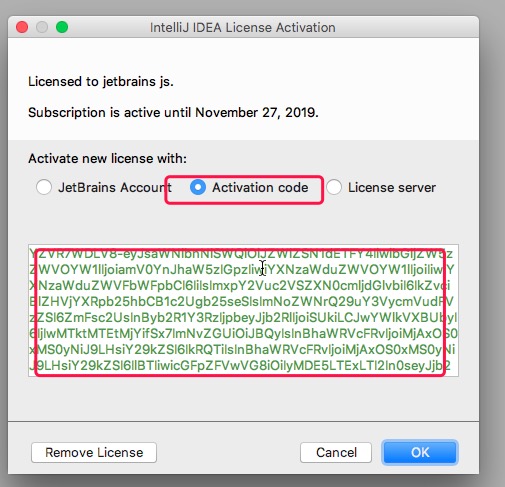
### 1.Maven安装配置文档

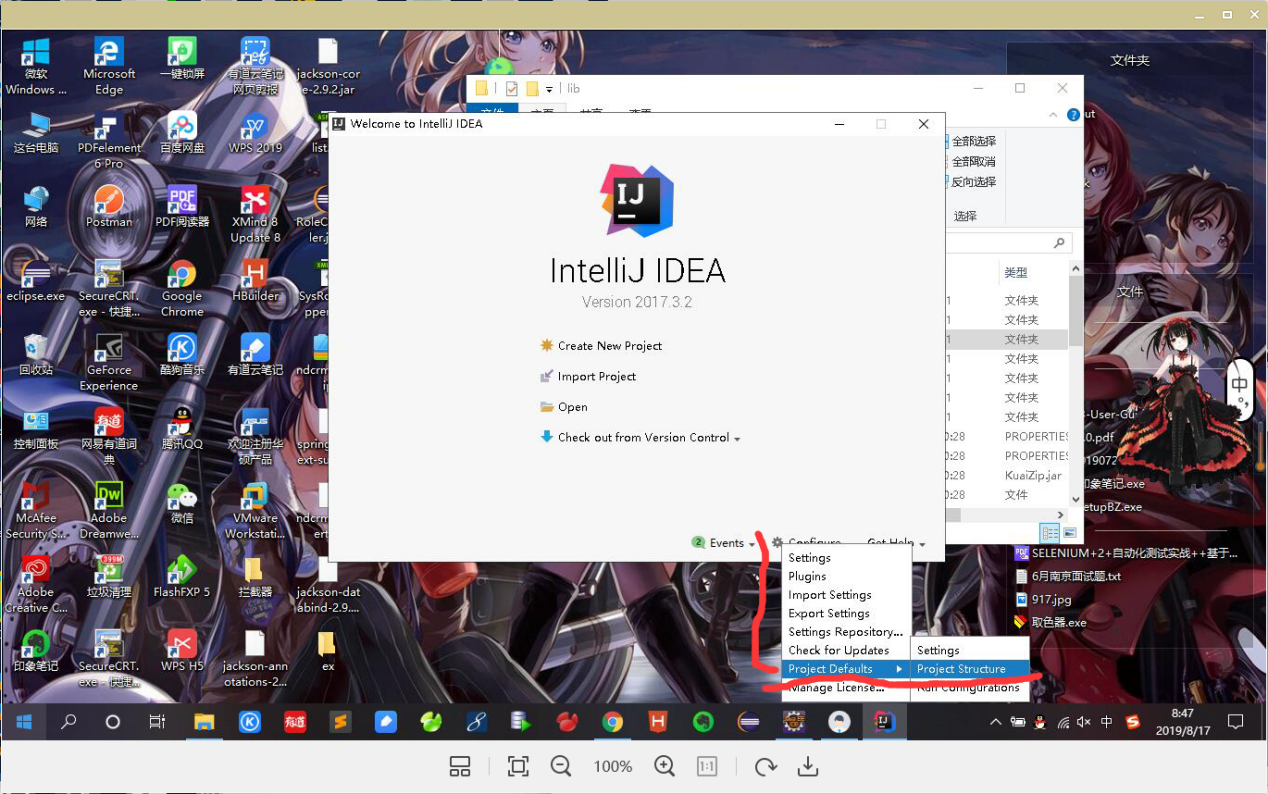
#### 首先去下载软件，在官网或者资源

#### 下载后点击.exe应用，然后一直点next.

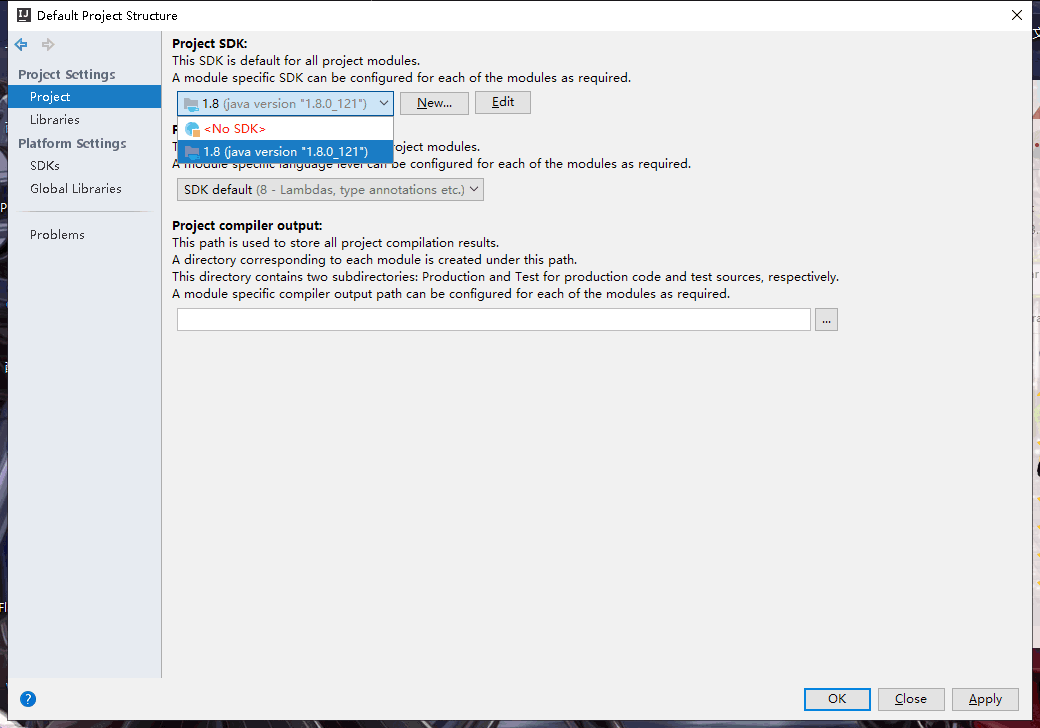
#### 出现如下页面，点击中间activation code选项（注册），下面填入注册码，注册码自己网上百度



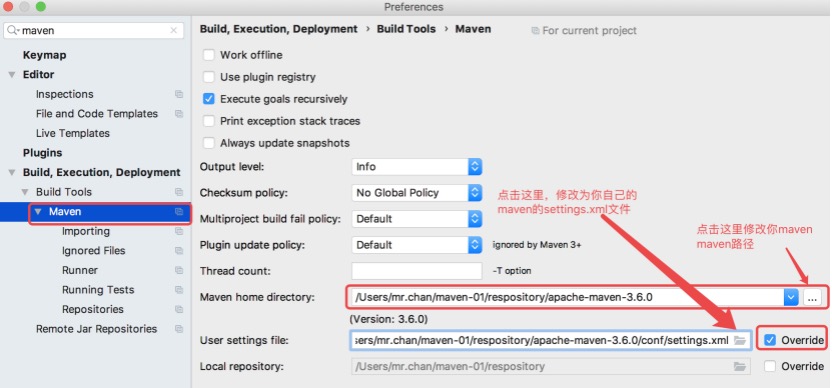
#### 然后点击在桌面上的快键图标，点击运行，出现如下图片，点击右下角的工具



#### 点进去后这个主要配置jdk,点击SDKs在中间块有个点+号，点击添加自己jdk按照的路径，（此处和配置jdk环境类似），点击应用。然后点击project,在右侧有个project SDK,此处下拉刚刚添加的jdk,然后点击应用

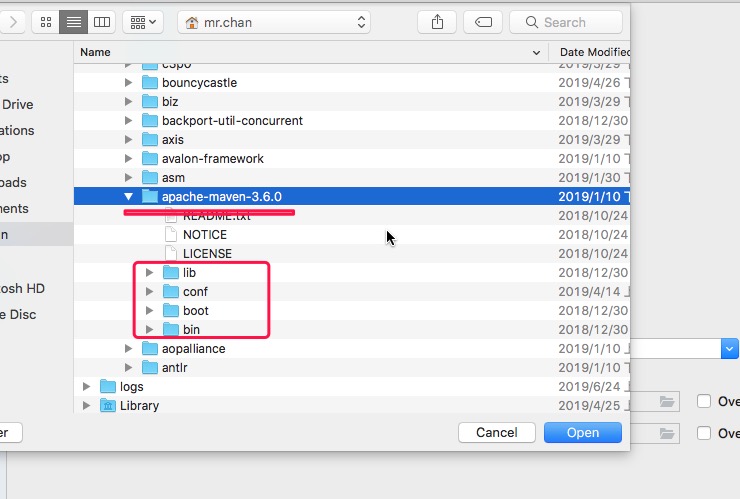


#### 到此就开始配置maven，同样点击开始页面的上一个settings,去配置maven,搜查maven,出现如下图

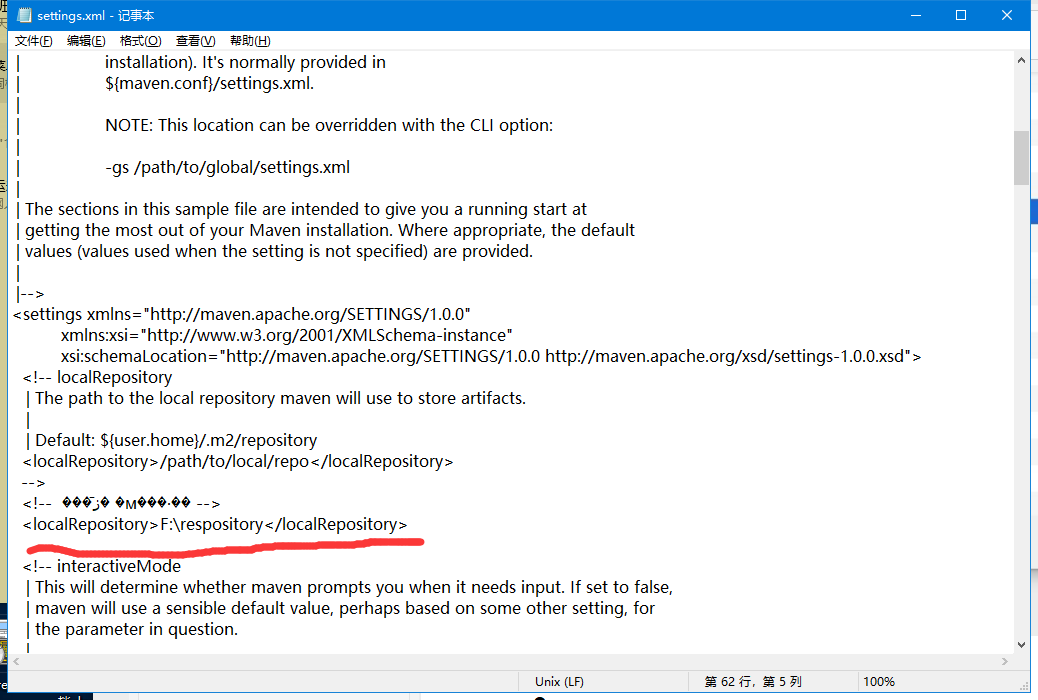


#### 然后选择Maven home directory,点击后面的按钮，选择你修改的maven路径，（在此之前先创建一个respository文件夹放入你下载好的maven文件）

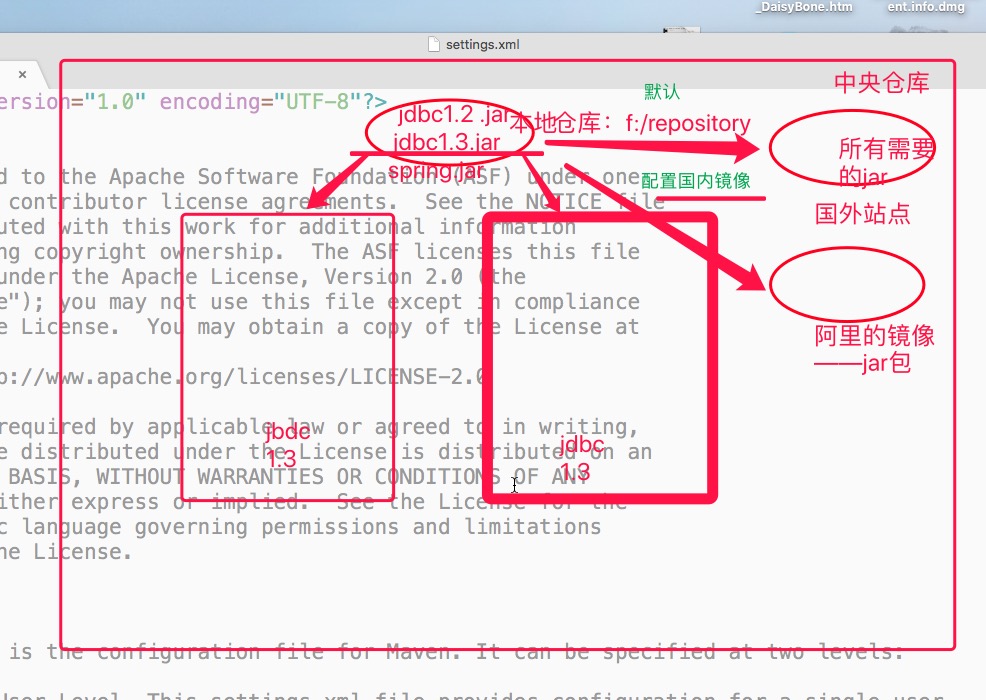
#### 找到你的settings.xml文件他一般在你下载好的conf文件夹里面



