

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：

```
1 package com.sinosoft.cms.api;
2
3 import org.mybatis.generator.api.MyBatisGenerator;
4 import org.mybatis.generator.config.Configuration;
5 import org.mybatis.generator.config.xml.ConfigurationParser;
6 import org.mybatis.generator.internal.DefaultShellCallback;
7
8 import java.io.File;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 /**
13  * @program: sale-server
14  * @description:
15  * @author: MR.wjw
16  * @create:
17  */
18 public class MybatisGeneratorApp {
19     public static void main(String[] args) throws Exception {
20         try {
21             MybatisGeneratorApp generatorSqlmap = new MybatisGeneratorApp();
22             generatorSqlmap.generator();
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26     }
27 }
28
29 public void generator() throws Exception {
30
31     List<String> warnings = new ArrayList<String>();
32     boolean overwrite = true;
33     //指定 逆向工程配置文件全路径
34     File configFile = new File("D:\\GitLocal\\mybatis-generator\\src\\main\\resources\\mybatis-generator.xml");
35     ConfigurationParser cp = new ConfigurationParser(warnings);
36     Configuration config = cp.parseConfiguration(configFile);
```

```

37  DefaultShellCallback callback = new DefaultShellCallback(overwrite);
38  MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,
39  callback, warnings);
40  myBatisGenerator.generate(null);
41  System.out.println("=====");
42
43  }
44  }

```

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

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE generatorConfiguration
3  PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4  "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
5
6  <generatorConfiguration>
7  <context id="mysql" targetRuntime="MyBatis3">
8  <!-- 数据库链接信息 -->
9  <jdbcConnection driverClass="com.mysql.jdbc.Driver"
10 connectionURL="jdbc:mysql://localhost:3306/mysql"
11 userId="root"
12 password="root">
13 </jdbcConnection>
14 <javaTypeResolver >
15 <!-- 不需要将数据的小数类型，生成为BigDecimal -->
16 <property name="forceBigDecimals" value="false" />
17 </javaTypeResolver>
18 <!-- 生成数据库表对应的pojo类 -->
19 <javaModelGenerator targetPackage="com.sinosoft.cms.dao"
20 targetProject="src/main/java">
21 <property name="enableSubPackages" value="false" />
22 <property name="trimStrings" value="true" />
23 </javaModelGenerator>
24 <!-- 生成mapper -->
25 <sqlMapGenerator targetPackage="com.sinosoft.cms.po"
26 targetProject="src/main/java">
27 <property name="enableSubPackages" value="false" />
28 </sqlMapGenerator>
29 <!-- 生成mapper.xml -->
30 <javaClientGenerator type="XMLMAPPER"

```

```

31 targetPackage="com.sinosoft.cms.dao"
32 targetProject="src/main/java">
33 <property name="enableSubPackages" value="false" />
34 </javaClientGenerator>
35 <!-- 指定数据库的表 与数据库表字段一一对应-->
36 <table schema="" tableName="student"
37 domainObjectName="StudentPo" mapperName="StudentDao"></table>
38
39 </context>
40 </generatorConfiguration>

```

3、依赖：

```

1 <dependencies>
2 <!-- mybatis-generator逆向工程 -->
3 <dependency>
4 <groupId>org.mybatis.generator</groupId>
5 <version>1.3.5</version>
6 <artifactId>mybatis-generator-core</artifactId>
7 </dependency>
8 <!-- mysql驱动 -->
9 <dependency>
10 <groupId>mysql</groupId>
11 <artifactId>mysql-connector-java</artifactId>
12 <version>5.1.47</version>
13 </dependency>
14 <!-- mybatis -->
15 <dependency>
16 <groupId>org.mybatis</groupId>
17 <artifactId>mybatis</artifactId>
18 <version>3.4.6</version>
19 </dependency>
20 </dependencies>

```

1、启动类(跟上面一样)

```

1 package com.sinosoft.cms.api;
2
3 import org.mybatis.generator.api.MyBatisGenerator;

```

