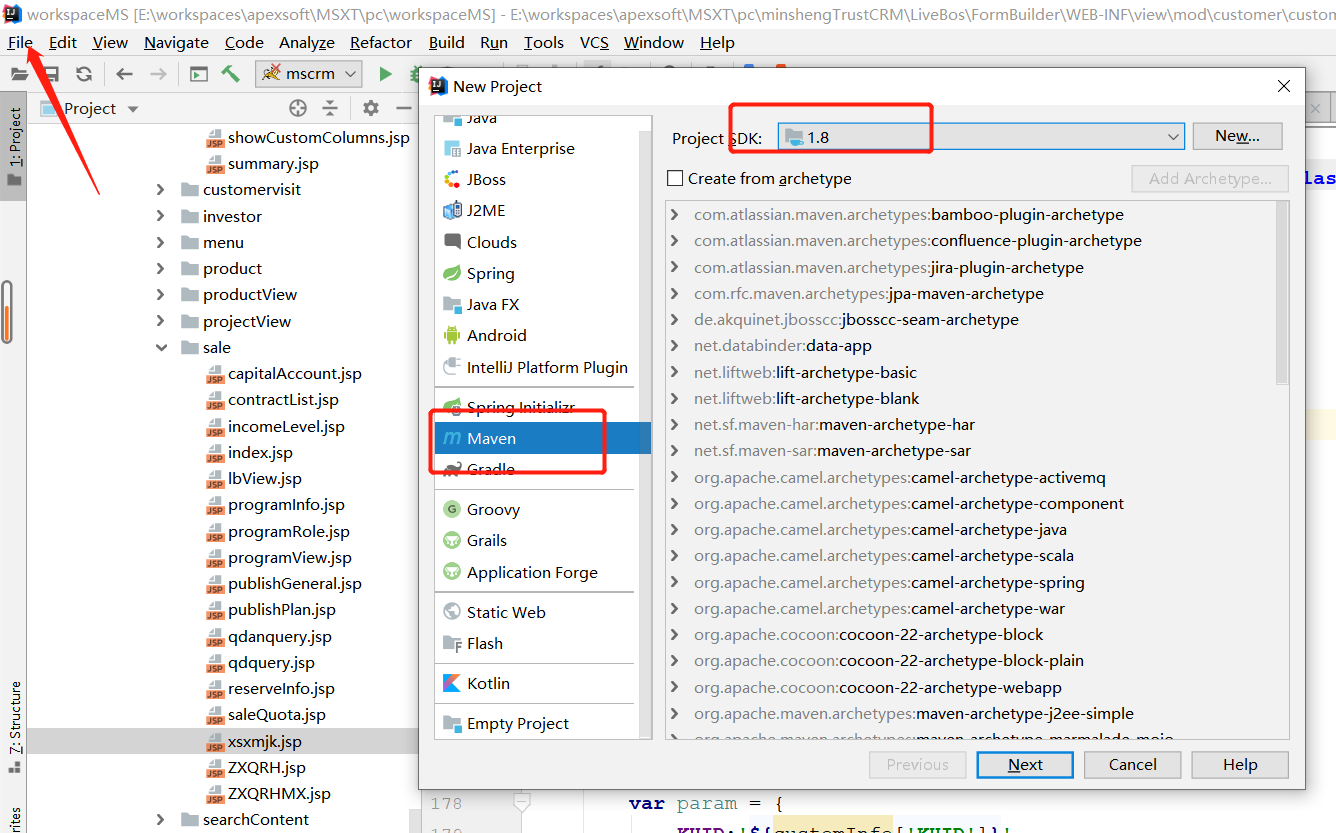
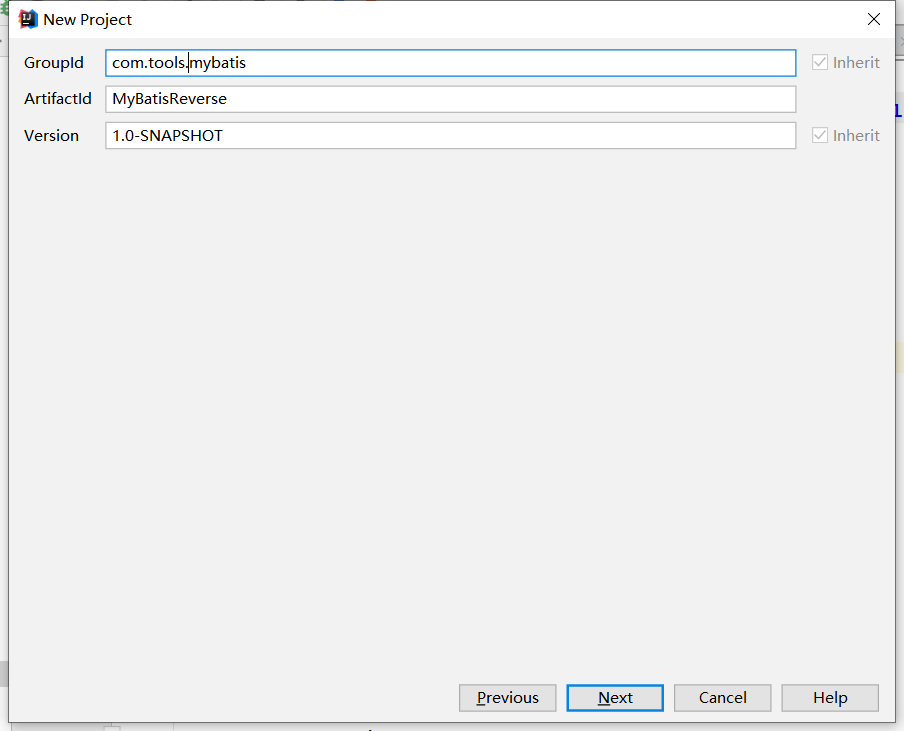
## **第一步，使用Idea创建Maven项目**