#### 找到localRespository标签，修改你的路径为你创建respository下的路径



#### 添加阿里的镜像，也可添加其他的，其构造如图所示

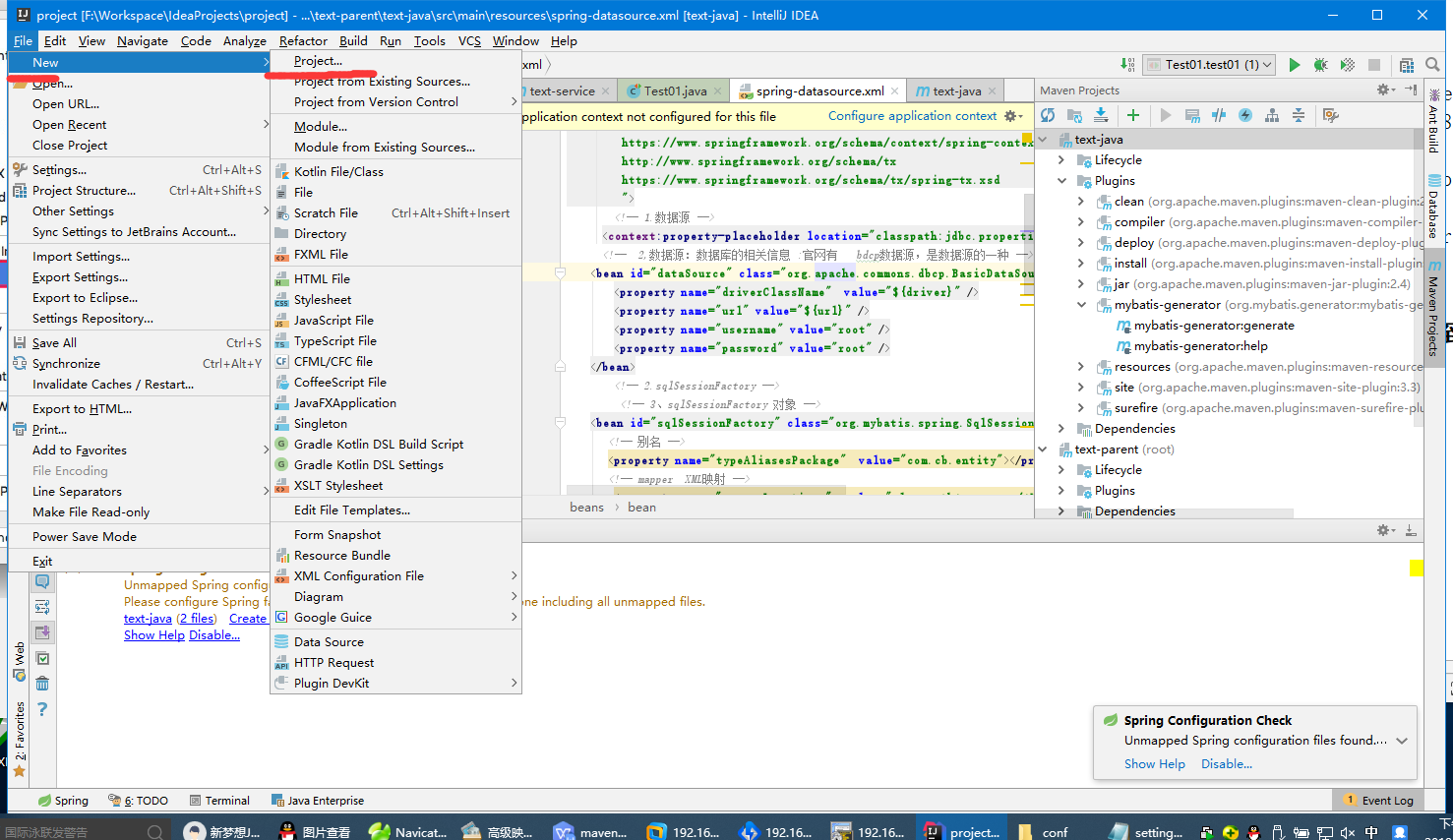


#### 找到mirrors标签，把下面代码复制进行

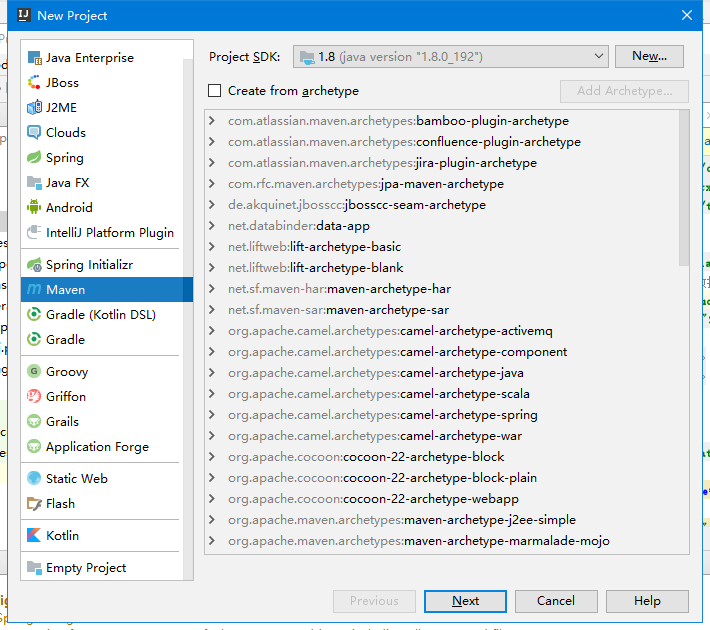
<!-- 阿里云仓库-->  
    <mirror>  
      <id>alimaven</id>  
      <mirrorOf>central</mirrorOf>  
      <name>aliyun maven</name>  
      <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>  
    </mirror>  
  
    <!-- 中央仓库1 -->  
    <mirror>  
      <id>repo1</id>  
      <mirrorOf>central</mirrorOf>  
      <name>Human Readable Name for this Mirror.</name>  
      <url>http://repo1.maven.org/maven2/</url>  
    </mirror>  
  
    <!-- 中央仓库2 -->  
    <mirror>  
      <id>repo2</id>  
      <mirrorOf>central</mirrorOf>  
      <name>Human Readable Name for this Mirror.</name>  
      <url>http://repo2.maven.org/maven2/</url>  
    </mirror>  
    <mirror>  
            <id>CN</id>  
            <name>OSChina Central</name>  
            <url>http://maven.oschina.net/content/groups/public/</url>  
            <mirrorOf>central</mirrorOf>  
    </mirror>  
  
     <mirror>  
            <id>nexus</id>  
            <name>internal nexus repository</name>  
            <!-- <url>http://192.168.1.100:8081/nexus/content/groups/public/</url>-->  
            <url>http://repo.maven.apache.org/maven2</url>  
            <mirrorOf>central</mirrorOf>  
        </mirror>

#### 点击首页的Create New Project进去。遇到弹窗。可以直接点击关闭按钮。

#### 点击左上角的file创建新的项目



#### 出现如下图所示，点击软件自动默认选中maven,其中不会勾选Create from archetype,不选中则为java模式，选中则为web模式。（此处先讲java模式）



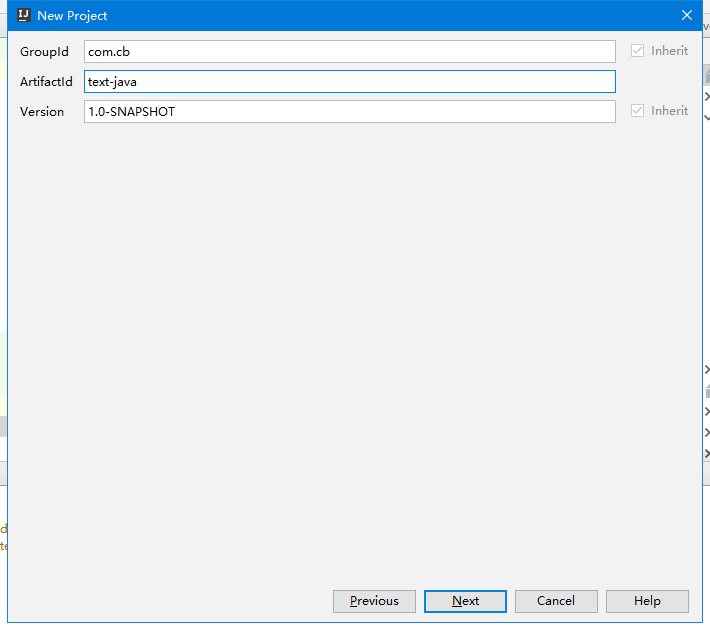
#### 点击next进入如下图片

##### 其中GroupId为定义当前Maven项目隶属项目（类似于以前的前缀com.cb）

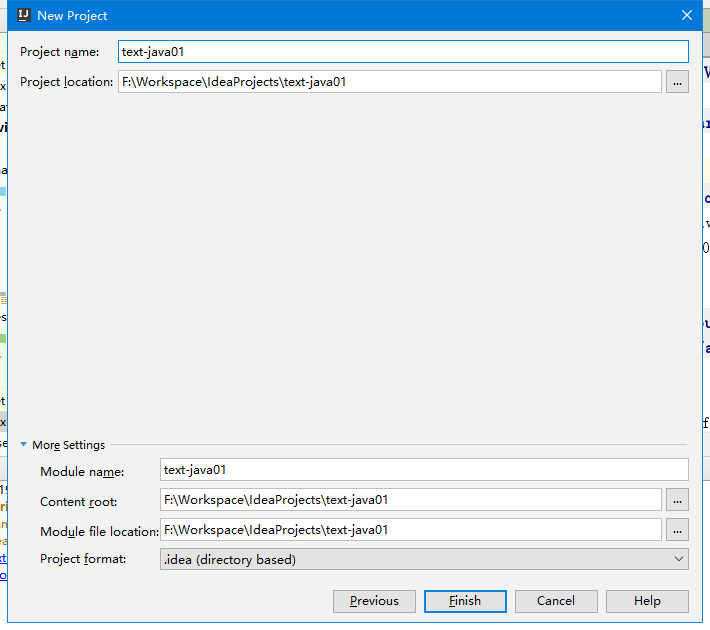
##### artifactId:定义实际项目中的一个模块(test-java)