```
4 import org.mybatis.generator.config.Configuration;
5 import org.mybatis.generator.config.xml.ConfigurationParser;
6 import org.mybatis.generator.internal.DefaultShellCallback;
7
8 import java.io.File;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 /**
13  * @program: sale-server
14  * @description:
15  * @author: MR.wjw
16  * @create:
17  */
18 public class MybatisGeneratorApp {
19     public static void main(String[] args) throws Exception {
20         try {
21             MybatisGeneratorApp generatorSqlmap = new MybatisGeneratorApp();
22             generatorSqlmap.generator();
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26
27     }
28
29     public void generator() throws Exception {
30
31         List<String> warnings = new ArrayList<String>();
32         boolean overwrite = true;
33         //指定 逆向工程配置文件全路径
34         File configFile = new File("D:\\GitLocal\\mybatis-generator\\src\\main\\resources\\mybatis-generator.xml");
35         ConfigurationParser cp = new ConfigurationParser(warnings);
36         Configuration config = cp.parseConfiguration(configFile);
37         DefaultShellCallback callback = new DefaultShellCallback(overwrite);
38         MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,
39             callback, warnings);
40         myBatisGenerator.generate(null);
41         System.out.println("=====");
42
43     }
```

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

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!DOCTYPE generatorConfiguration PUBLIC
3  "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
4  "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd" >
5  <generatorConfiguration>
6  <context id="DB2Tables" defaultModelType="flat"
7  targetRuntime="MyBatis3">
8  <!-- 生成的Java文件的编码 -->
9  <property name="javaFileEncoding" value="UTF-8"/>
10 <!-- 格式化java代码 -->
11 <property name="javaFormatter" value="org.mybatis.generator.api.dom.DefaultJavaFormatter"/>
12 <!-- beginningDelimiter和endingDelimiter: 指明数据库的用于标记数据库对象名的
13 的符号，比如ORACLE就是双引号，MYSQL默认是`反引号； -->
14 <property name="beginningDelimiter" value="`"/>
15 <property name="endingDelimiter" value="`"/>
16 <!-- JavaBean 实现 序列化 接口 -->
17 <plugin type="org.mybatis.generator.plugins.SerializablePlugin"/>
18
19 <!-- generat entity时,生成toString -->
20 <plugin type="org.mybatis.generator.plugins.ToStringPlugin"/>
21 <!-- 此处是将Example后缀改名为Selective 当然 想改成什么都行~ -->
22 <plugin type="org.mybatis.generator.plugins.RenameExampleClassPlugin">
23 <property name="searchString" value="Example$"/>
24 <property name="replaceString" value="Selective"/>
25 </plugin>
26
27 <commentGenerator type="org.mybatis.generator.internal.DefaultCommentGenerator">
28 <!-- 是否去除自动生成的注释 true: 是 : false:否 -->
29 <property name="suppressAllComments" value="true"/>
30 </commentGenerator>
31
32 <!--数据库连接的信息：驱动类、连接地址、用户名、密码 -->
33 <jdbcConnection driverClass="org.postgresql.Driver"
34 connectionURL="jdbc:postgresql://10.252.68.222:5432/xgpgdev" userId="pi
35 cc"

```

```

34 password="PICC@xgfwf">
35 <property name="nullCatalogMeansCurrent" value="true"/>
36 </jdbcConnection>
37 <!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer，为 true时把J
DBC DECIMAL 和
38 NUMERIC 类型解析为java.math.BigDecimal -->
39 <javaTypeResolver>
40 <property name="forceBigDecimals" value="true"/>
41 </javaTypeResolver>
42
43 <!-- targetProject:生成PO类的位置 -->
44 <javaModelGenerator targetPackage="com.sinosoft.cms.po.base"
45 targetProject="src/main/java">
46 <!-- 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 -->
47 <property name="enableSubPackages" value="true"/>
48 <!-- 从数据库返回的值被清理前后的空格 -->
49 <property name="trimStrings" value="true"/>
50 </javaModelGenerator>
51 <!-- targetProject:mapper映射文件生成的位置 -->
52 <sqlMapGenerator targetPackage="base"
53 targetProject="src/main/resources/mapping">
54 <!-- enableSubPackages:是否让schema作为包的后缀 -->
55 <property name="enableSubPackages" value="false"/>
56 </sqlMapGenerator>
57
58 <!-- targetPackage: mapper接口生成的位置 -->
59 <javaClientGenerator type="XMLMAPPER" targetPackage="com.sinosoft.cms.d
ao.base"
60 targetProject="src/main/java">
61 <!-- enableSubPackages:是否让schema作为包的后缀 -->
62 <property name="enableSubPackages" value="true"/>
63 </javaClientGenerator>
64
65 <!-- 指定数据库表 schema不用填写 tableName="表名" domainObjectName=生成的P
O的名字
66 mapperName=生成的mapper的名字-->
67 <!-- <table schema="" tableName="WECHAT_MESSAGE"/> -->
68
69 <table schema="" tableName="laassessnotquarter"
70 domainObjectName="LAAssessNotQuarterPo" mapperName="LAAssessNotQuarterD
ao" ></table>

