

# 03\_SpringBoot#DAO#TK-Mybatis#代码生成器

---

## 一、pom文件

## 二、生成器具体配置

## 三、使用测试

【TK-Mybatis#代码生成器】

时间：2023年04月27日 20:45:03

---

Mybatis代码生成器工作在DAO层，它会帮你生成entity和mapper（DAO+XML），使用它，你需要在项目中配置两个地方：

- maven项目的pom文件（project object model）添加新的插件（plugin）
  - mybatis代码生成器配置文件（需要在pom文件中配置该文件的具体位置）
- 

## 一、pom文件

- ✓你需要添加一个名为 `mybatis-generator-maven-plugin` 的插件，在你的maven项目配置文件中；
- ✓这个插件你可以在[Maven官方仓库](#)中找到最新版。
- ✓插件需要使用两个核心依赖，这些依赖只会在代码生成时使用，不会打包到你的项目中：
  - `org.mybatis.generator#mybatis-generator-core`
  - `tk.mybatis#mapper`
- ✓当然还需要提供对应的数据库连接驱动：
  - `mysql#mysql-connector-java`
- ✓`configuration`标签下，你可以配置如下属性（使用IDEA的话，可直接Ctrl+左键查看介绍）：
  - `verbose`

```
<parameter>
  <name>verbose</name>
  <type>boolean</type>
  <required>false</required>
  <editable>true</editable>
  <description>Specifies whether the mojo writes progress messages to the log.</description>
</parameter>
```

是否输出过程日志



- 
- overwrite
  - 是否覆盖旧的文件（默认false）
- configurationFile
  - 配置文件位置

```
1 <plugin>
2   <groupId>org.mybatis.generator</groupId>
3   <artifactId>mybatis-generator-maven-plugin</artifactId>
4   <version>1.4.2</version>
5   <dependencies>
6     <dependency>
7       <groupId>mysql</groupId>
8       <artifactId>mysql-connector-java</artifactId>
9       <version>8.0.31</version>
10    </dependency>
11    <dependency>
12      <groupId>org.mybatis.generator</groupId>
13      <artifactId>mybatis-generator-core</artifactId>
14      <version>1.4.2</version>
15    </dependency>
16    <dependency>
17      <groupId>tk.mybatis</groupId>
18      <artifactId>mapper</artifactId>
19      <version>4.2.2</version>
20    </dependency>
21  </dependencies>
22  <executions>
23    <execution>
24      <id>Generate MyBatis Artifacts</id>
25      <phase>package</phase>
26      <goals>
27        <goal>generate</goal>
28      </goals>
29    </execution>
30  </executions>
31  <configuration>
32    <verbose>true</verbose>
33    <overwrite>true</overwrite>
34    <configurationFile>src/main/resources/mybatis-generator.xml</configurationFile>
35  </configuration>
36 </plugin>
```

🔧 如果你想要将你的SpringBoot项目打包成可运行的jar包，你可能会用到springboot打包插件：

```
1 <plugin>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-maven-plugin</artifactId>
4   <version>2.7.10</version>
5   <configuration>
6     <mainClass>org.q.w.e.App</mainClass>
7   </configuration>
8   <executions>
9     <execution>
10      <goals>
11        <goal>repackage</goal>
12      </goals>
13    </execution>
14  </executions>
15 </plugin>
```

## 二、生成器具体配置

这里不会详细介绍所有参数，会提供一个拿来即用的配置

- 官方文档：<https://gitee.com/free/Mybatis-Generator/wikis/4.generator/4.1.mapper.generator>