##### Version：定义当前项目的当前版本不需要改

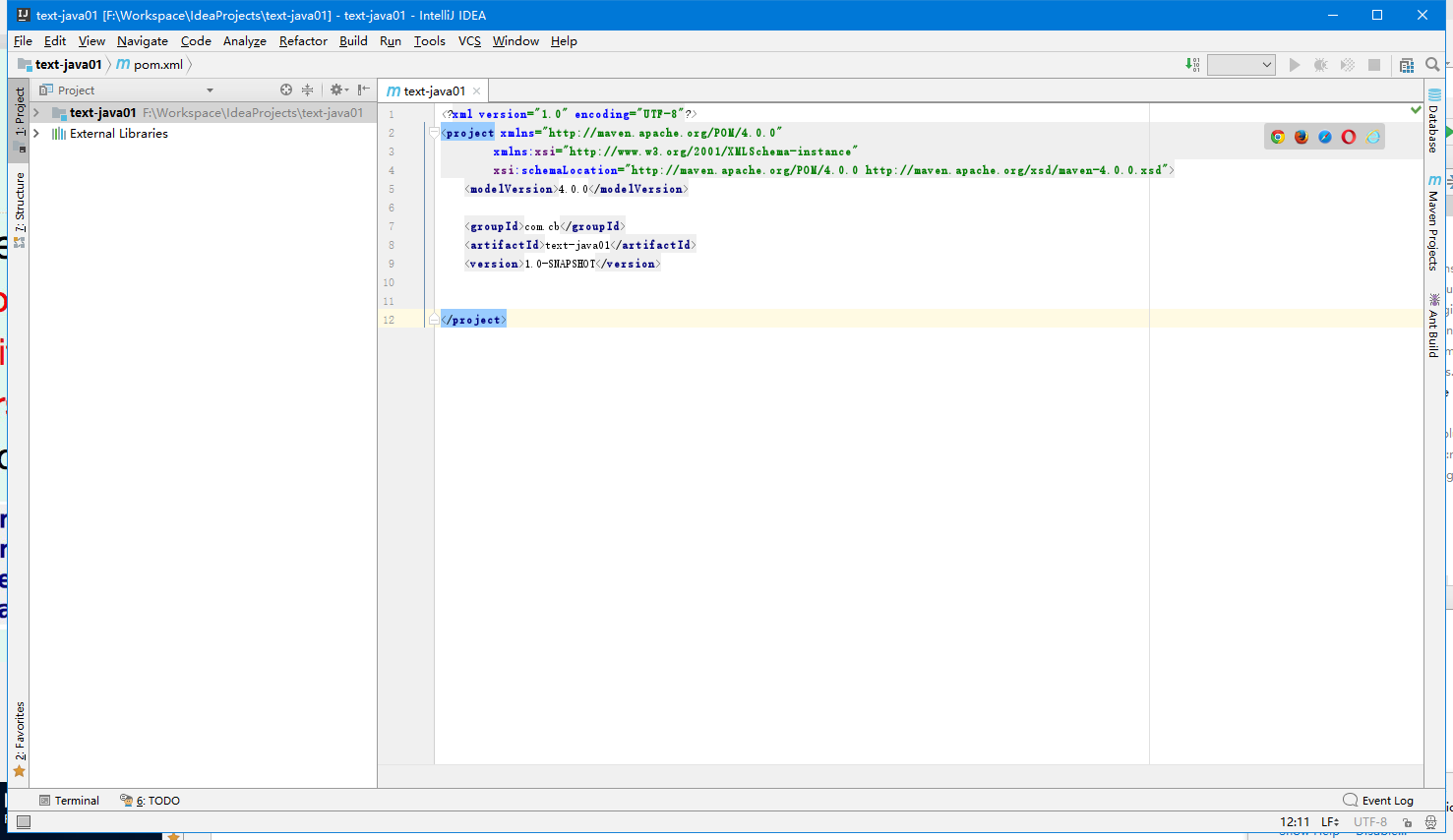
#### 如下图所示

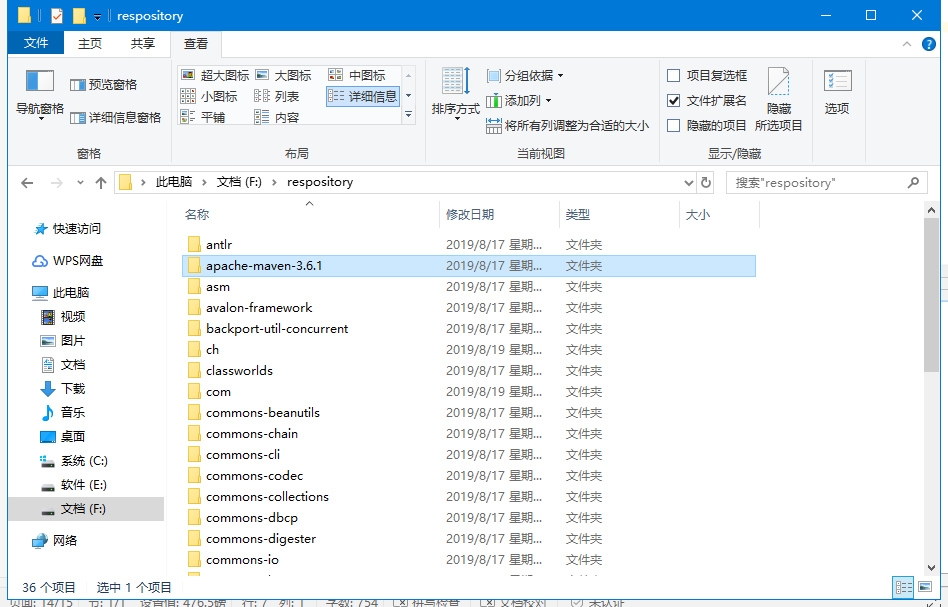


#### 再出现如下图所示，最好在其中加和横杆如下所示



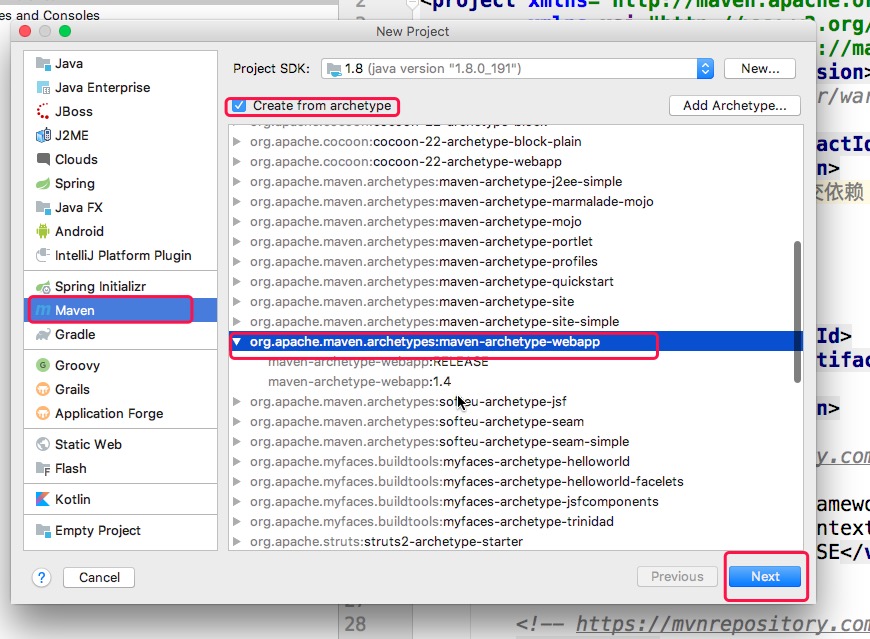
#### 出现如下图所示,在右下角会有一个小弹窗，点击弹窗中右的链接，（意思就是自动加载其中的配置，及你加写的阿里镜像里的）成功后在你的respository文件夹下会出现如下图中文件（只是截取了一部分，一般如果出错是不会出现的，但是出现了就是OK了）





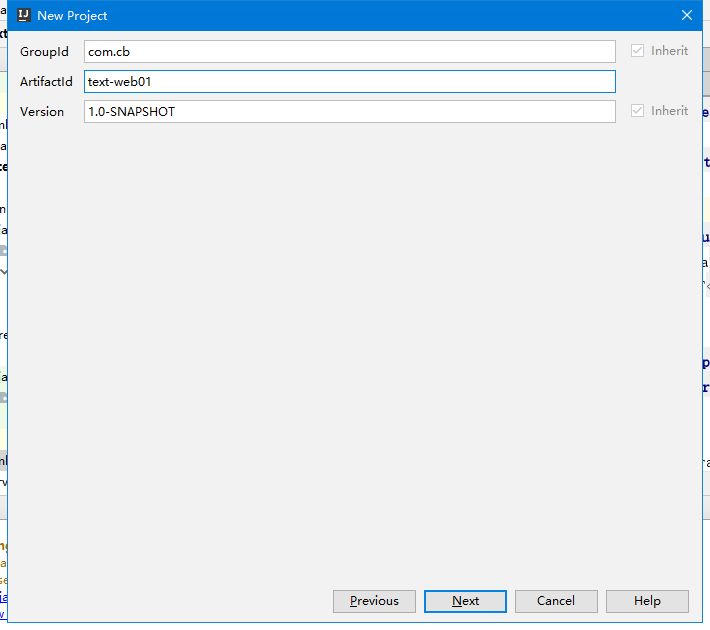
### 2.idea配置文档

#### 此处以创建web模式为例

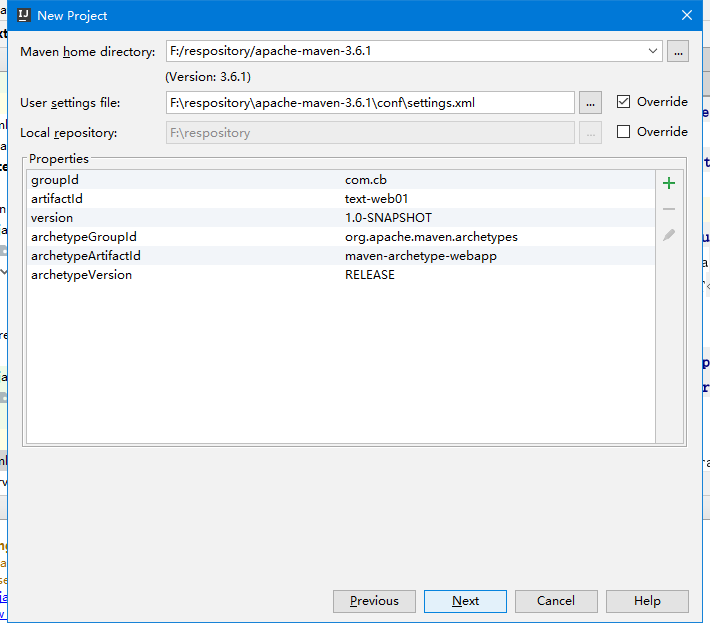


勾选create from archetype 然后选择如图的选择，再点击next.

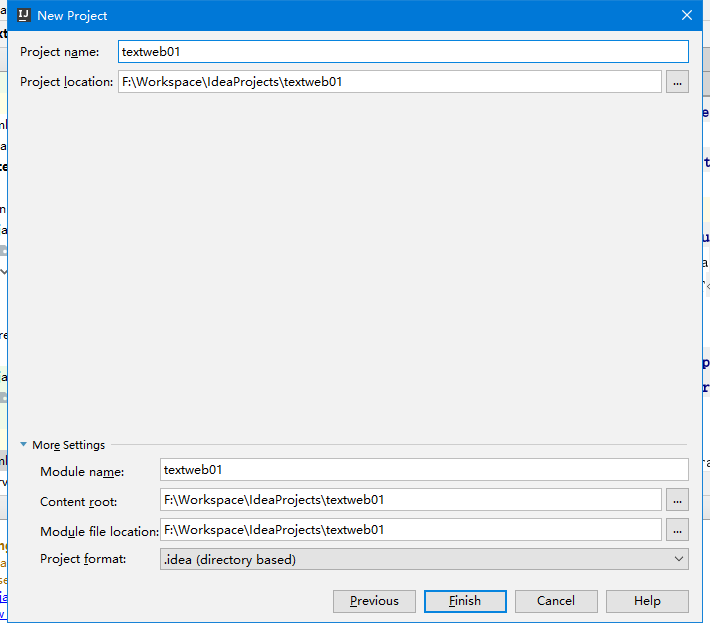
#### 填写如图所示



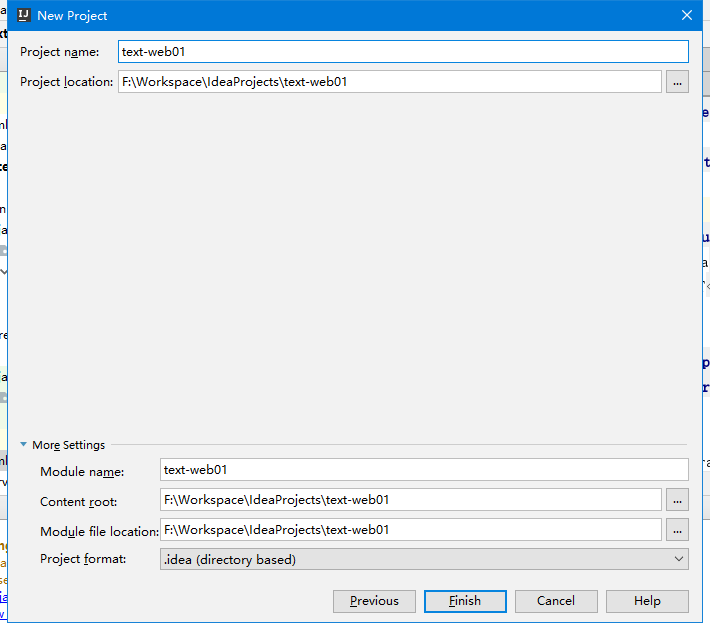
#### 再点击next出现如下图所示，这里一般路径是对的，然后继续点击next



