# MyBatis-Generator代码生成框架

[MyBatis-Generator代码生成框架 1](#_Toc25194)

[基于Maven插件集成MyBatis-Generator 代码生成框架 1](#_Toc3057)

[基于代码+配置文件实现MyBatis-Generator 代码生成 8](#_Toc8490)

[基于命令行实现MyBatista-Generator 实现代码生成 9](#_Toc6214)

## 基于Maven插件集成MyBatis-Generator 代码生成框架

第一步：pom.xml 文件，添加MyBatis-Generator插件依赖。

<build>

<plugins>

<plugin>

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

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

<version>1.3.5</version>

<configuration> <configurationFile>src/main/resources/mybatis-generator/mybatis-generator-cfg.xml</configurationFile>

<verbose>true</verbose>

<overwrite>true</overwrite>

</configuration>

<executions>

</executions>

<dependencies>

<dependency>

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

<artifactId>mybatis-generator-core</artifactId>

<version>1.3.5</version>

</dependency>

<!--oracle 驱动程序 -->

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc6</artifactId>

<version>11.2.0.2.0</version>

</dependency>

</dependencies>

</plugin>

</plugins>

</build>

第二步：添加MyBatis-Generator 代码生成器依赖配置文件（mybatis-generator-cfg.xml）

在Spring Boot 项目的resource目录下新增mybatis-generator 文件夹，新增mybatis-generator-cfg.xml文件。

<?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>

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

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

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

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

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

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

<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="true"/> <!-- 是否生成注释代时间戳-->

</commentGenerator>

<!--数据库链接地址账号密码-->

<jdbcConnection driverClass="oracle.jdbc.driver.OracleDriver"

connectionURL="jdbc:oracle:thin:@x.x.x.x:1521:orcl"

userId="用户名"

password="密码">

</jdbcConnection>

<!--生成Model类存放位置-->

<javaModelGenerator targetPackage="com.zzg.model" targetProject="src/main/java">

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

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

</javaModelGenerator>

<!-- 生成mapxml文件 -->

<sqlMapGenerator targetPackage="mapper" targetProject="src/main/resources">

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

</sqlMapGenerator>

<!-- 生成mapxml对应client，也就是接口dao -->

<javaClientGenerator targetPackage="com.zzg.mapper" targetProject="src/main/java" type="XMLMAPPER">

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

</javaClientGenerator>

<!-- 要生成的表 tableName是数据库中的表名或视图名 domainObjectName是实体类名 mapperName是Dao名-->

<table tableName="数据库表名" domainObjectName="生成Entity名" mapperName="生成dao名" enableCountByExample="false"

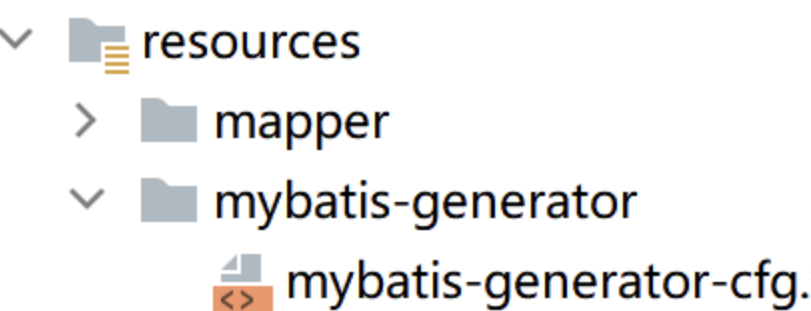
enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false">