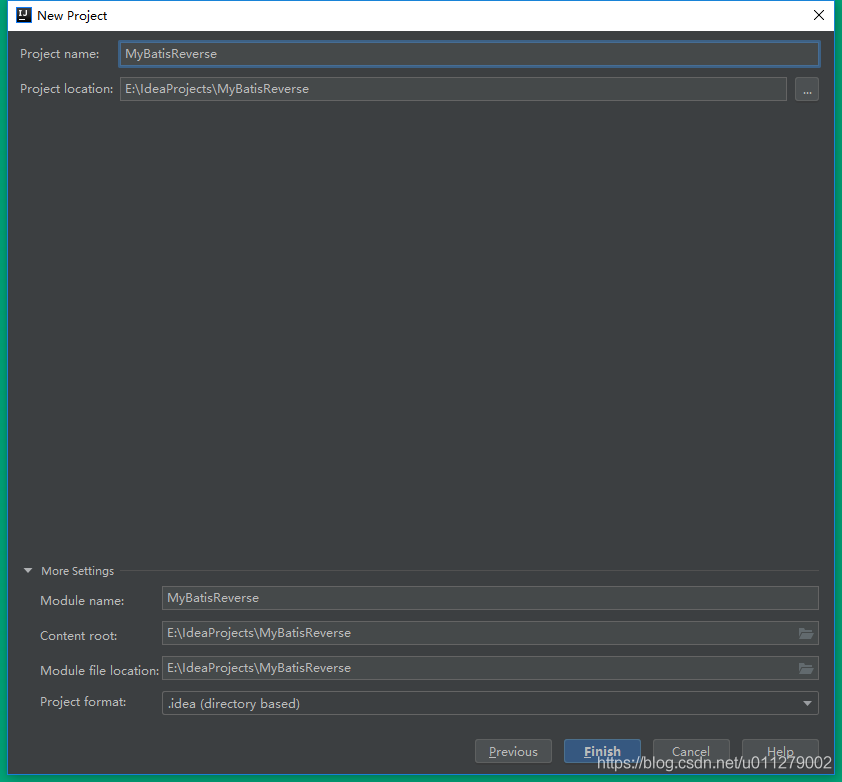
**1、**因为我的idea是打开的，所以直接点击file-new-projects-选择maven，jdk，直接next。