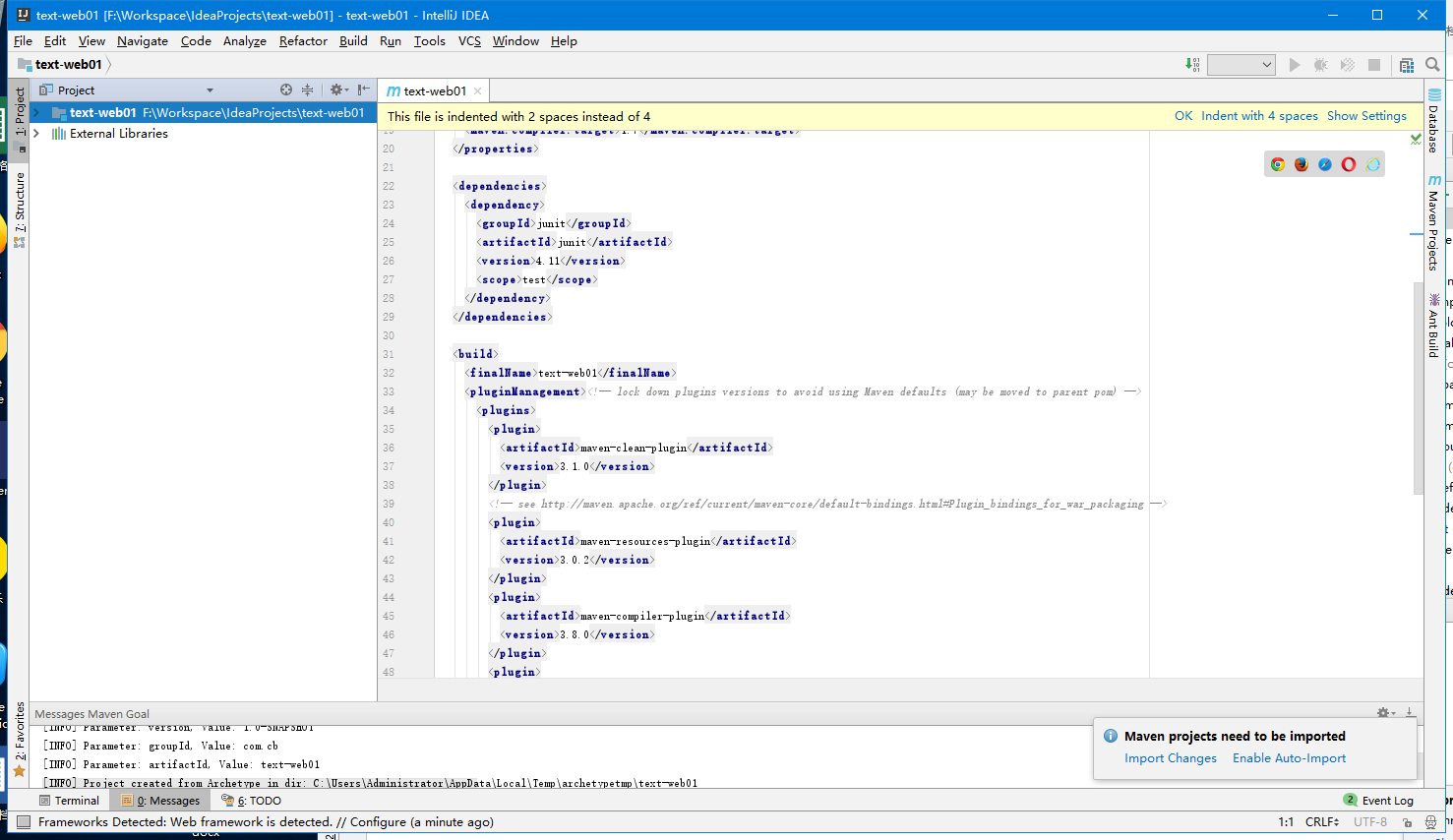
#### 如下图所示



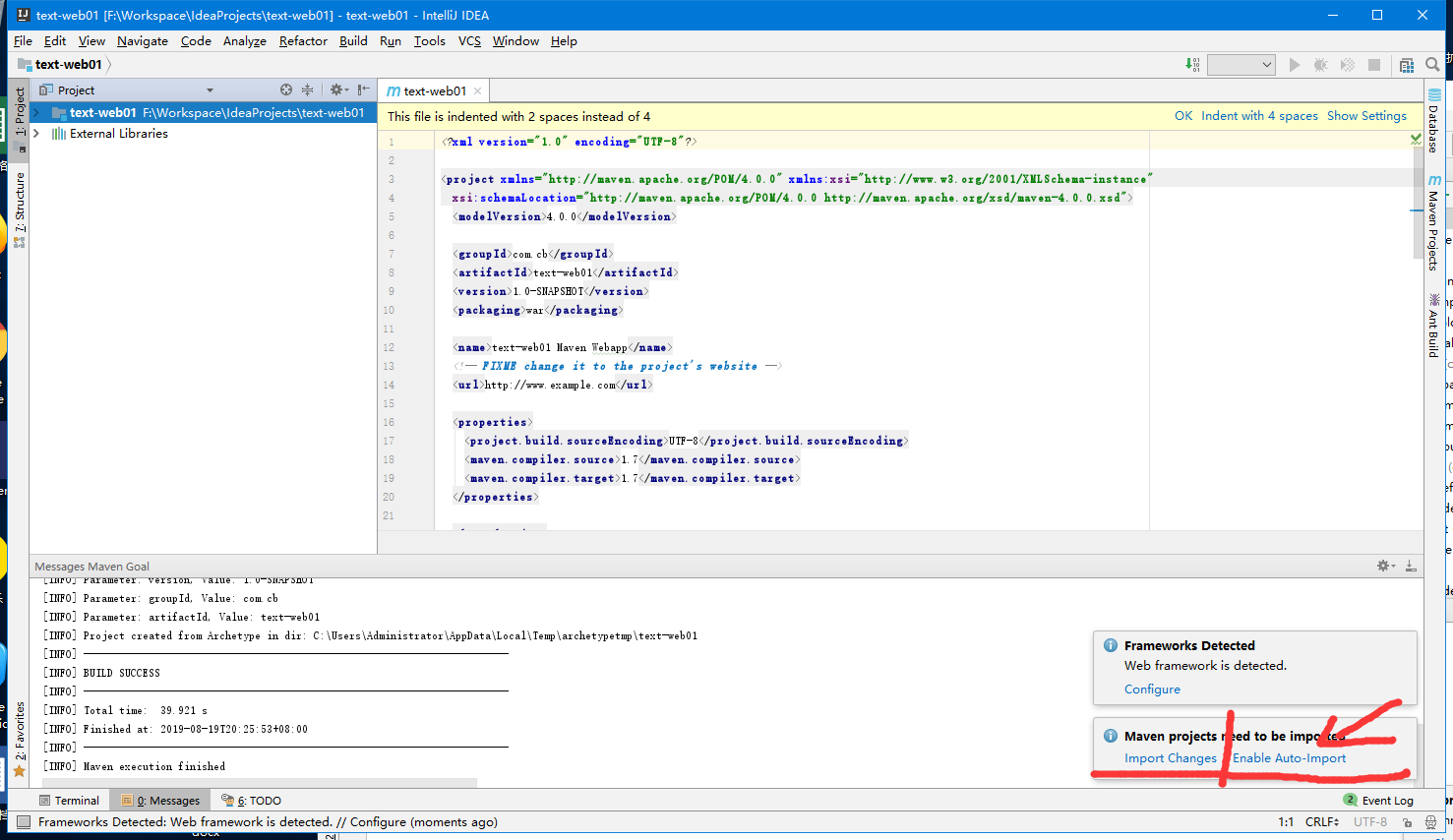
#### 再改变



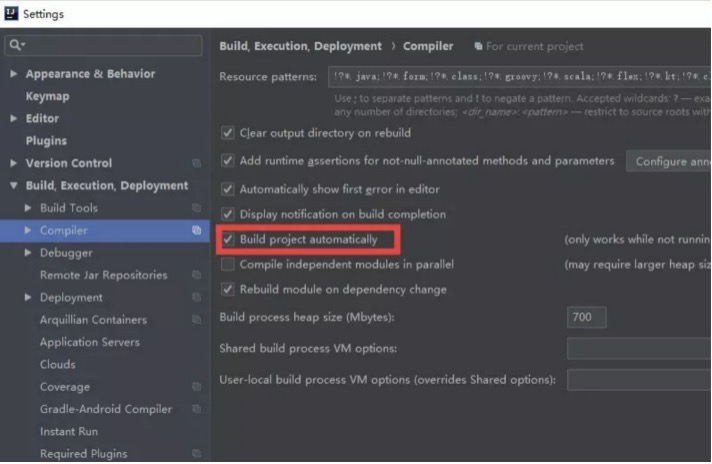
#### 出现如图所示就正确了



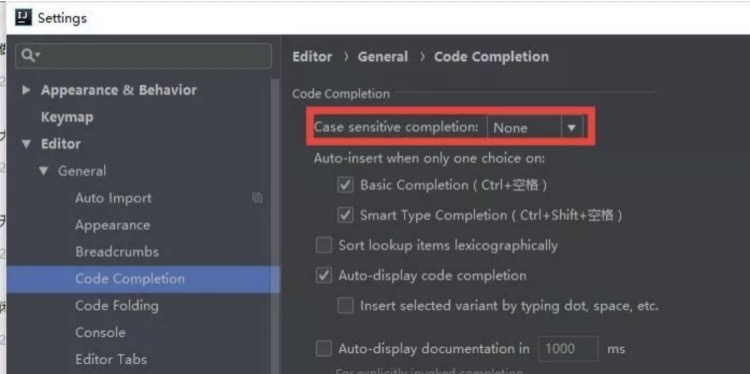
#### 点击箭头的链接，自动导入



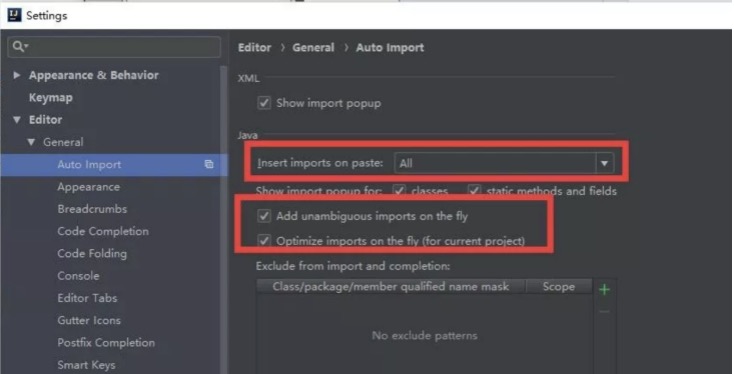
#### 8.2.1.3.1.在Eclipse中自动编译开关是开着的，如下所示



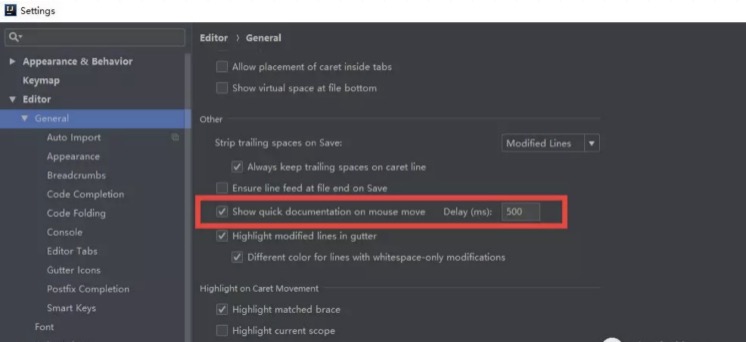