</table>

</context>

</generatorConfiguration>

效果截图：

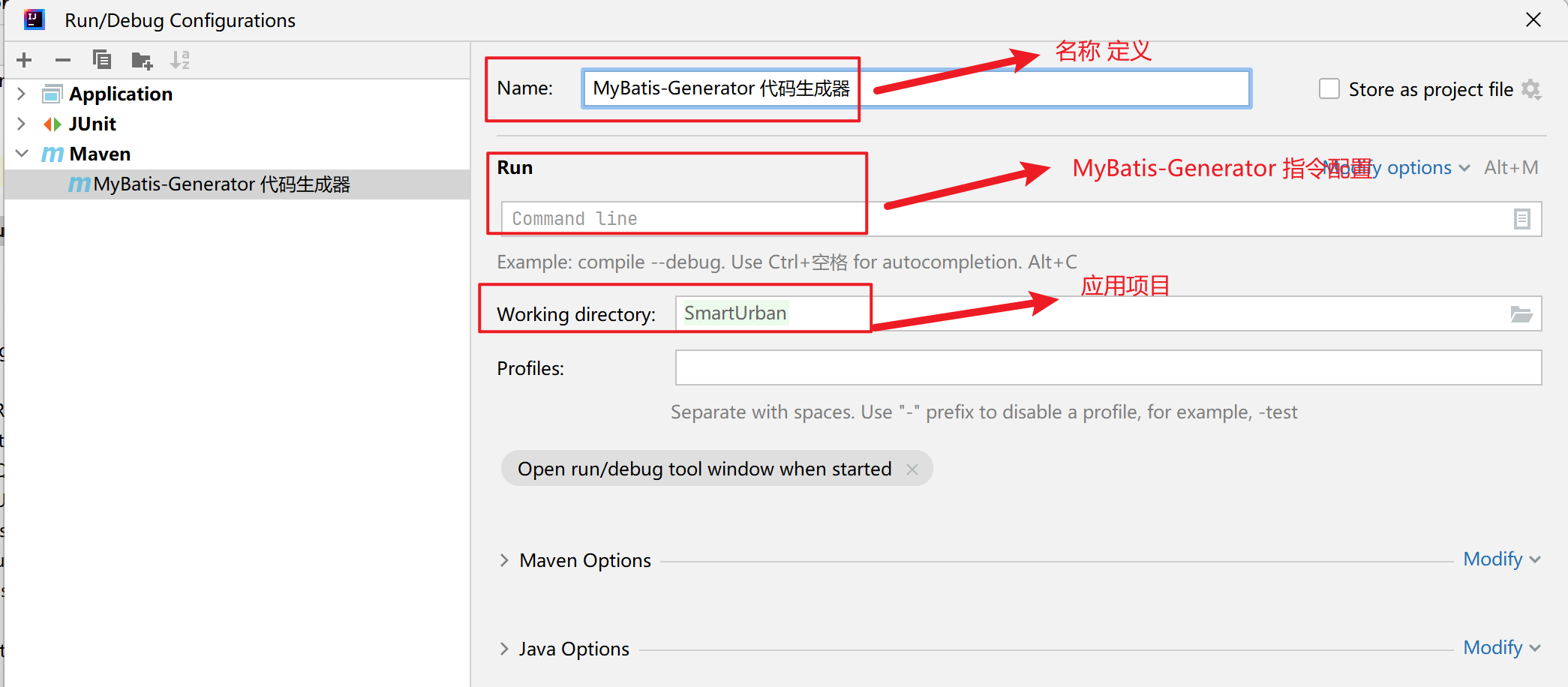


第三步：SpringBoot项目添加MyBatis-Generator 代码生成器Maven 指令。

点击菜单栏Run-》Edit Configurations.，弹出如下截图窗口。

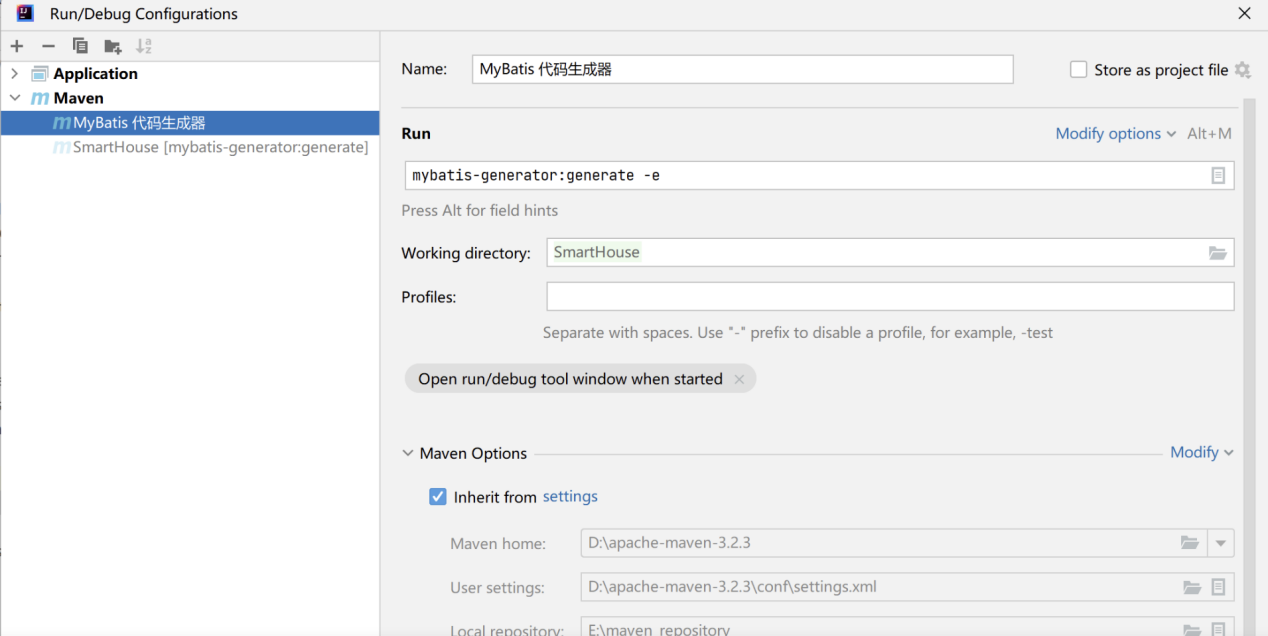


点击+，选择添加Maven 指令, 弹出如下截图窗口。



最后点击Apply + OK , 完成MyBatis-Generator Maven指令配置。

实例截图：

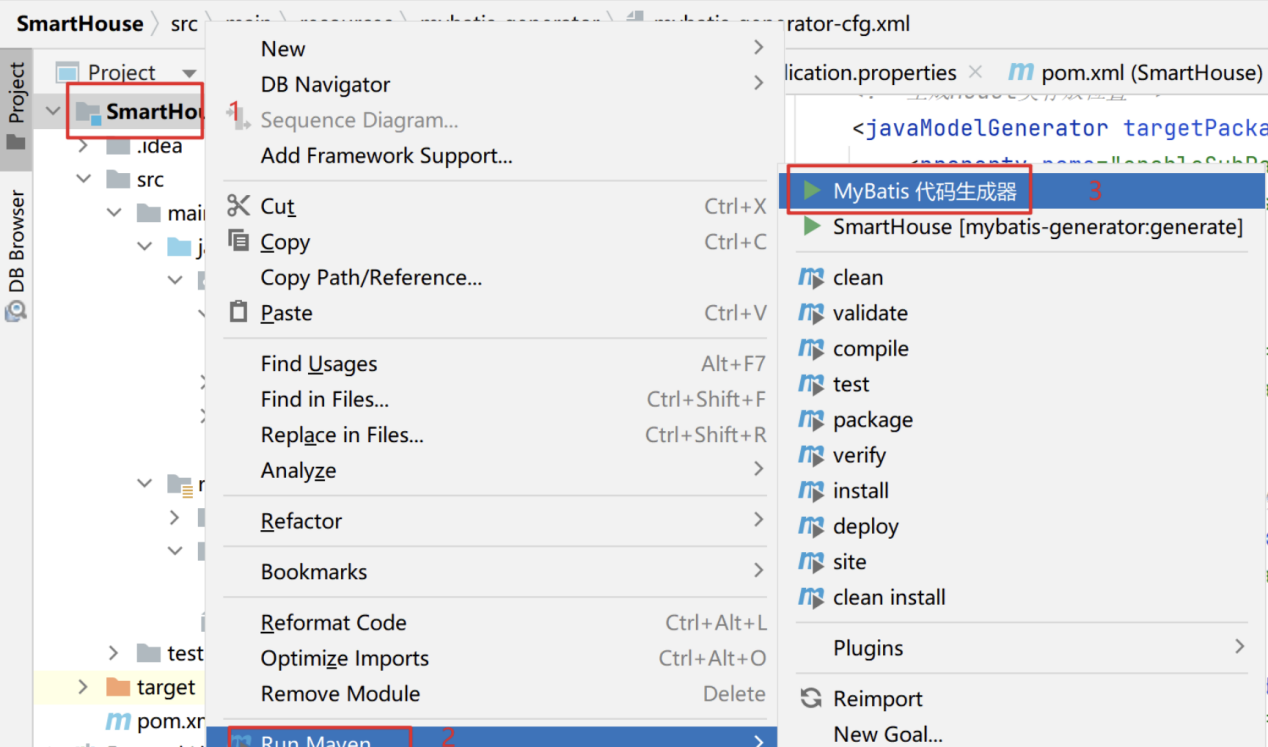


MyBatis-Generator Maven 指令：

mybatis-generator:generate -e

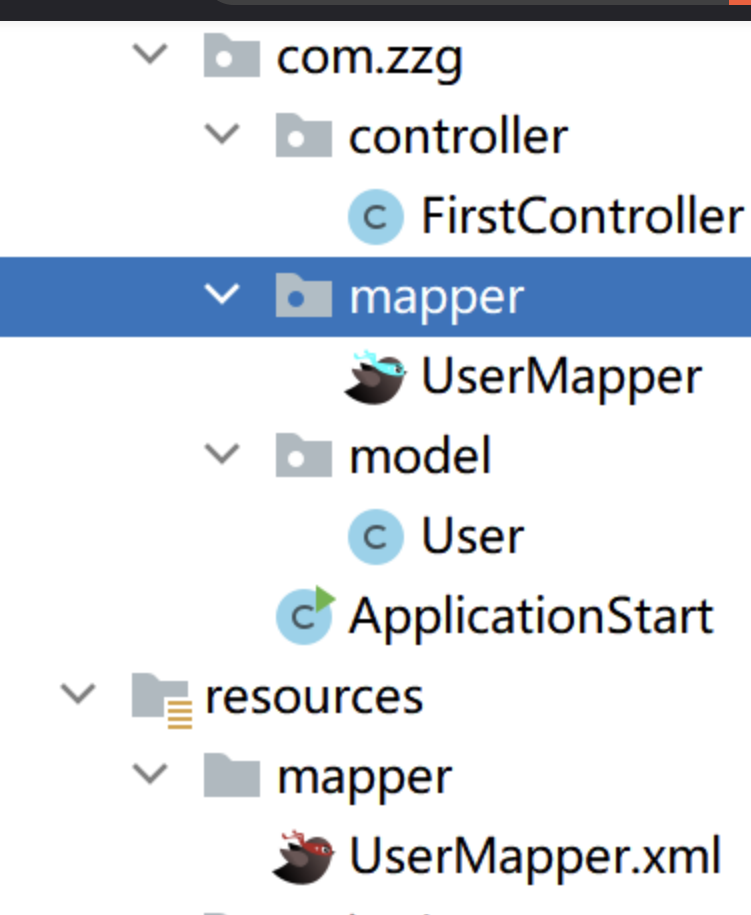
第四步：基于MyBatis-Generator Maven 代码生成指令，生成Entity\Mapper

选择SpringBoot 项目，右击Run Maven ，选择MyBatis-Generator 代码生成指定。截图如下：



温馨提示：请按照数字序号进行选择操作。

生成Entity\Mapper 效果截图：



## 基于代码+配置文件实现MyBatis-Generator 代码生成

第一步：pom.xml 文件, 添加mybatis-generator-core.jar 包依赖。

