList;
 import java.util.List;

 /**
  * @program: sale-server
  * @description:
  * @author: MR.wjw
  * @create:
  */
 public class MybatisGeneratorApp {
  public static void main(String[] args) throws Exception {
  try {
  MybatisGeneratorApp generatorSqlmap = new MybatisGeneratorApp();
  generatorSqlmap.generator();
  } catch (Exception e) {
  e.printStackTrace();
  }

  }

  public void generator() throws Exception {

  List<String> warnings = new ArrayList<String>();
  boolean overwrite = true;
  //指定 逆向工程配置文件全路径
  File configFile = new File("D:\\GitLocal\\mybatis-generator\\src\\main\\resources\\mybatis-generator.xml");
  ConfigurationParser cp = new ConfigurationParser(warnings);
  Configuration config = cp.parseConfiguration(configFile);
  DefaultShellCallback callback = new DefaultShellCallback(overwrite);
  MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,
  callback, warnings);
  myBatisGenerator.generate(null);
  System.out.println("==========================");

  }
 }
```

2、配置文件：mybatis-generator.xml 这是PG数据库逆向工程

```
<?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>
 <context id="DB2Tables" defaultModelType="flat" targetRuntime="MyBatis3">
 <!-- 生成的Java文件的编码 -->
 <property name="javaFileEncoding" value="UTF-8"/>
 <!-- 格式化java代码 -->
  <property name="javaFormatter" value="org.mybatis.generator.api.dom.DefaultJavaFormatter"/>
  <!-- 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="`"/>
  <!-- JavaBean 实现 序列化 接口 -->
  <plugin type="org.mybatis.generator.plugins.SerializablePlugin"/>

  <!-- generat entity时,生成toString -->
  <plugin type="org.mybatis.generator.plugins.ToStringPlugin"/>
  <!-- 此处是将Example后缀改名为Selective 当然 想改成什么都行~ -->
  <plugin type="org.mybatis.generator.plugins.RenameExampleClassPlugin">
  <property name="searchString" value="Example$"/>
  <property name="replaceString" value="Selective"/>
  </plugin>


  <commentGenerator type="org.mybatis.generator.internal.DefaultCommentGenerator">
  <!-- 是否去除自动生成的注释 true：是 ： false:否 -->
  <property name="suppressAllComments" value="true"/>
  </commentGenerator>

  <!--数据库连接的信息：驱动类、连接地址、用户名、密码 -->
  <jdbcConnection driverClass="org.postgresql.Driver"
  connectionURL="jdbc:postgresql://10.252.68.222:5432/xgpgdev" userId="picc"
```

```
password="PICC@xgwfw">
<property name="nullCatalogMeansCurrent" value="true"/>
</jdbcConnection>
<!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer，为 true时把JDBC DECIMAL 和
NUMERIC 类型解析为java.math.BigDecimal -->
<javaTypeResolver>
<property name="forceBigDecimals" value="true"/>
</javaTypeResolver>

<!-- targetProject:生成PO类的位置 -->
<javaModelGenerator targetPackage="com.sinosoft.cms.po.base"
targetProject="src/main/java">
<!-- 在targetPackage的基础上，根据数据库的schema再生成一层package，最终生成的类放在这个package下，默认为false -->
<property name="enableSubPackages" value="true"/>
<!-- 从数据库返回的值被清理前后的空格 -->
<property name="trimStrings" value="true"/>
</javaModelGenerator>
<!-- targetProject:mapper映射文件生成的位置 -->
<sqlMapGenerator targetPackage="base"
targetProject="src/main/resources/mapping">
<!-- enableSubPackages:是否让schema作为包的后缀 -->
<property name="enableSubPackages" value="false"/>
</sqlMapGenerator>

<!-- targetPackage：mapper接口生成的位置 -->
<javaClientGenerator type="XMLMAPPER" targetPackage="com.sinosoft.cms.dao.base"
targetProject="src/main/java">
<!-- enableSubPackages:是否让schema作为包的后缀 -->
<property name="enableSubPackages" value="true"/>
</javaClientGenerator>