这里有几个核心配置项：

- ✓数据库连接参数
- ✓生成的实体位置
- ✓生成的mapper（DAO）位置
- ✓生成的mapper（XML）位置
- ✓要生成哪张表（还是所有表）

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration
3     PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//E
4     N"
5         "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
6 <!-- 官方文档: https://gitee.com/free/Mapper/wikis/4.generator/4.1.mapperge
7     nerator -->
8 <generatorConfiguration>
9     <context id="Mysql" targetRuntime="MyBatis3Simple" defaultModelType="f
10         lat">
11         <property name="javaFileEncoding" value="UTF-8"/>
12
13         <plugin type="tk.mybatis.mapper.generator.MapperPlugin">
14             <property name="mappers" value="tk.mybatis.mapper.common.Mappe
15                 r"/>
16             <property name="useMapperCommentGenerator" value="true"/>
17             <property name="beginningDelimiter" value="`"/>
18             <property name="endingDelimiter" value="`"/>
19             <property name="caseSensitive" value="false"/>
20             <property name="forceAnnotation" value="true"/>
21             <property name="generateColumnConsts" value="false"/>
22             <property name="lombok" value="Data,NoArgsConstructor"/>
23         </plugin>
24
25         <jdbcConnection driverClass="com.mysql.cj.jdbc.Driver"
26             connectionURL="jdbc:mysql://192.168.204.101:23306/
27                 dev_test?useUnicode=true&characterEncoding=utf8&useOldAliasMetadat
28                 aBehavior=true&zeroDateTimeBehavior=convertToNull&allowMultiQuerie
29                 s=true&serverTimezone=UTC"
30                 userId="root"
31                 password="Mysql12345">
32             <property name="useInformationSchema" value="true"/>
33             <property name="nullCatalogMeansCurrent" value="true"/>
34         </jdbcConnection>
35
36         <!-- 数值与时间格式: BigDecimal、LocalDateTime -->
37         <javaTypeResolver>
38             <property name="forceBigDecimals" value="true"/>
39             <property name="useJSR310Types" value="true"/>
40         </javaTypeResolver>
```

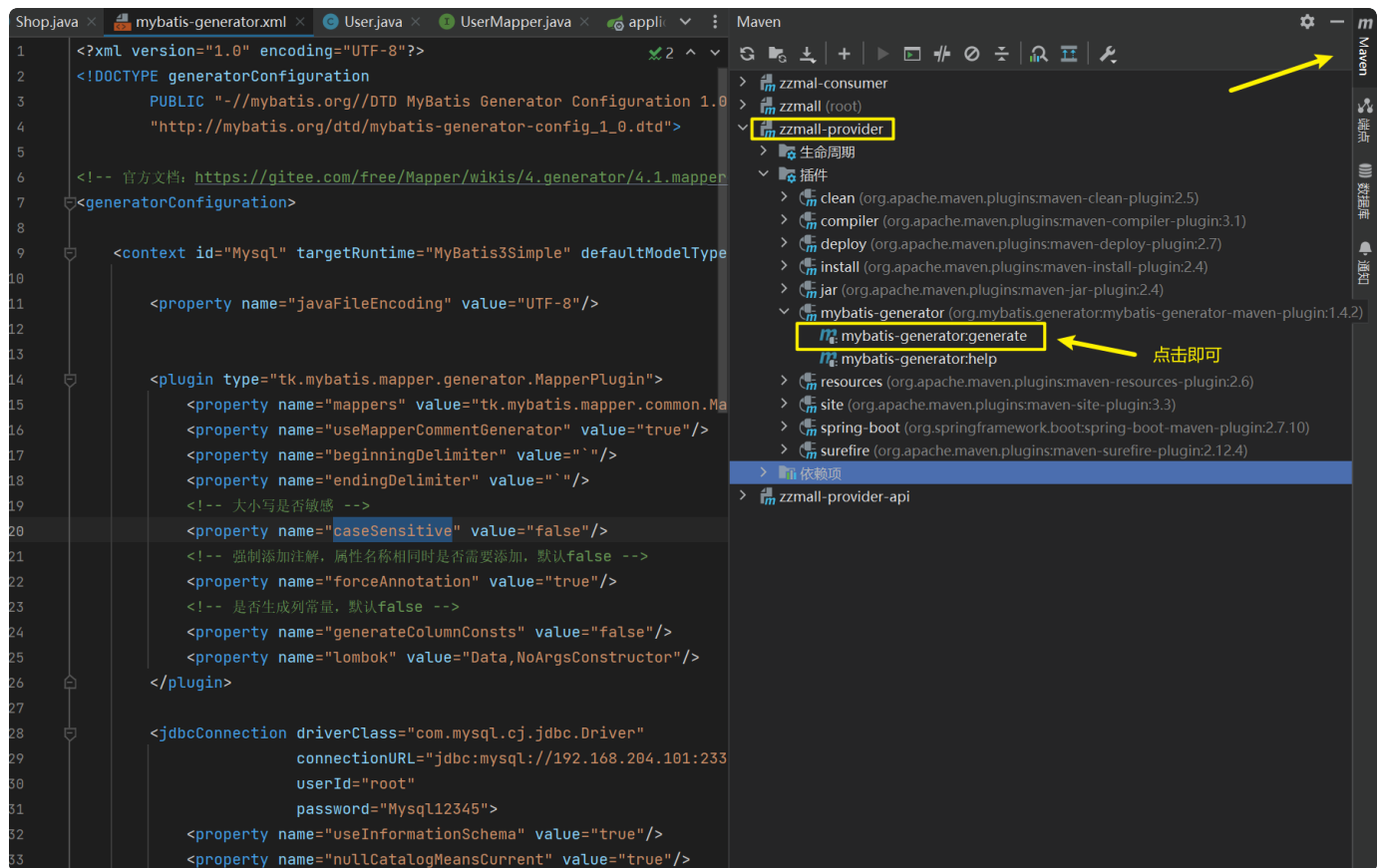
```