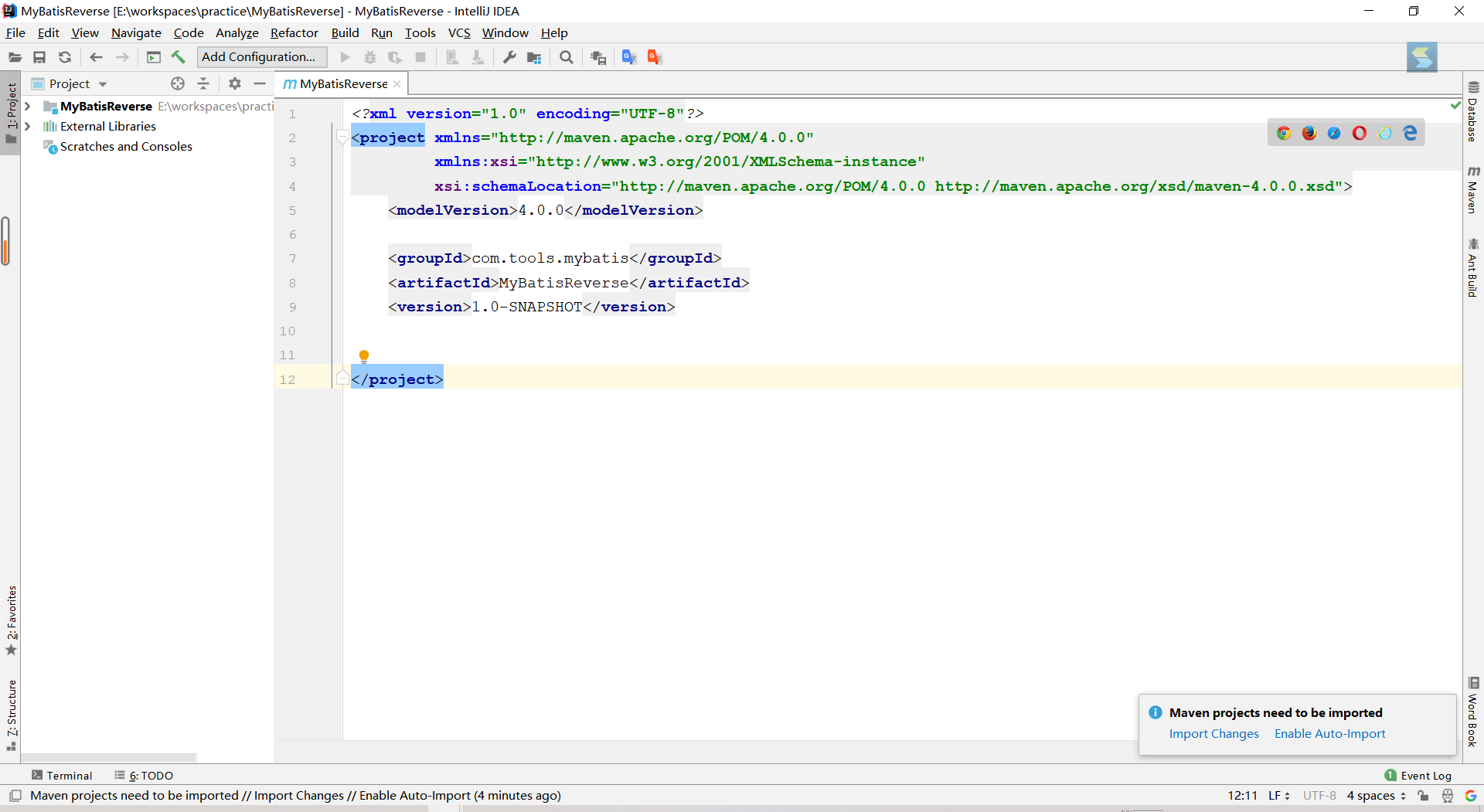
1. 输入项目的GroupId以及ArtifactId,然后Next



1. 继续finish

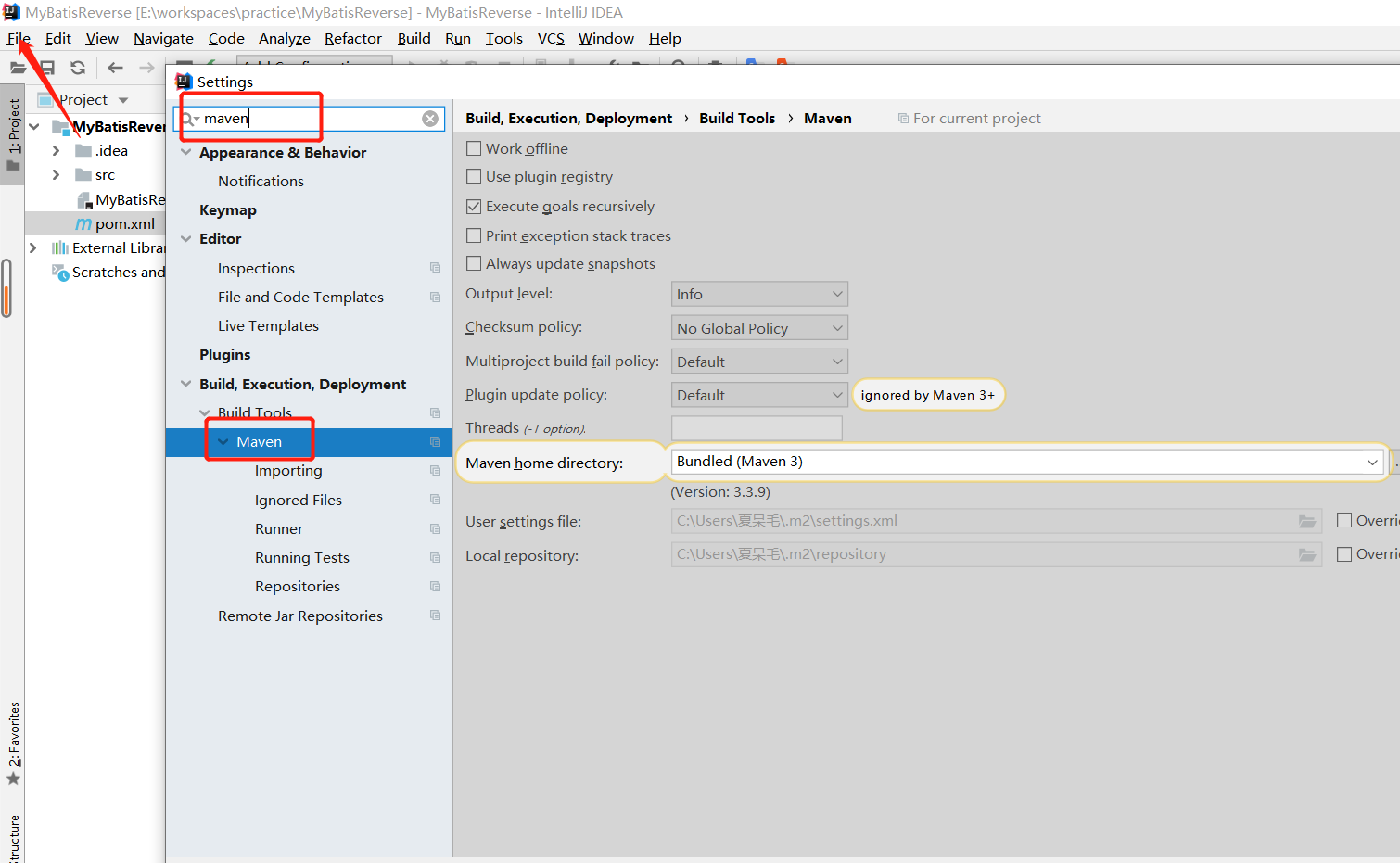


1. 这样，一个IDEA的Maven项目就创建完成了，大致