<!-- 指定数据库表 schema不用填写 tableName=“表名” domainObjectName=生成的PO的名字
mapperName=生成的mapper的名字-->
<!-- <table schema="" tableName="WECHAT_MESSAGE"/> -->

<table schema="" tableName="laassessnotquarter"
domainObjectName="LAAssessNotQuarterPo" mapperName="LAAssessNotQuarterDao" ></table>
```

```
 <!-- 指定数据库所有表 -->
 <!-- <table schema="" tableName="%"/> -->
 <!-- 有些表的字段需要指定java类型
 <table schema="" tableName="">
 <columnOverride column="" javaType="" />
 </table> -->
 </context>
</generatorConfiguration>
```

3、依赖：

```
<dependencies>
 <!-- mybatis-generator逆向工程 -->
 <dependency>
 <groupId>org.mybatis.generator</groupId>
 <version>1.3.5</version>
 <artifactId>mybatis-generator-core</artifactId>
 </dependency>
 <!--连接PG库的驱动-->
 <dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.2.5</version>
  </dependency>
  <!-- mybatis -->
  <dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.4.6</version>
  </dependency>
</dependencies>
```

第二种形式：插件逆向工程的形式

这里只列举PG库逆向工程的pom.xml中的配置方法(不需要启动类)

1、mybatis-generator.xml文件和上面的一样

2、配置pom.xml，配置完在右侧的maven中点击Plugins——>mybatis-generator——>mybatis-generator:generate即可。

```
<build>
 <finalName>${project.artifactId}</finalName>
 <plugins>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-surefire-plugin</artifactId>
 <version>2.9</version>
 <configuration>
 <includes>
 <include>**/SwaggerTest.java</include>
 </includes>
 </configuration>
 </plugin>

 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-surefire-plugin</artifactId>
 <configuration>
 <includes>
 <include>**/SwaggerTest.java</include>
 </includes>
 <systemPropertyVariables>
 <io.springfox.staticdocs.outputDir>${project.build.directory}/swagger
 </io.springfox.staticdocs.outputDir>
 <io.swagger.json.uris>/v2/api-docs?group=UI,/v2/api-docs?group=API</io.swagger.json.uris>
 <io.swagger.json.output.name>acrc-ui-v1.json,acrc-api-v1.json</io.swagger.json.output.name>
 </systemPropertyVariables>
 </configuration>
 </plugin>


 <!--Mybatis代码自动生成器-->
 <plugin>
 <groupId>org.mybatis.generator</groupId>
 <artifactId>mybatis-generator-maven-plugin</artifactId>
 <version>1.3.5</version>
 <dependencies>
 <dependency>
 <groupId>mysql</groupId>
 <artifactId>mysql-connector-java</artifactId>
 <version>5.1.39</version>
 </dependency>
 <dependency>
 <groupId>org.mybatis.generator</groupId>
 <artifactId>mybatis-generator-core</artifactId>
 <version>1.3.5</version>
 </dependency>
 <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
 <dependency>
 <groupId>org.postgresql</groupId>
 <artifactId>postgresql</artifactId>
 <version>42.2.5</version>
 </dependency>


 </dependencies>
 <executions>
 <execution>
 <id>Generate MyBatis Artifacts</id>
 <phase>package</phase>
 <goals>
 <goal>generate</goal>
 </goals>
 </execution>
 </executions>
 <configuration>
 <!--允许移动生成的文件 -->
 <verbose>true</verbose>
 <!-- 是否覆盖 -->
 <overwrite>true</overwrite>
 <!-- 自动生成的配置，指定generator文件的地址-->
 <configurationFile>src/main/resources/mybatis-generator.xml</configurationFile>
 </configuration>
 </plugin>

 </plugins>
</build>
```