39     <javaModelGenerator targetPackage="org.example.zzmall.provider.domain.entity111"
40
41         targetProject="src/main/java">
42         <!--继承BaseModel-->
43         <!--<property name="rootClass" value="BaseModel"/>-->
44         <property name="enableSubPackages" value="true"/>
45         <property name="trimStrings" value="true"/>
46     </javaModelGenerator>
47
48     <sqlMapGenerator targetPackage="org.example.zzmall.provider.mapper111" targetProject="src/main/resources">
49         <property name="enableSubPackages" value="true"/>
50     </sqlMapGenerator>
51
52     <javaClientGenerator targetPackage="org.example.zzmall.provider.mapper111"
53
54         targetProject="src/main/java"
55         type="ANNOTATEDMAPPER">
56         <property name="enableSubPackages" value="true"/>
57     </javaClientGenerator>
58
59     <!-- 已生成的表-->
60     <!-- <table tableName="scn_rwm_ostore_dtl_lt" domainObjectName="ScnRwmOstoreDtlLt"-->
61     <!-- enableCountByExample="false" enableUpdateByExample="false"
62     <!-- enableDeleteByExample="false" enableSelectByExample="false"
63     <!-- selectByExampleQueryId="false"-->
64     <!-- </table-->
65     <table tableName="%" enableCountByExample="false"
66         enableUpdateByExample="false" enableDeleteByExample="false"
67         enableSelectByExample="false" selectByExampleQueryId="false">
68     </table>
69 </context>
</generatorConfiguration>

```

- **caseSensitive** 是否区分大小写，默认值 `false`。如果数据库区分大小写，这里就需要配置为 `true`，这样当表名为 `USER` 时，会生成 `@Table(name = "USER")` 注解，否则使用小写 `user` 时会找不到表。
- **forceAnnotation** 是否强制生成注解，默认 `false`，如果设置为 `true`，不管数据库名和字段名是否一致，都会生成注解（包含 `@Table` 和 `@Column`）。
- **beginningDelimiter** 和 **endingDelimiter** 开始和结束分隔符，对于有关键字的情况下适用。
- **useMapperCommentGenerator** 是否使用通用 Mapper 提供的注释工具，默认 `true` 使用，这样在生成代码时会包含字段的注释（目前只有 `mysql` 和 `oracle` 支持），设置 `false` 后会用默认的，或者你可以配置自己的注释插件。
- **generateColumnConsts** 在生成的 `model` 中，增加字段名的常量，便于使用 `Example` 拼接查询条件的时候使用。
- **lombok** 增加 `model` 代码生成时，可以直接生成 `lombok` 的 `@Getter@Setter@ToString@Accessors(chain = true)` 四类注解，使用者在插件配置项中增加 `<property name="lombok" value="Getter,Setter,ToString,Accessors"/>` 即可生成对应包含注解的 `model` 类。

### 三、使用测试



```

import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * 表名: user
 * 表注释: 用户表
 */
5 个用法
@Data
@NoArgsConstructor
@Table(name = "`user`")
public class User {

    /**
     * 主键ID
     */
    @Id
    @Column(name = "`id`")
    private String id;

    /**
     * 用户名
     */
    @Column(name = "`username`")
    private String username;

    /**
     * 密码
     */
    @Column(name = "`password`")
    private String password;
}

```

表注释 (某些旧版本mybatis生成器不支持表注释)

lombok注解 (某些旧版本mybatis生成器不支持lombok)

生成的实体

列注释

- 1.配置强制注解: 同名也会生成注解
- 2.注意属性左右的`符号 (某些旧版本mybatis生成器不支持这两个配置)

提示: 请使用最新版

```

import org.example.zzmall.provider.domain.entity111.User;
import tk.mybatis.mapper.common.Mapper;

1 用法
public interface UserMapper extends Mapper<User> {
}

```

继承自定义接口

```

<plugin type="tk.mybatis.mapper.generator.MapperPlugin">
  <property name="mappers" value="tk.mybatis.mapper.common.Mapper"/>
  <property name="useMapperCommentGenerator" value="true"/>
</plugin>

```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="org.example.zzmall.provider.mapper111.UserMapper">
  <resultMap id="BaseResultMap" type="org.example.zzmall.provider.domain.entity111.User">
    <!--
      WARNING - @mbg.generated
    -->
    <id column="id" jdbcType="VARCHAR" property="id" />
    <result column="username" jdbcType="VARCHAR" property="username" />
    <result column="password" jdbcType="VARCHAR" property="password" />
  </resultMap>
</mapper>
```

推荐插件: MybatisX

生成的mapper (XML)