的项目结构如下图所示，如果是第一次创建Maven项目，还请耐心等待一下，系统需要下载相应的Jar包。

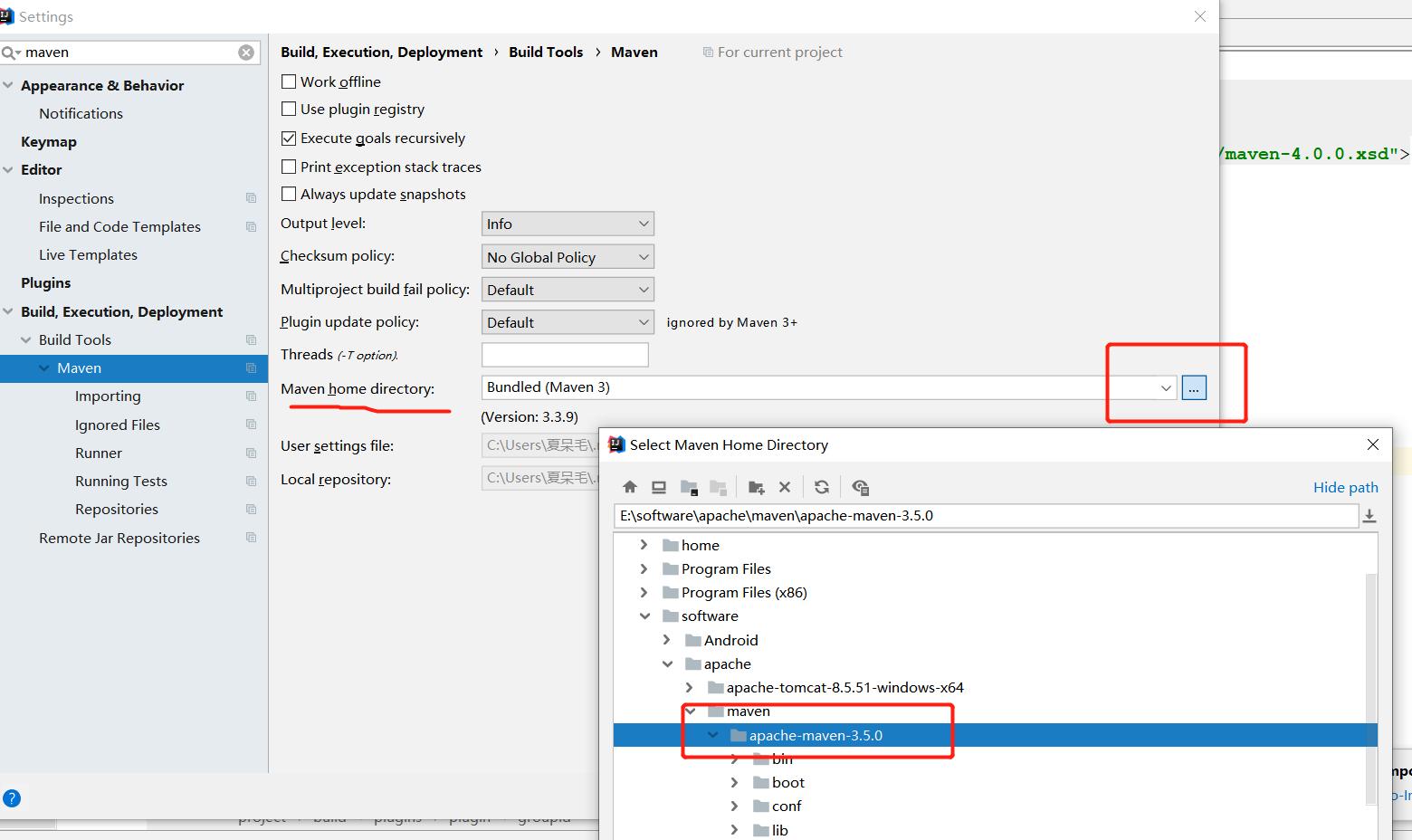


## **第二步，配置maven**

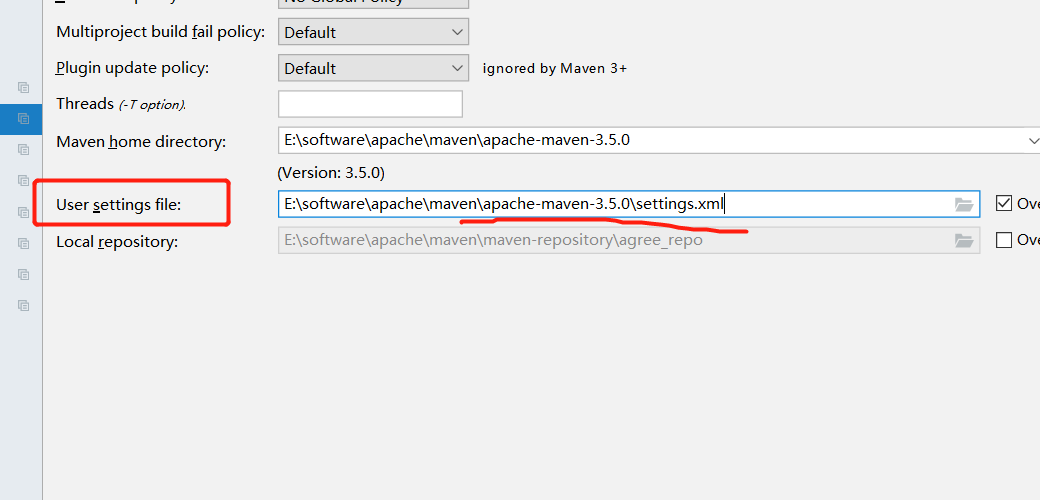