```

```

71 <!-- 指定数据库所有表 -->
72 <!-- <table schema="" tableName=""/> -->
73 <!-- 有些表的字段需要指定java类型
74 <table schema="" tableName="">
75 <columnOverride column="" javaType="" />
76 </table> -->
77 </context>
78 </generatorConfiguration>

```

3、依赖：

```

1 <dependencies>
2 <!-- mybatis-generator逆向工程 -->
3 <dependency>
4 <groupId>org.mybatis.generator</groupId>
5 <version>1.3.5</version>
6 <artifactId>mybatis-generator-core</artifactId>
7 </dependency>
8 <!--连接PG库的驱动-->
9 <dependency>
10 <groupId>org.postgresql</groupId>
11 <artifactId>postgresql</artifactId>
12 <version>42.2.5</version>
13 </dependency>
14 <!-- mybatis -->
15 <dependency>
16 <groupId>org.mybatis</groupId>
17 <artifactId>mybatis</artifactId>
18 <version>3.4.6</version>
19 </dependency>
20 </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。

```
1 <build>
2   <finalName>${project.artifactId}</finalName>
3   <plugins>
4     <plugin>
5       <groupId>org.apache.maven.plugins</groupId>
6       <artifactId>maven-surefire-plugin</artifactId>
7       <version>2.9</version>
8       <configuration>
9         <includes>
10          <include>*/SwaggerTest.java</include>
11        </includes>
12      </configuration>
13    </plugin>
14
15    <plugin>
16      <groupId>org.apache.maven.plugins</groupId>
17      <artifactId>maven-surefire-plugin</artifactId>
18      <configuration>
19        <includes>
20          <include>*/SwaggerTest.java</include>
21        </includes>
22        <systemPropertyVariables>
23          <io.springfox.staticdocs.outputDir>${project.build.directory}/swagger
24        </io.springfox.staticdocs.outputDir>
25          <io.swagger.json.uris>/v2/api-docs?group=UI,/v2/api-docs?
26          group=API</io.swagger.json.uris>
27          <io.swagger.json.output.name>acrc-ui-v1.json,acrc-api-v1.json</io.swagger.json.output.name>
28        </systemPropertyVariables>
29      </configuration>
30    </plugin>
31
32    <!--Mybatis代码自动生成器-->
33    <plugin>
34      <groupId>org.mybatis.generator</groupId>
35      <artifactId>mybatis-generator-maven-plugin</artifactId>
36      <version>1.3.5</version>
37      <dependencies>
38        <dependency>
39          <groupId>mysql</groupId>
```



```
40 <artifactId>mysql-connector-java</artifactId>
41 <version>5.1.39</version>
42 </dependency>
43 <dependency>
44 <groupId>org.mybatis.generator</groupId>
45 <artifactId>mybatis-generator-core</artifactId>
46 <version>1.3.5</version>
47 </dependency>
48 <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
49 <dependency>
50 <groupId>org.postgresql</groupId>
51 <artifactId>postgresql</artifactId>
52 <version>42.2.5</version>
53 </dependency>
54
55
56 </dependencies>
57 <executions>
58 <execution>
59 <id>Generate MyBatis Artifacts</id>
60 <phase>package</phase>
61 <goals>
62 <goal>generate</goal>
63 </goals>
64 </execution>
65 </executions>
66 <configuration>
67 <!-- 允许移动生成的文件 -->
68 <verbose>true</verbose>
69 <!-- 是否覆盖 -->
70 <overwrite>true</overwrite>
71 <!-- 自动生成的配置，指定generator文件的地址-->
72 <configurationFile>src/main/resources/mybatis-generator.xml</configurationFile>
73 </configuration>
74 </plugin>
75
76 </plugins>
77 </build>
```