<dependency>

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

<artifactId>mybatis-generator-core</artifactId>

<version>1.4.0</version>

</dependency>

第二步：添加MyBatis-Generator 代码生成器依赖配置文件（mybatis-generator-cfg.xml）

此内容请参考：基于Maven插件集成MyBatis-Generator代码生成框架。

第三步：通用代码生成器核心功能代码：

List<String> warnings = new ArrayList<>();

// 如果已经存在生成过的文件是否进行覆盖

boolean overwrite = true;

File configFile = new File("ClassPath路径/mybatis-generator-cfg.xml")；

ConfigurationParser cp = new ConfigurationParser(warnings);

Configuration config = cp.parseConfiguration(configFile);

DefaultShellCallback callback = new DefaultShellCallback(overwrite);

MyBatisGenerator generator = new MyBatisGenerator(config, callback, warnings);

generator.generate(null);

## 基于命令行实现MyBatista-Generator 实现代码生成

第一步：下载mybatis-generator-core-\*.\*.\*.\*.jar 包, 我这里选用的版本为:mybatis-generator-core-1.3.2.jar

第二步：添加MyBatis-Generator 代码生成器依赖配置文件（mybatis-generator-cfg.xml）

此内容请参考：基于Maven插件集成MyBatis-Generator代码生成框架。

第三步：使用Win + R 打开控制台命令窗口，目录切换至mybatis-generator-core-1.3.2.jar 所在目录，执行如下指令：

java -jar mybatis-generator-core-1.3.2.jar -configfile mybatis-generator-cfg.xml -overwrite

温馨提示：Oracle 数据库驱动包地址，需要在mybatis-generator-cfg.xml 文件中指定。

在配置文件中，添加如下配置：

<!-- 数据库驱动包位置 -->

<classPathEntry location="F:/mybatis-generator-parent/generator/ojdbc6.jar" />