1. file-setting,直接搜索maven即可。

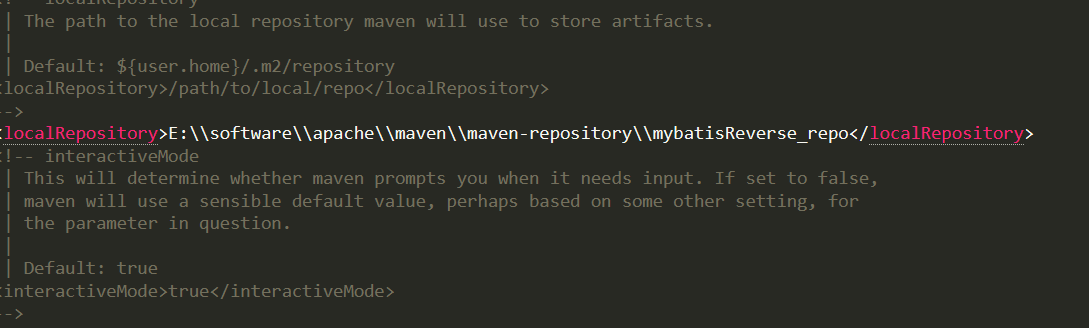


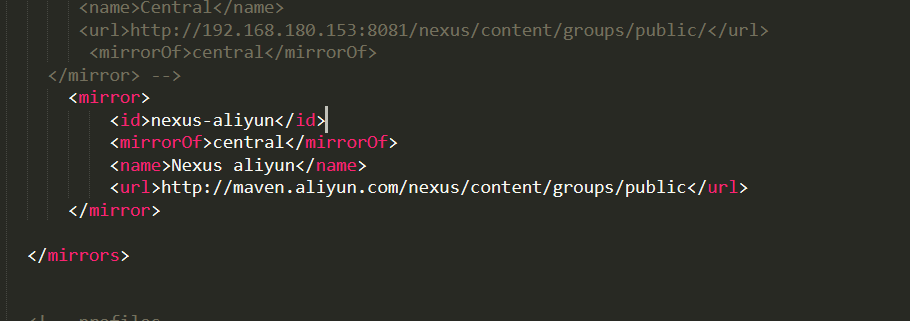
1. 配置maven路径



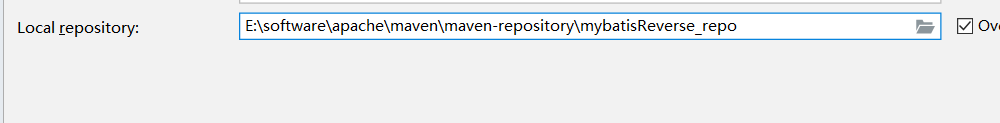
1. 配置maven的settings







1. 配置本地仓库



## **第三步，配置pon.xml**

在pom.xml中添加如下依赖：

1. mybatis-generator-maven-plugin插件
2. mybatis
3. log4j
4. mybatis-generator
5. Mysql