#### 9.2.1.3.2.忽略大小写开关



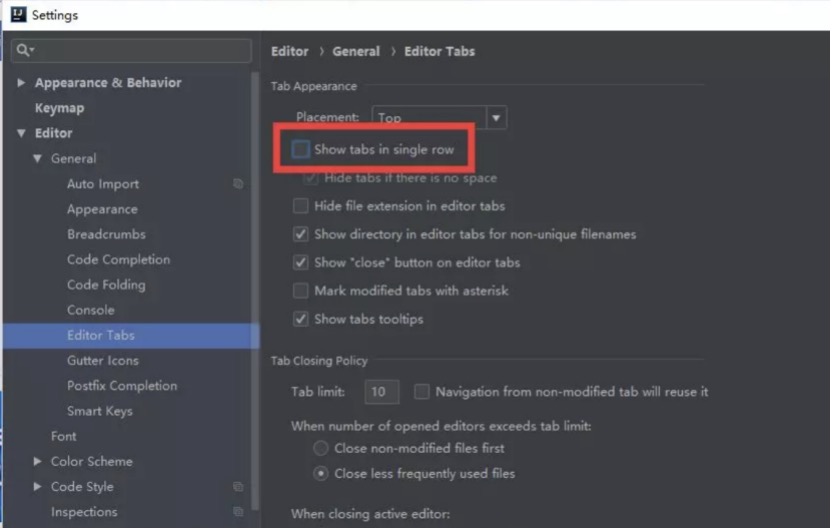
#### 10.2.1.3.3.智能导包开关



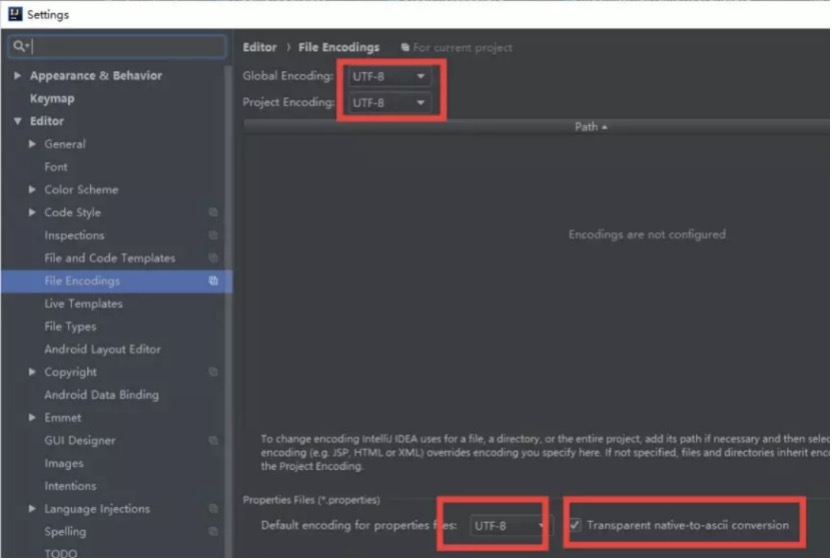
#### 11.2.1.3.4.悬浮提示开关



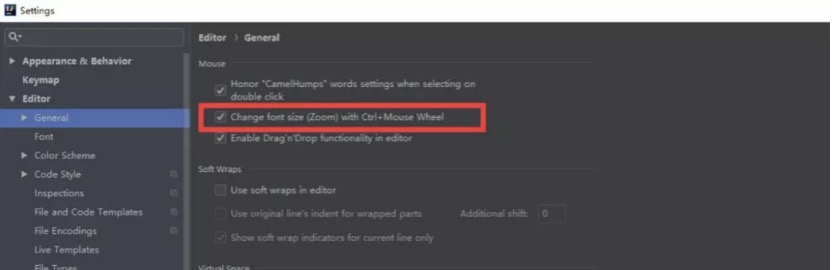
#### 12.2.1.3.5.取消单行显示tabs的操作



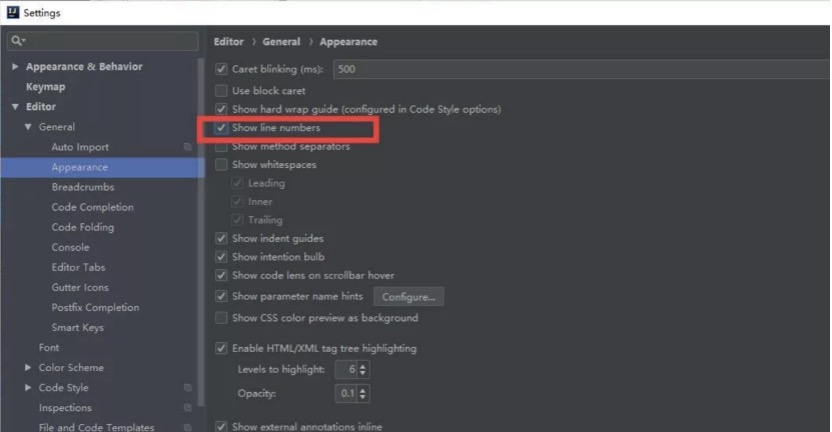
#### 13.2.1.3.6.项目文件编码



#### 14.2.1.3.7.滚轴修改字体大小



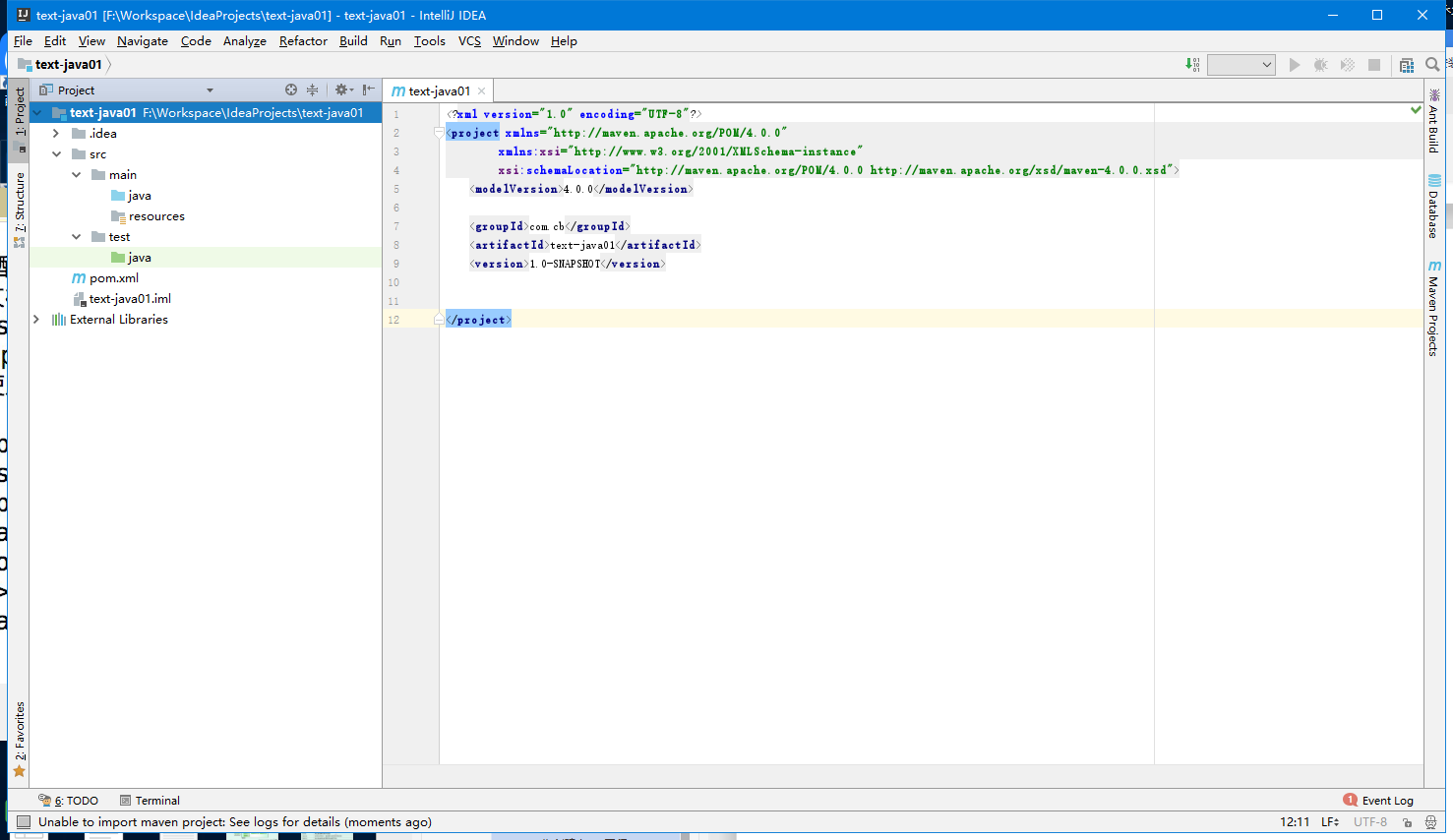
