**Mybatis -代码自动生成（generatorConfig.xml）**

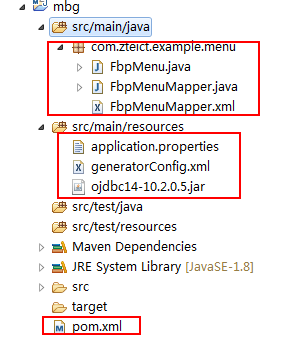
<https://blog.csdn.net/wangxy799/article/details/60870361>

参考：<http://blog.csdn.net/jinshiyill/article/details/51546676>   
官方网址: <http://www.mybatis.org/generator/configreference/xmlconfig.html>   
原码:<http://download.csdn.net/detail/wangxy799/9773975>

## 1. 环境：

Maven 3.39   
Oracle 11g   
Mybatis 3.4.1   
Maven project

## 2. 目录结构



## 3. pom添加插件

<dependencies>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.4.1</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

</dependency>

</dependencies>

<build>

<plugins>

<!-- mybatis-generator自动生成代码插件 -->

<plugin>

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

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

<version>1.3.5</version>

</plugin>

</plugins>

</build>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24

## 4. 新建generatorConfig.xml 文件(放在src/main/resources目录下)

<?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>

<!-- 引入配置文件 -->

<properties resource="application.properties"/>

<!-- 指定数据库连接驱动jara地址 -->

<classPathEntry

location="${generator.location}" />

<!-- 一个数据库一个context -->

<context id="sqlserverTables">

<!-- 生成的pojo，将implements Serializable -->

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

<!-- 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URL、用户名、密码 -->

<jdbcConnection driverClass="${db.example.driver}"

connectionURL="${db.example.url}" userId="${db.example.username}"

password="${db.example.password}">

</jdbcConnection>

<!-- 类型转换 -->

<javaTypeResolver>

<!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer true，把JDBC DECIMAL

和 NUMERIC 类型解析为java.math.BigDecimal -->

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

</javaTypeResolver>

<!-- 生成model模型，对应的包路径，以及文件存放路径(targetProject)，targetProject可以指定具体的路径,如./src/main/java，

也可以使用“MAVEN”来自动生成，这样生成的代码会在target/generatord-source目录下 -->

<javaModelGenerator targetPackage="${generator.targetPackage}"

targetProject="./src/main/java">

<!-- 是否在当前路径下新加一层schema,eg：fase路径com.oop.eksp.user.model， true:com.oop.eksp.user.model.[schemaName] -->

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

<!-- 从数据库返回的值被清理前后的空格 -->

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

</javaModelGenerator>

<!--对应的mapper.xml文件 -->

<sqlMapGenerator targetPackage="${generator.targetPackage}"

targetProject="./src/main/java">

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

</sqlMapGenerator>

<!-- 对应的Mapper接口类文件 -->

<javaClientGenerator type="XMLMAPPER"

targetPackage="${generator.targetPackage}" targetProject="./src/main/java">

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

</javaClientGenerator>

<!-- 列出要生成代码的所有表，这里配置的是不生成Example文件 -->

<!-- schema即为数据库名 tableName为对应的数据库表 domainObjectName是要生成的实体类 enable\*ByExample

是否生成 example类 -->

<table tableName="${gererator.tableName}" domainObjectName="${gererator.objectName}"

schema="${gererator.schema}"

enableCountByExample="false" enableUpdateByExample="false"

enableDeleteByExample="false" enableSelectByExample="false"

selectByExampleQueryId="false">

<!-- 忽略列，不生成bean 字段

<ignoreColumn column="FRED" />-->

<!-- 指定列的java数据类型

<columnOverride column="LONG\_VARCHAR\_FIELD" jdbcType="VARCHAR" /> -->

<!-- 用于指定生成实体类时是否使用实际的列名作为实体类的属性名。false是 Camel Case风格-->

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

</table>

</context>

</generatorConfiguration>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78

## 5. 设置需要生成的表及创建目录参数（application.properties放在src/main/resources目录下）

db.example.type=oracle

db.example.driver=oracle.jdbc.driver.OracleDriver

db.example.url=jdbc:oracle:thin:@ip:port:sid

db.example.username=username

db.example.password=password

#MBGInfo

generator.location=D:/project/eas\_std/BaseApp/lib/ojdbc14-10.2.0.5.jar

generator.targetPackage=com.zteict.example.menu

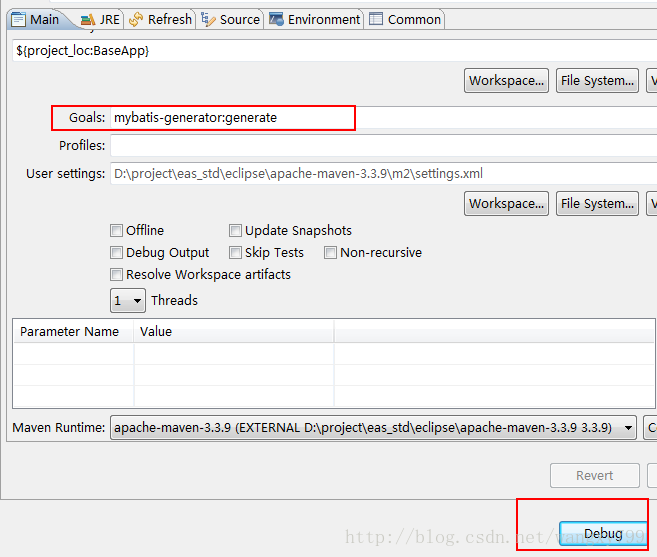
gererator.schema=fbp

gererator.tableName=fbp\_menu

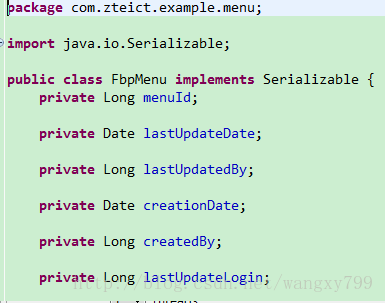
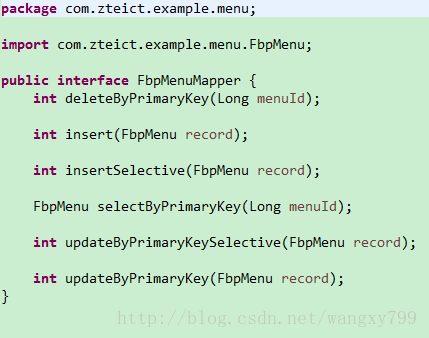
gererator.objectName=FbpMenu

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

## 6. 项目右键-RUN AS-MAVEN BUILD..

输入 mybatis-generator:generate   


## 7. 生成的文件

FbpMenu：   
   
FbpMenuMapper:   
   
Xml：   


版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/wangxy799/article/details/60870361