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.tools.mybatis</**groupId**>  
 <**artifactId**>MyBatisReverse</**artifactId**>  
 <**version**>1.0-SNAPSHOT</**version**>  
  
 <**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**>1.3.6</**version**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
 *<!-- 添加相关依赖-->* <**dependencies**>  
 <**dependency**>  
 <**groupId**>log4j</**groupId**>  
 <**artifactId**>log4j</**artifactId**>  
 <**version**>1.2.17</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.mybatis</**groupId**>  
 <**artifactId**>mybatis</**artifactId**>  
 <**version**>3.2.3</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
 <**version**>6.0.6</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.mybatis.generator</**groupId**>  
 <**artifactId**>mybatis-generator-core</**artifactId**>  
 <**version**>1.3.6</**version**>  
 </**dependency**>  
 </**dependencies**>  
  
</**project**>

## **第三步，创建generationConfig.xml文件并配置内容**

*<?***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"*>****<!-- Generator 配置文件 -->  
<!-- Tapan 2019年5月7日 16:18:08 -->*<**generatorConfiguration**>  
 <**context id="mybatisGenerator" targetRuntime="MyBatis3"**>  
 <**commentGenerator**>  
 *<!-- 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="com.mysql.cj.jdbc.Driver"  
 connectionURL="jdbc:mysql://localhost:3306/数据库名称?  
 serverTimezone=CTT&amp;useUnicode=true&amp;characterEncoding=utf-8&amp;allowMultiQueries=true" userId="root"  
 password="密码"**>  
 </**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="false"** />  
 </**javaTypeResolver**>  
  
 *<!-- targetProject:生成实体类的位置 -->* <**javaModelGenerator targetPackage="pojo"  
 targetProject=".\src"**>  
 *<!-- enableSubPackages:是否让schema作为包的后缀 -->* <**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**>  
 *<!-- targetProject:mapper映射文件生成的位置 -->* <**sqlMapGenerator targetPackage="mapper"  
 targetProject=".\src"**>  
 *<!-- 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="mapper"  
 targetProject=".\src"**>  
 *<!-- enableSubPackages:是否让schema作为包的后缀 -->* <**property name="enableSubPackages" value="false"** />  
 </**javaClientGenerator**>  
 *<!-- 指定数据库表 -->  
 <!-- 如果要生成所有表，将tableName的属性修改为“%”即可 -->* <**table tableName="表名或%"**></**table**>  
 *<!--<table tableName="orders"></table>  
 <table tableName="orderdetail"></table>  
 <table tableName="user"></table>-->  
  
 <!-- 有些表的字段需要指定java类型  
 <table schema="" tableName="">  
 <columnOverride column="" javaType="" />  
 </table> -->* </**context**>  
</**generatorConfiguration**>

## **第四步，创建Java类Generator，由官方提供**

package test;

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;

/\*\*

\* MyBatis官方提供的逆向工程Java类

\* @author Tapan

\* 2019年5月7日 16:15:47

\*/

public class Generator {

public void generator() throws Exception{

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

boolean overwrite = true;

/\*\*指向逆向工程的配置文件\*/

File configFile = new File("src/main/resources/generationConfig.xml");

ConfigurationParser parser = new ConfigurationParser(warnings);

Configuration config = parser.parseConfiguration(configFile);

DefaultShellCallback callback = new DefaultShellCallback(overwrite);

MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,

callback, warnings);

myBatisGenerator.generate(null);

}

public static void main(String[] args) throws Exception {

try {

Generator generatorSqlmap = new Generator();

generatorSqlmap.generator();