#### 15.2.1.3.8.设置行号显示



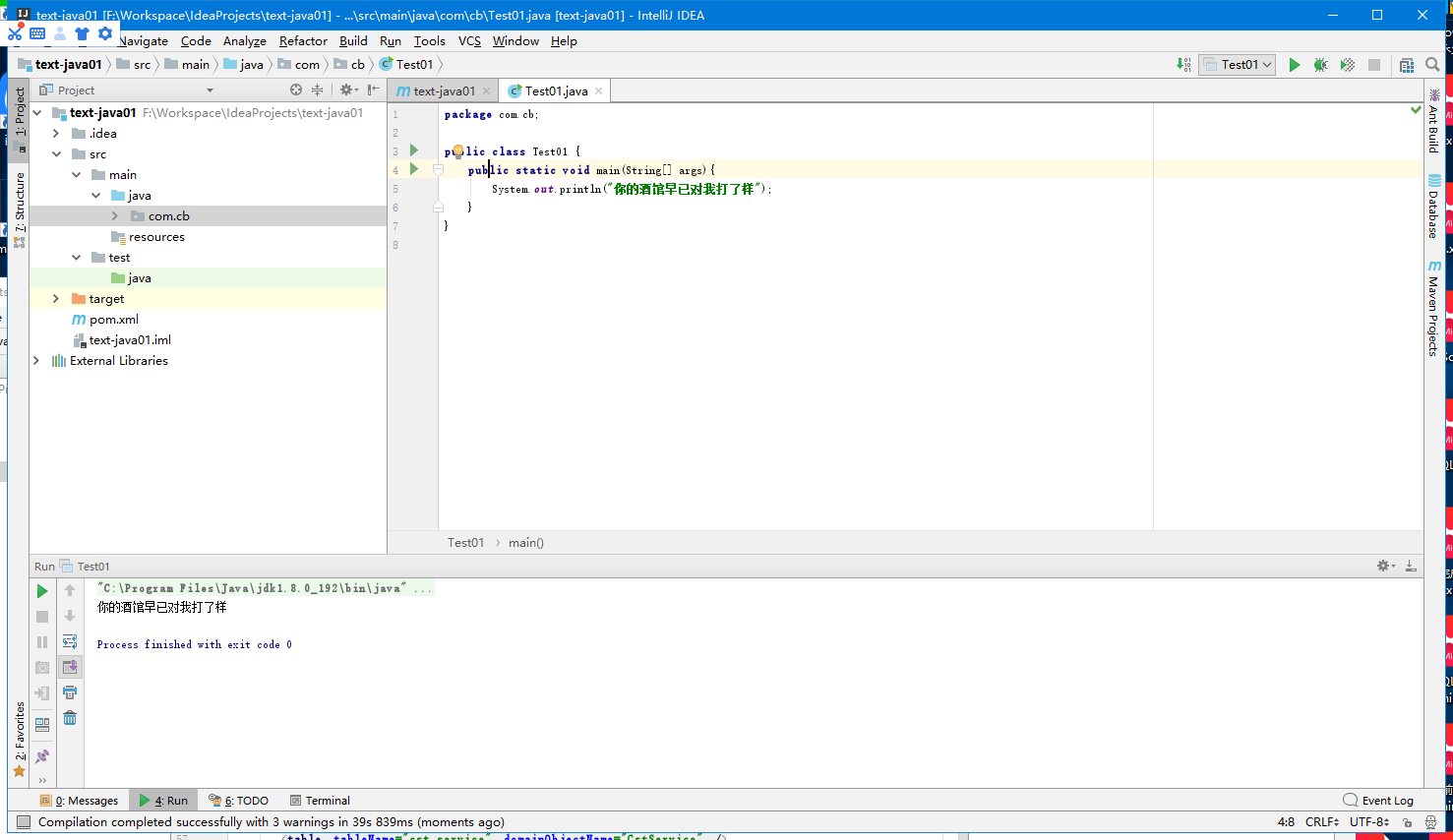
### idea搭建工程

#### 3.1.idea搭建jar工程

##### 3.1.1先创建java工程，点开各级目录如下图所示

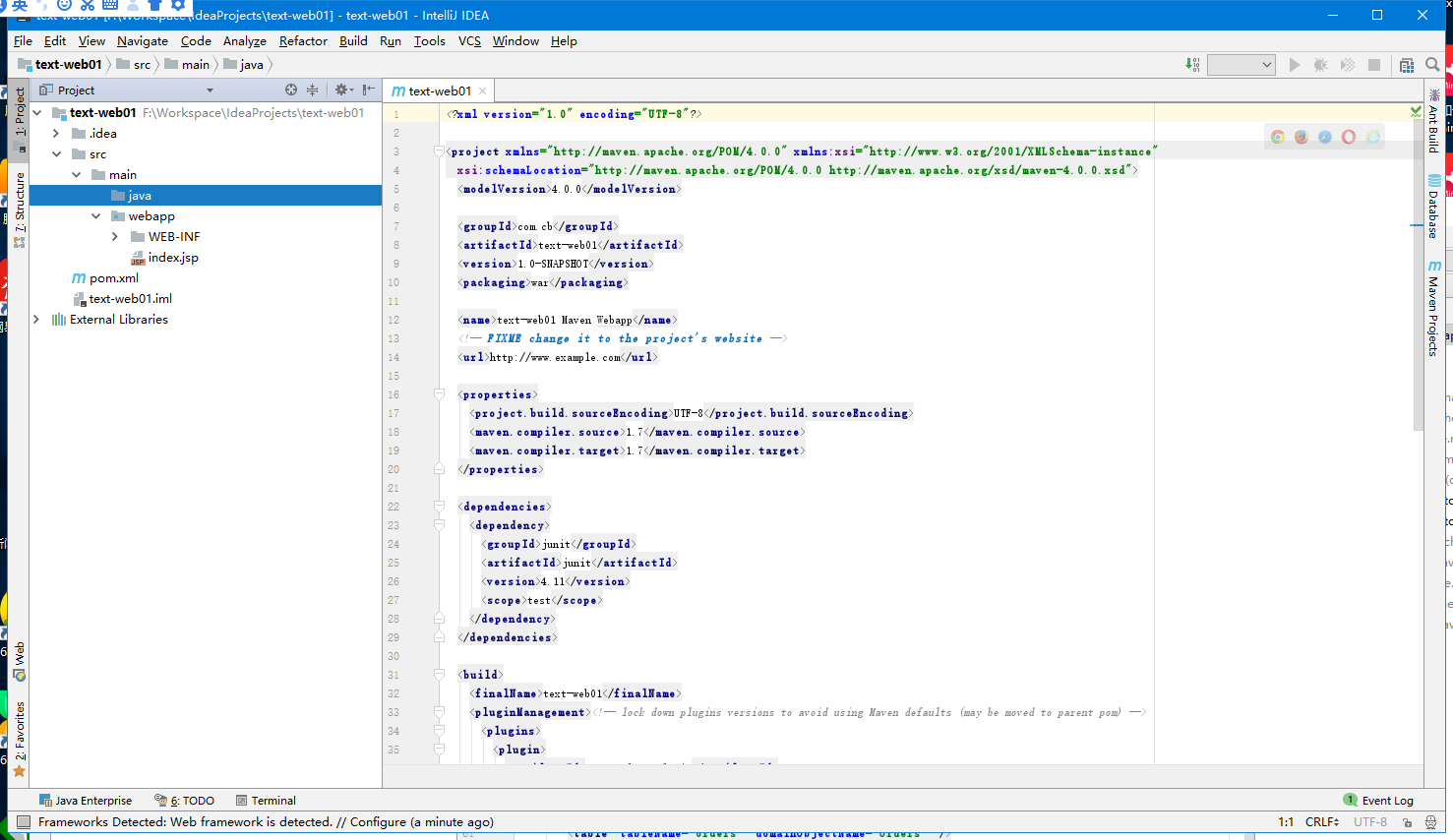


##### 3.1.2.点击src创建包和java格式一样，然后创建.java文件测试如下图所示

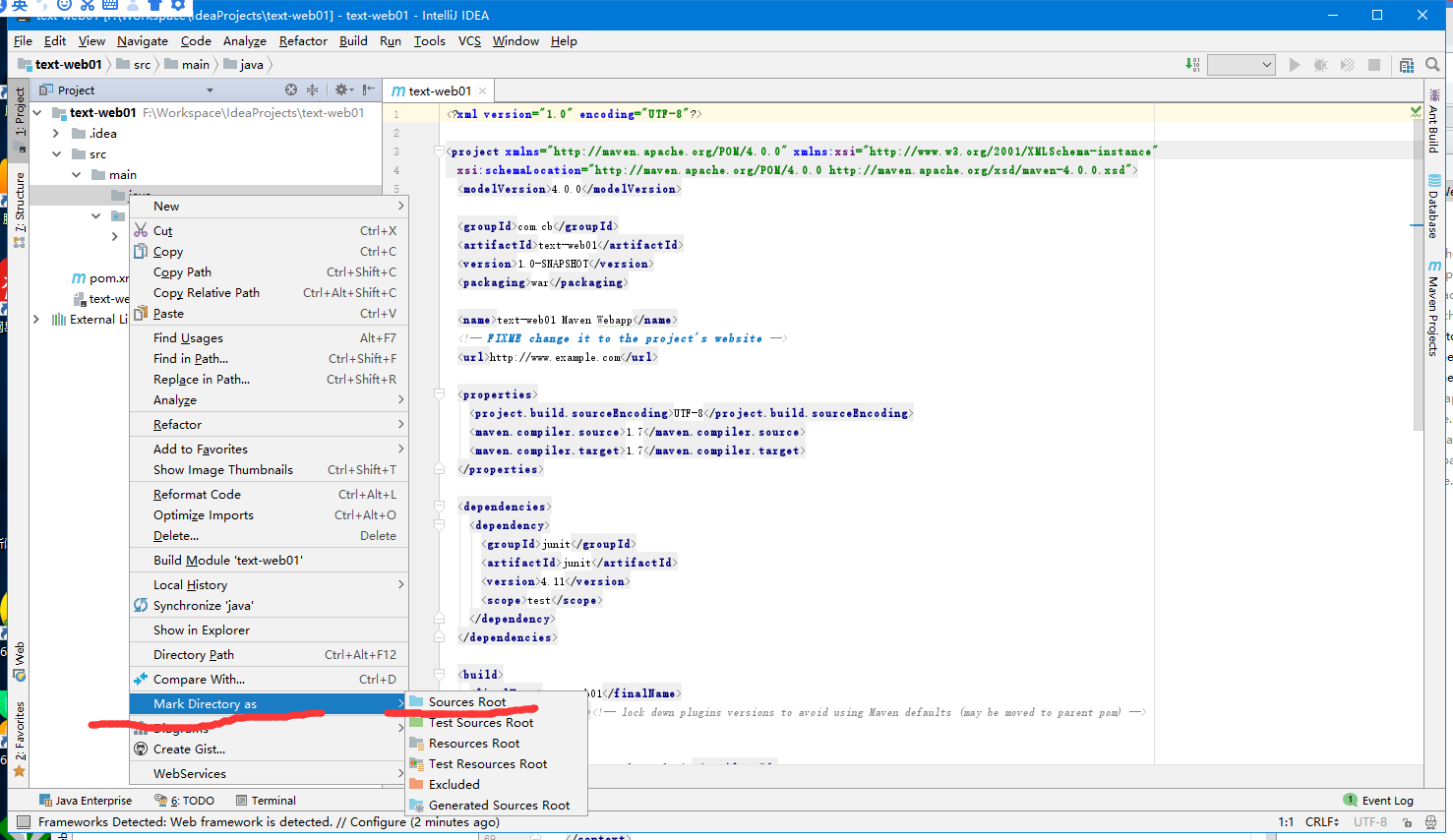


##### 3.1.3.配置SSM的dao层

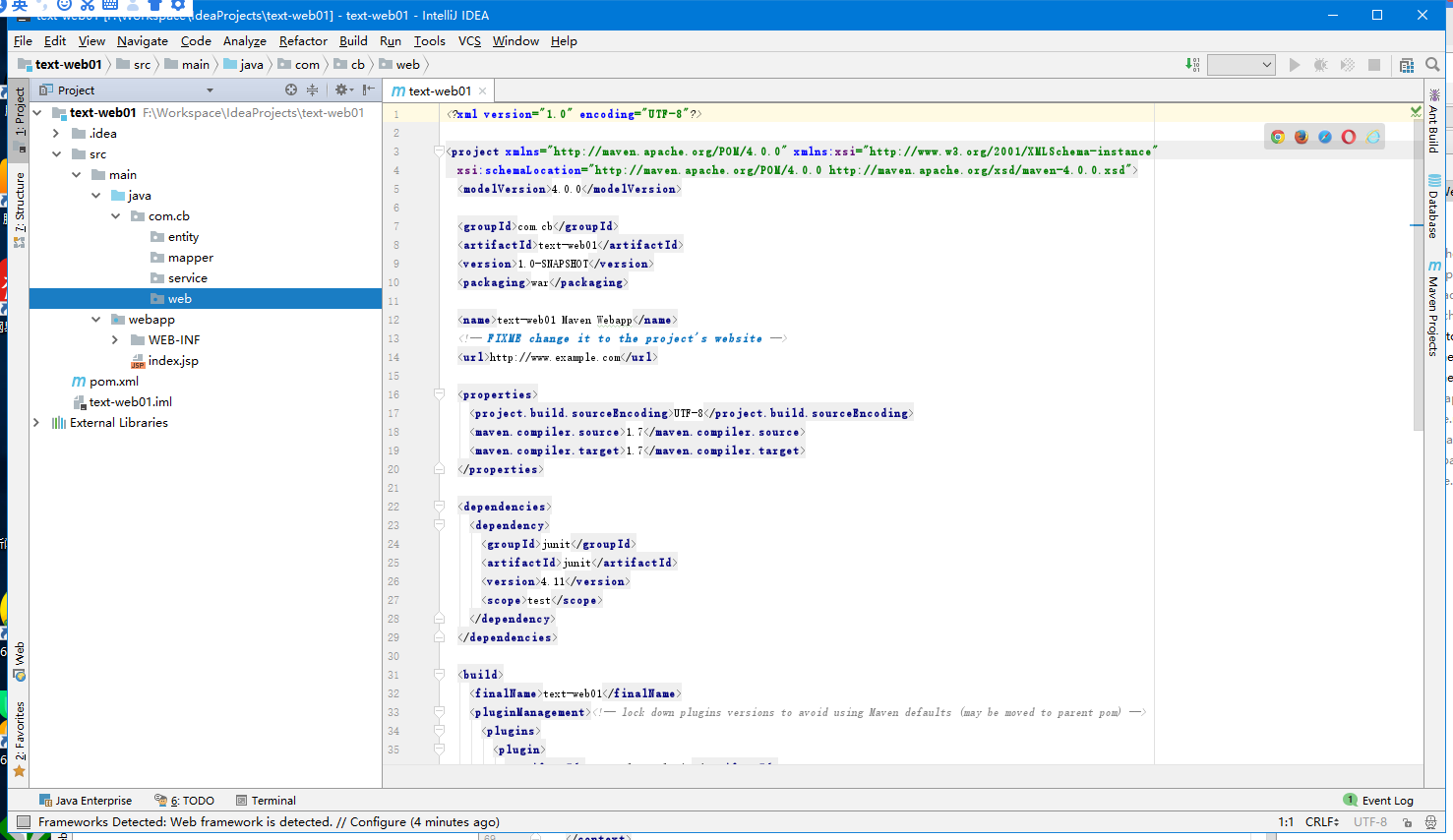
###### 1.创建web工程，然后在main下创建java文件，然后改变次java文件的类型



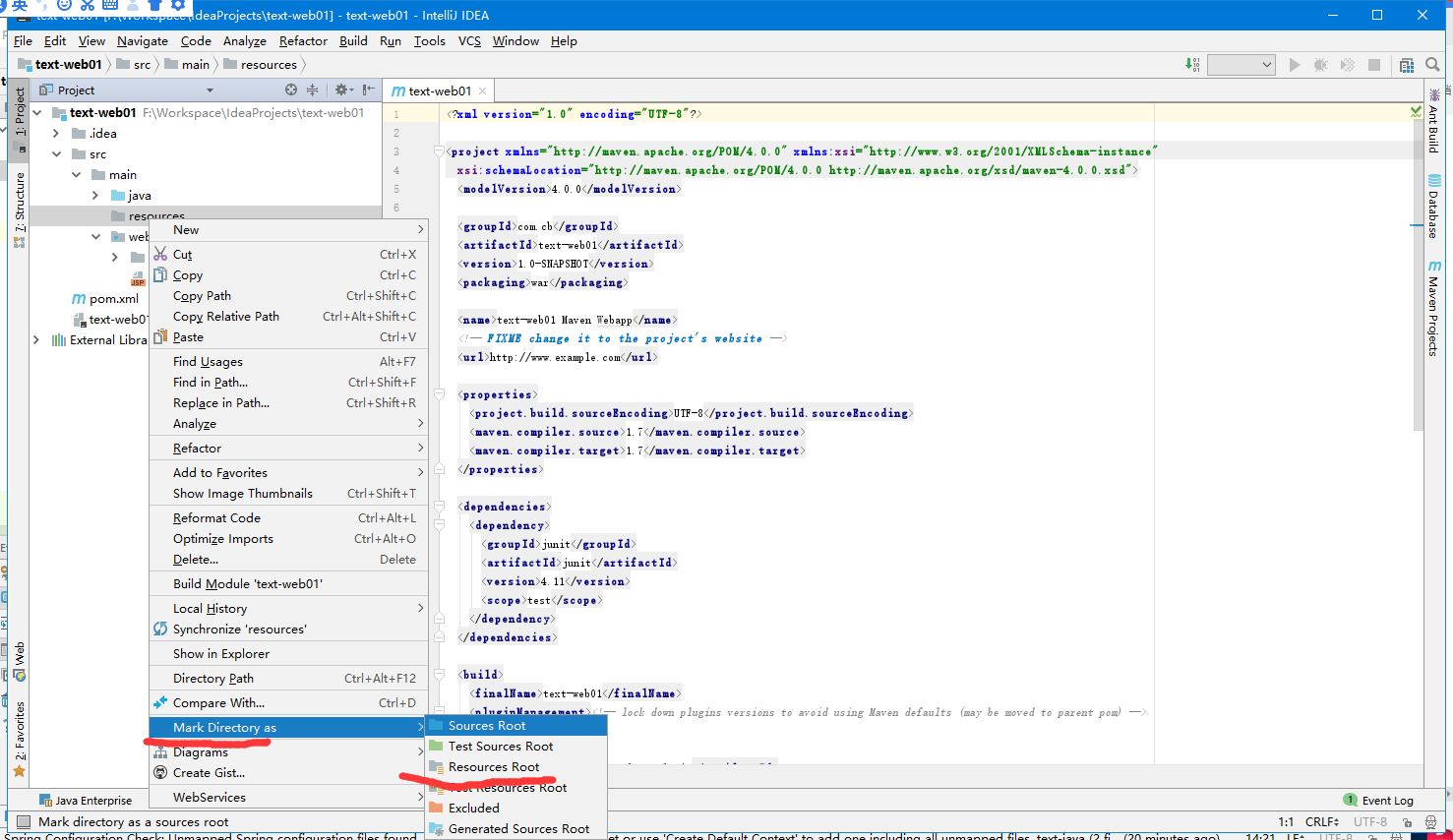
改变后如下所示



如上图所示操作，创建文件位置格式如下图所示

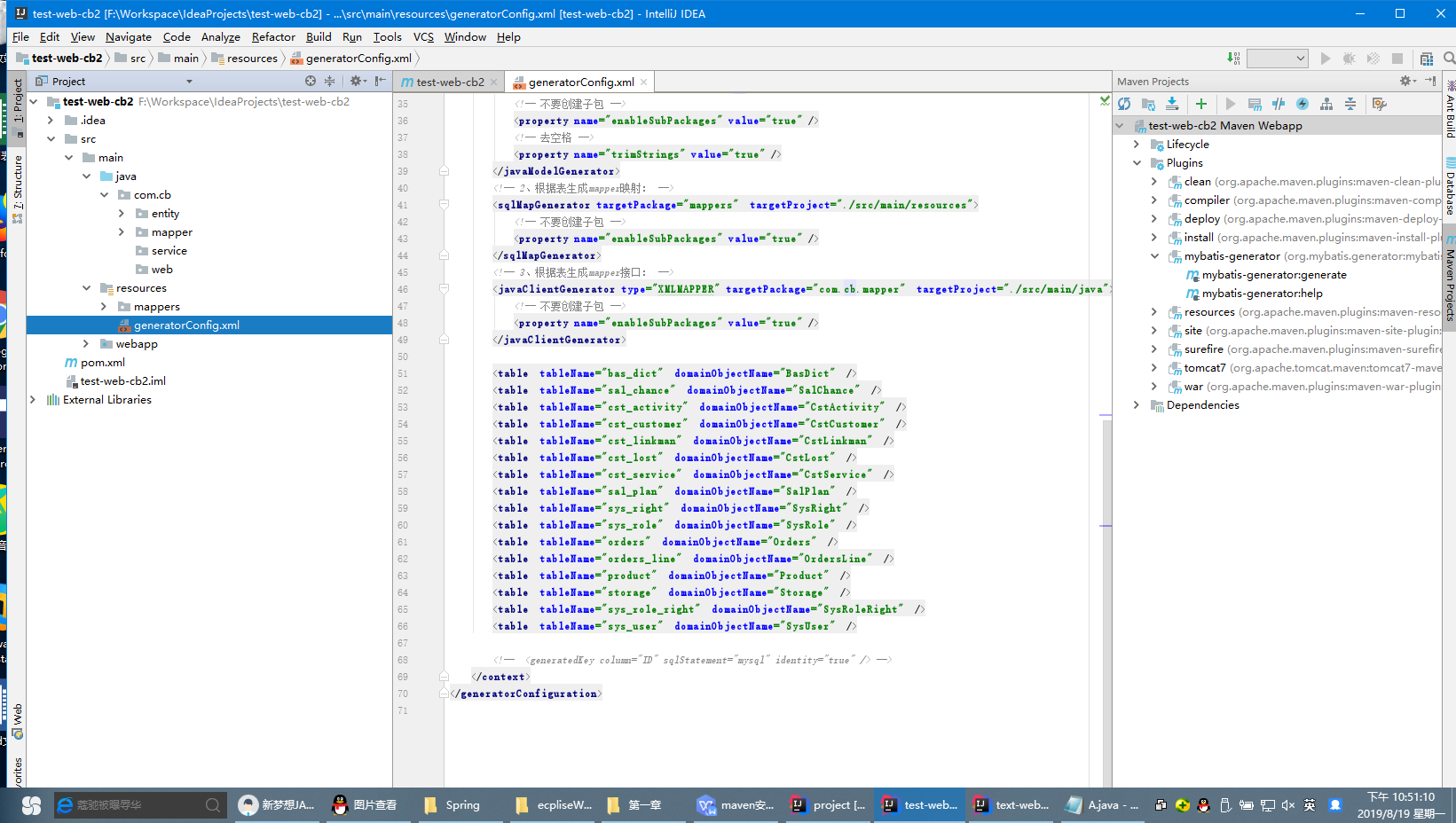


同理创建resources,按如下图操作



###### 添加MBG.xml配置

1. 这里的MBG.xml文件和ecilpse的不同，是放在resources文件夹中，取名为generatorConfig.xml



1. generatorConfig.xml中的配置和eclpise中的类似但是有一些生产文件位置有变化，*本地数据库驱动程序jar包的全路径发生变化*



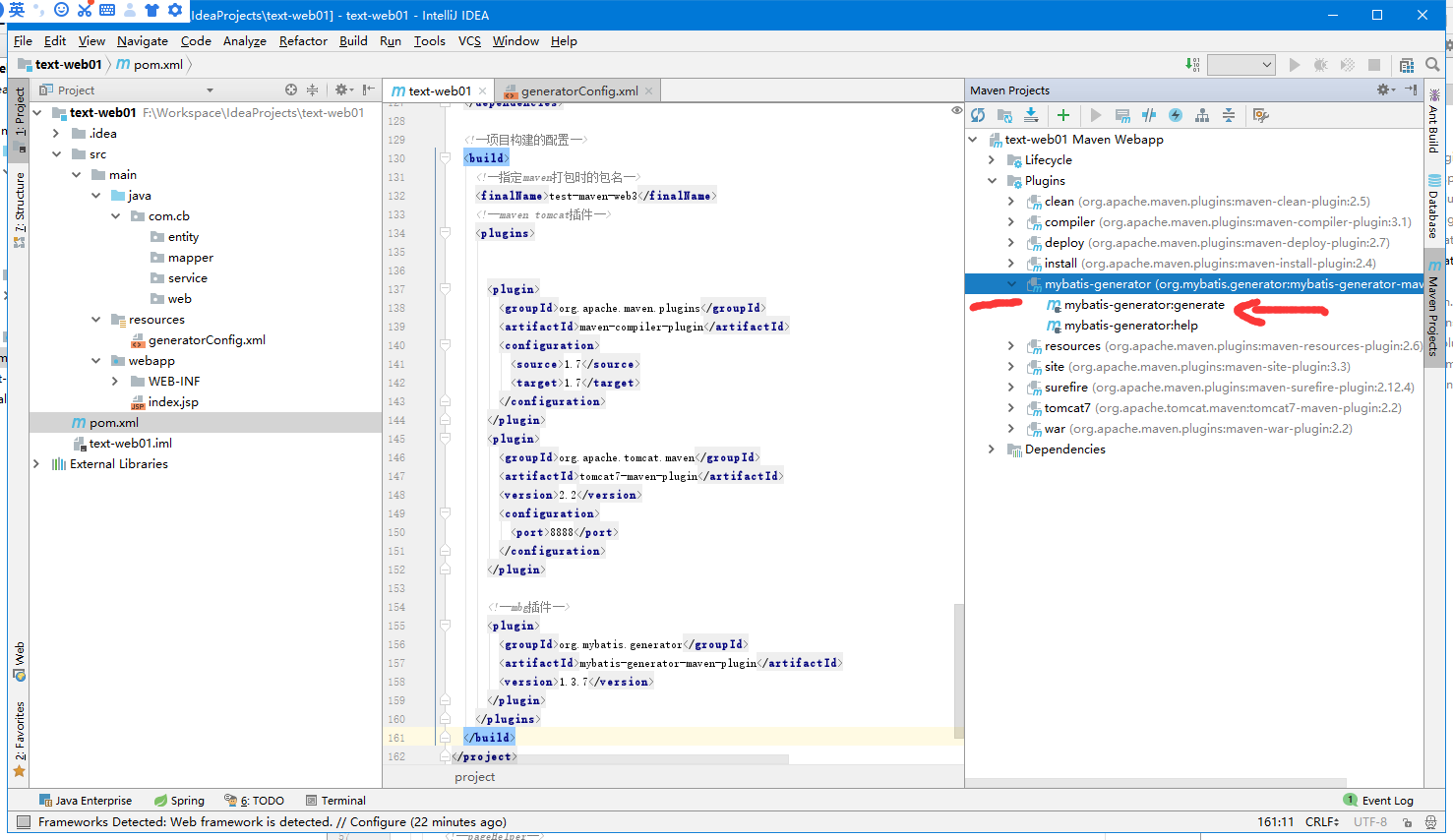
1. 生产的对应文件位置发生改变



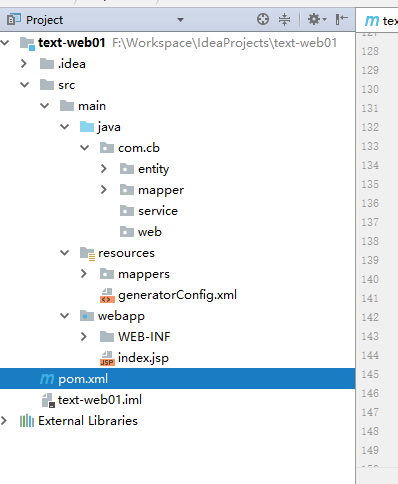
1. 配置该工程的pom.xml

<**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**maven.compiler.source**>1.7</**maven.compiler.source**>  
 <**maven.compiler.target**>1.7</**maven.compiler.target**>  
 <**junit.version**>4.12</**junit.version**>  
 <**mybatis.version**>3.4.6</**mybatis.version**>  
 <**spring.version**>5.0.14.RELEASE</**spring.version**>  
</**properties**>  
  
 *<!--工程的依赖-->* <**dependencies**>  
 <**dependency**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 <**version**>${junit.version}</**version**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 *<!--mybatis start-->  
 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->* <**dependency**>  
 <**groupId**>org.mybatis</**groupId**>  
 <**artifactId**>mybatis</**artifactId**>  
 <**version**>${mybatis.version}</**version**>  
 </**dependency**>  
  
 *<!--mybatis-spring-->  
 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->* <**dependency**>  
 <**groupId**>org.mybatis</**groupId**>  
 <**artifactId**>mybatis-spring</**artifactId**>  
 <**version**>1.3.2</**version**>  
 </**dependency**>  
  
 *<!--mbg-->  
 <!-- https://mvnrepository.com/artifact/org.mybatis.generator/mybatis-generator-core -->* <**dependency**>  
 <**groupId**>org.mybatis.generator</**groupId**>  
 <**artifactId**>mybatis-generator-core</**artifactId**>  
 <**version**>1.3.7</**version**>  
 </**dependency**>  
  
 *<!--pageHelper-->  
 <!-- https://mvnrepository.com/artifact/com.github.pagehelper/pagehelper -->* <**dependency**>  
 <**groupId**>com.github.pagehelper</**groupId**>  
 <**artifactId**>pagehelper</**artifactId**>  
 <**version**>5.1.10</**version**>  
 </**dependency**>  
  
 *<!-- https://mvnrepository.com/artifact/commons-dbcp/commons-dbcp -->* <**dependency**>  
 <**groupId**>commons-dbcp</**groupId**>  
 <**artifactId**>commons-dbcp</**artifactId**>  
 <**version**>1.4</**version**>  
 </**dependency**>  
  
  
 *<!--mybatis end-->  
  
  
 <!--mysql驱动-->  
 <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->* <**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
 <**version**>5.1.48</**version**>  
 </**dependency**>  
  
 *<!--spring配置开始-->  
 <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->* <**dependency**>  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>spring-context</**artifactId**>  
 <**version**>${spring.version}</**version**>  
 </**dependency**>  
  
 *<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->* <**dependency**>  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>spring-orm</**artifactId**>  
 <**version**>${spring.version}</**version**>  
 </**dependency**>  
 *<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->* <**dependency**>  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>spring-jdbc</**artifactId**>  
 <**version**>${spring.version}</**version**>  
 </**dependency**>  
  
 *<!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->* <**dependency**>  
 <**groupId**>org.springframework</**groupId**>  
 <**artifactId**>spring-test</**artifactId**>  
 <**version**>${spring.version}</**version**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
  
  
 *<!--spring配置结束-->  
 <!---log4j日志-->  
 <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->* <**dependency**>  
 <**groupId**>org.apache.logging.log4j</**groupId**>  
 <**artifactId**>log4j-core</**artifactId**>  
 <**version**>2.11.1</**version**>  
 </**dependency**>  
  
  
  
  
  
 </**dependencies**>  
  
 *<!--项目构建的配置-->* <**build**>  
 *<!--指定maven打包时的包名-->* <**finalName**>test-maven-web3</**finalName**>  
 *<!--maven tomcat插件-->* <**plugins**>  
  