} catch (Exception e) {

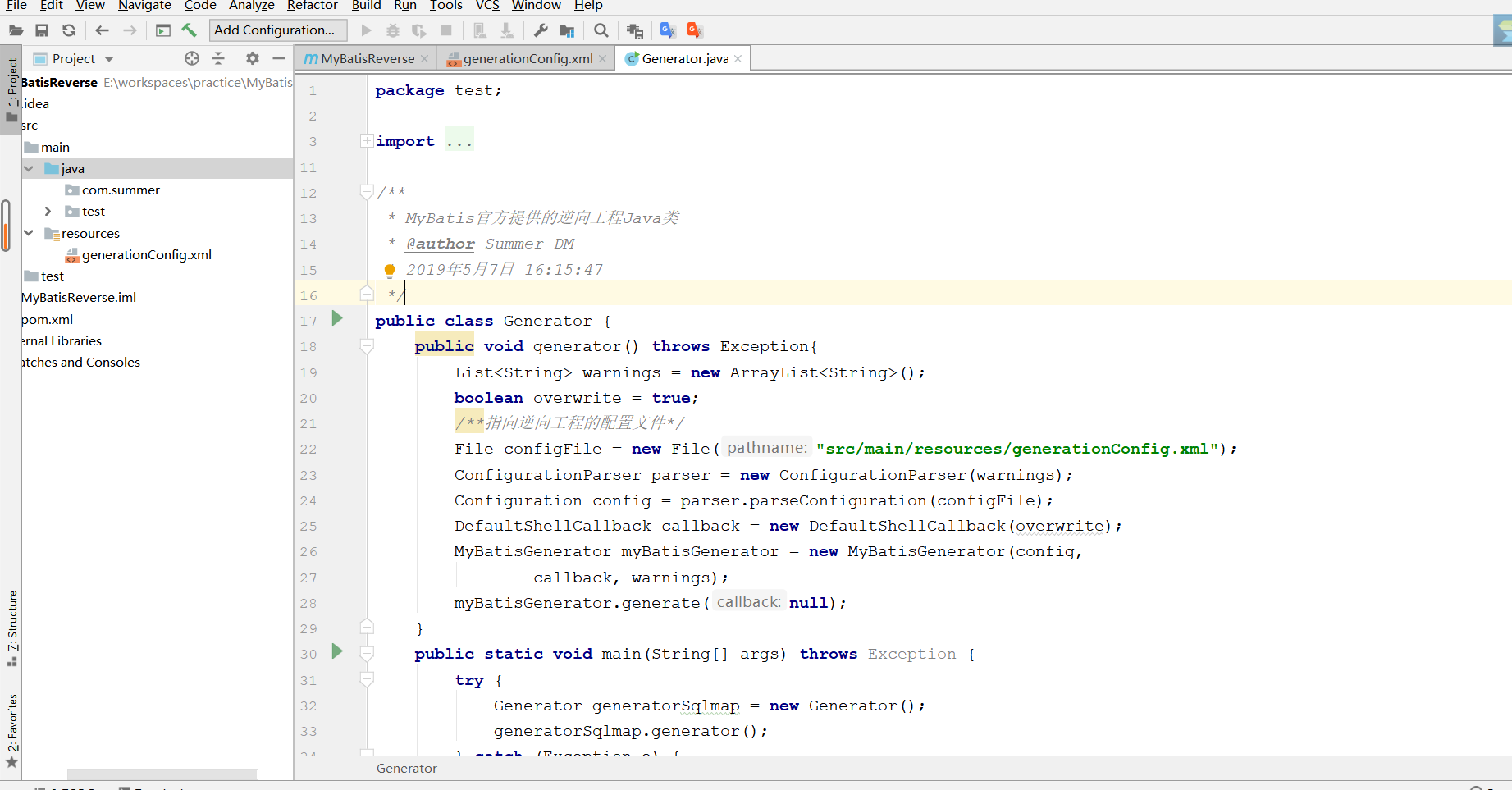
e.printStackTrace();

}

}

}

**自此，MyBatis逆向工程项目就创建完成了，现在的项目结构如图所示：**



**接着，你只要运行Java类Generator中的Main方法，即可得到由MyBatis为你生成的实体类以及接口，Mapper文件等！运行完之后是不是发现多了两个很可爱的文件夹？哈哈，很神奇呢！**