  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**configuration**>  
 <**source**>1.7</**source**>  
 <**target**>1.7</**target**>  
 </**configuration**>  
 </**plugin**>  
 <**plugin**>  
 <**groupId**>org.apache.tomcat.maven</**groupId**>  
 <**artifactId**>tomcat7-maven-plugin</**artifactId**>  
 <**version**>2.2</**version**>  
 <**configuration**>  
 <**port**>8888</**port**>  
 </**configuration**>  
 </**plugin**>  
  
 *<!--mbg插件-->* <**plugin**>  
 <**groupId**>org.mybatis.generator</**groupId**>  
 <**artifactId**>mybatis-generator-maven-plugin</**artifactId**>  
 <**version**>1.3.7</**version**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>

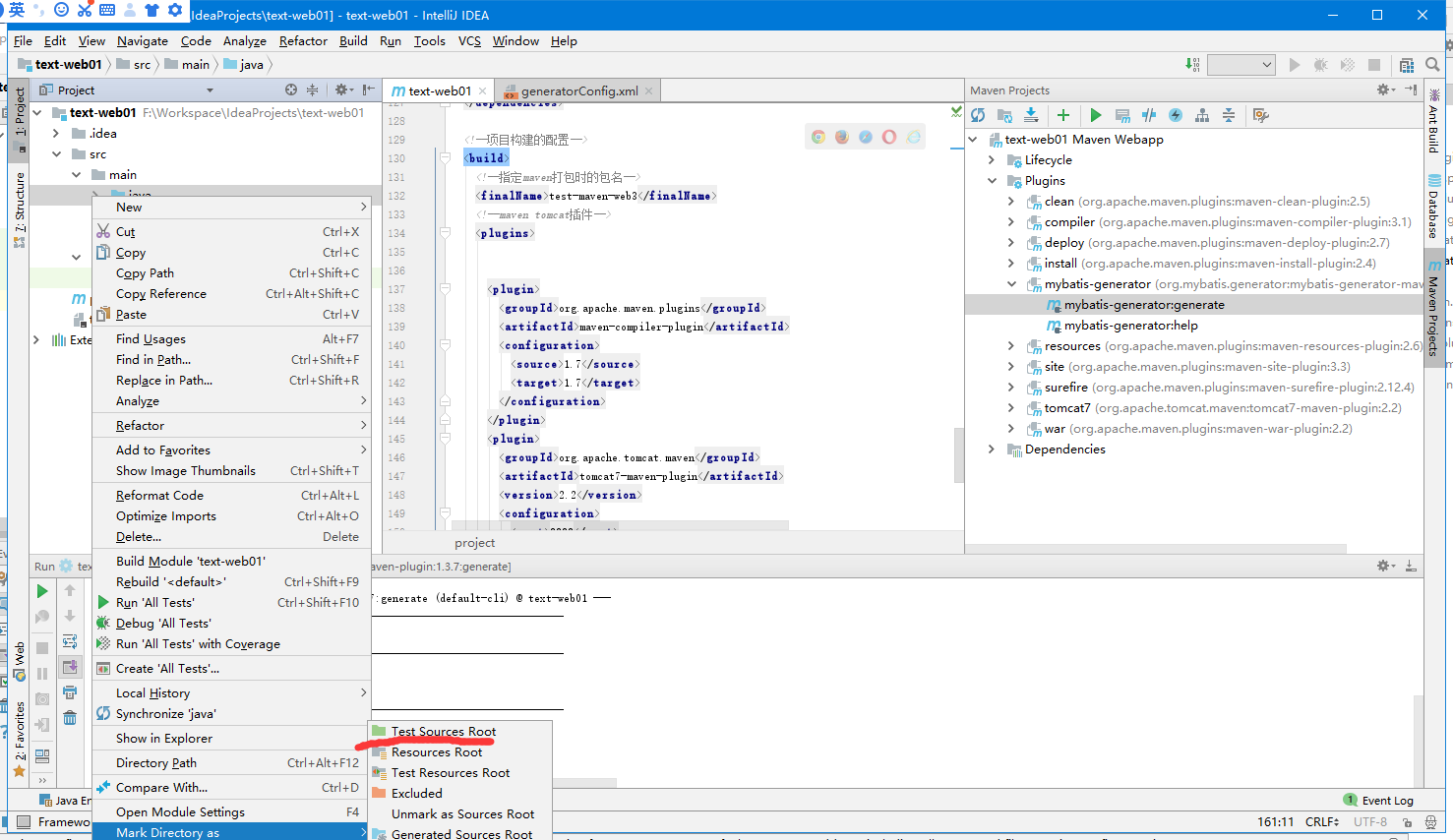
###### 启动mybatis



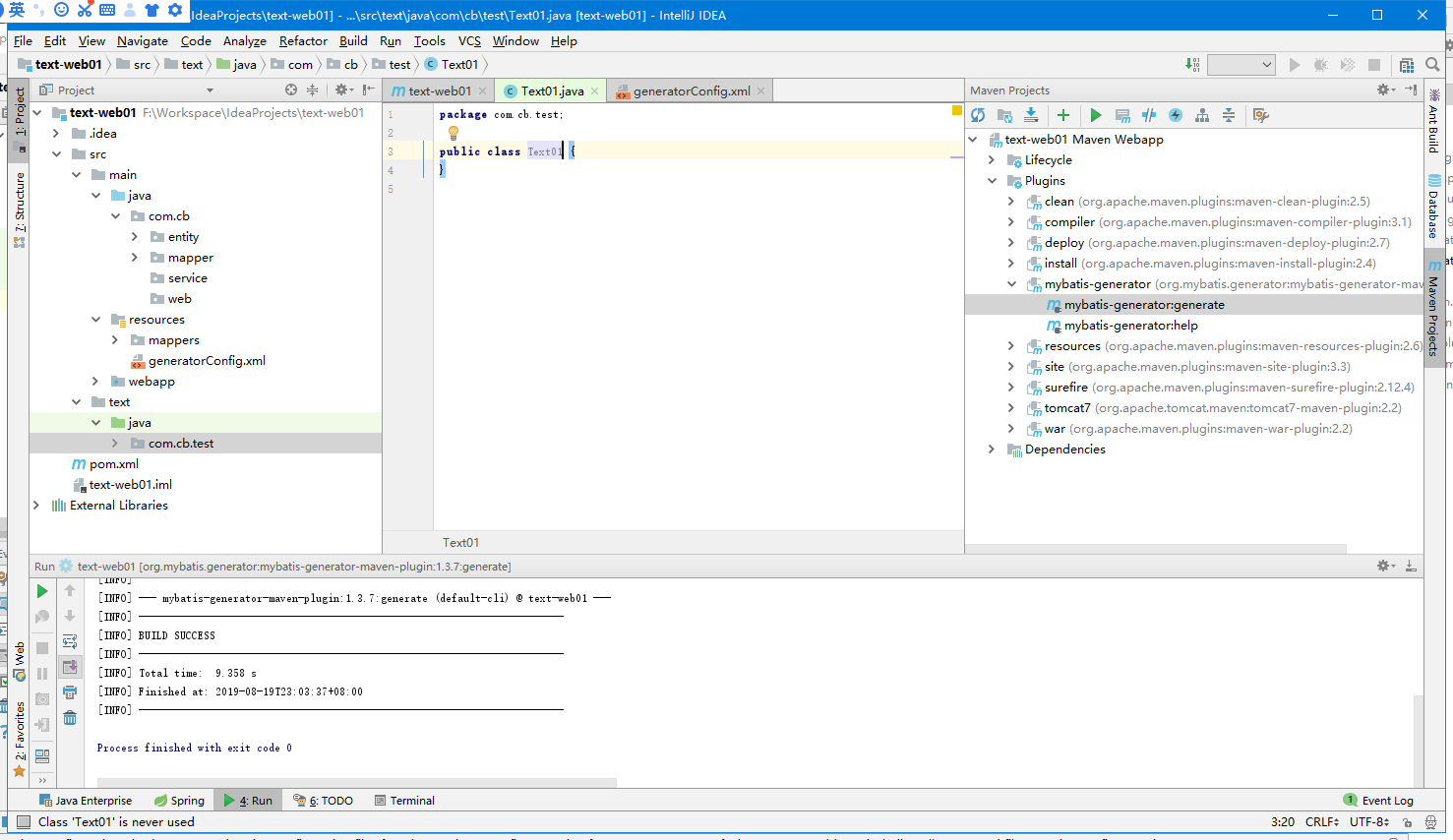
###### 成功后自动出现如下图文件



###### 测试，创建文件夹



如图



其中测试需要beans.xml和spring-datasources.xml文件，所有配置这两个xml同ecplise一样也是放在resources文件夹中

测试代码

**package** com.cb.test;  
  
**import** com.cb.mapper.CstCustomerMapper;  
**import** org.junit.Test;  
**import** org.junit.runner.RunWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.test.context.ContextConfiguration;  
**import** org.springframework.test.context.junit4.SpringJUnit4ClassRunner;  
  
@RunWith(SpringJUnit4ClassRunner.**class**)  
@ContextConfiguration(**"classpath:beans.xml"**)  
**public class** Test01 {  
 @Autowired  
 **private** CstCustomerMapper **cstCustomerMapper**;  
  
 @Test  
 **public void** test01(){  
 System.***out***.println(**cstCustomerMapper**.selectByExample(**null**));  
 }  
}

##### 3.1.4.配置SSM的web层

###### 1.配置beans.xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:p="http://www.springframework.org/schema/p"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context  
 https://www.springframework.org/schema/context/spring-context.xsd  
 "**>  
 *<!-- 1.注解扫描 -->* <**context:component-scan base-package="com.cb.\*\*"**>  
 *<!-- 这个xml好像对于spring -->  
 <!-- 排除@Controller注解(为什么？) -->* <**context:exclude-filter type="annotation"  
 expression="org.springframework.stereotype.Controller"** />  
 </**context:component-scan**>  
 *<!-- 2.datasource:mybatis -->* <**import resource="spring-datasource.xml"**/>  
 *<!-- <import resource="spring-quartz.xml"/> -->  
 <!-- 3.事务管理器 -->*</**beans**>

###### 配置spring-datasource.xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:p="http://www.springframework.org/schema/p"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:tx="http://www.springframework.org/schema/tx"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context  
 https://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/tx  
 https://www.springframework.org/schema/tx/spring-tx.xsd  
 "**>  
 *<!-- 1.数据源 -->* <**context:property-placeholder location="classpath:jdbc.properties"**/>  
 *<!-- 2,数据源：数据库的相关信息 :官网有 bdcp数据源，是数据源的一种 -->* <**bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"**>  
 <**property name="driverClassName" value="${driver}"** />  
 <**property name="url" value="${url}"** />  
 <**property name="username" value="root"** />  
 <**property name="password" value="root"** />  
 </**bean**>  
 *<!-- 2.sqlSessionFactory -->  
 <!-- 3、sqlSessionFactory 对象 -->* <**bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"**>  
 *<!-- 别名 -->* <**property name="typeAliasesPackage" value="com.cb.entity"**></**property**>  
 *<!-- mapper XML映射 -->* <**property name="mapperLocations" value="classpath\*:mappers/\*Mapper.xml"**></**property**>  
 *<!-- 数据源 -->* <**property name="dataSource" ref="dataSource"**></**property**>  
 *<!-- 分页插件 -->  
 <!-- 分页插件 -->* <**property name="plugins"**>  
 <**array**>  
 <**bean class="com.github.pagehelper.PageInterceptor"**>  
 <**property name="properties"**>  
 *<!--使用下面的方式配置参数，一行配置一个 -->* <**value**>  
 helperDialect=mysql  
 reasonable=true  
 supportMethodsArguments=true  
 params=count=countSql  
 autoRuntimeDialect=true  
 </**value**>  
 </**property**>  
 </**bean**>  
 </**array**>  
 </**property**>  
 </**bean**>  
 *<!-- 3. mapper代理对象 -->* <**bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>  
 *<!-- 扫描mapper接口的包 -->* <**property name="basePackage" value="com.cb.mapper"**></**property**>  
 </**bean**>  
 *<!-- 4. 注解事务 -->* <**bean id="txTransaction" class="org.springframework.jdbc.datasource.DataSourceTransactionManager "**>  
 <**property name="dataSource" ref="dataSource"**/>  
 </**bean**>  
 <**tx:annotation-driven transaction-manager="txTransaction" proxy-target-class="true"**/>  
 </**beans**>

###### jdbc.properties的配置

**driver**=**com.mysql.jdbc.Driver  
url**=**jdbc:mysql://127.0.0.1:3306/cb\_crm?useUnicode=true&characterEncoding=utf8&useSSL=true  
username**=**root  
password**=**root**

###### log4j.properties的配置

*# Global logging configuration\uFF0C\u5EFA\u8BAE\u5F00\u53D1\u73AF\u5883\u4E2D\u8981\u7528debug  
# debug***log4j.rootLogger**=**DEBUG, stdout***# Console output...***log4j.appender.stdout**=**org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.layout**=**org.apache.log4j.PatternLayout  
log4j.appender.stdout.layout.ConversionPattern**=**%5p [%t] - %m%n**

###### 配置web.xml文件