天猫商城

### 一 开发环境配置

#### linux系统安装

在VMware中安装linux系统，本项目使用的是CentOS-6.9-x86\_64-bin-DVD1。

#### linux软件源配置

##### 1.2.1 备份

|  |
| --- |
| mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup |

##### 1.2.2 下载

下载centos-base.repo到/etc/yum.reposd/

|  |
| --- |
| wget -O /etc/yum.repos.d/CentOS-Base.repo <http://mirrors.aliyun.com/repo/Centos-6.repo> |

##### 1.2.3 运行

|  |
| --- |
| yum makecache |

#### linux下安装JDK

##### 下载JDK

|  |
| --- |
| wget jdk地址 |

##### 删除自带JDK

|  |
| --- |
| java –version  rpm –qa | grep jdk  sudo yum remove “系统自带的JDK” |

##### 安装JDK

|  |
| --- |
| 赋予JDK权限  sudo chmod 777 jdk-8u162-linux-x64.rpm  解压JDK，解压后默认在/usr/java目录下  sudo rpm –ivh jdk-8u162-linux-x64.rpm  配置环境变量  sudo vim /etc/profile  在文件最下方添加环境变量  export JAVA\_HOME=/usr/java/jdk-8u162-linux-x64  export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  export PATH=$PATH:$JAVA\_HOME/bin  使环境变量生效  source /etc/profile |

#### linux下安装tomcat

##### 1.4.1 下载tomcat

|  |
| --- |
| wget tomcat地址 |

##### 1.4.2 解压tomcat

|  |
| --- |
| tar –zxvf apche-tomcat-7.0.85.tar.gz |

##### 1.4.3 安装tomcat

|  |
| --- |
| // 复制解压的tomcat到指定文件夹  cp –rf /root/apache-tomcat-7.0.85 /developer  //配置环境变量  sudo vim /etc/profile  // 在文件最下方添加环境变量  export CATALINA\_HOME=/developer/apache-tomcat-7.0.85  // 使环境变量生效  source /etc/profile |

##### 1.4.4 配置UTF-8字符集

|  |
| --- |
| sudo vim /developer/config/server.xml  // 在8080端口配置的末尾添加字符集  Encoding=”UTF-8” |

#### linux下安装maven

##### 1.5.1 下载maven

|  |
| --- |
| wget maven地址 |

##### 1.5.2 解压maven

|  |
| --- |
| tar –zxvf apach-maven-3.5.3-bin.tar.gz |

##### 1.5.3 安装maven

|  |
| --- |
| // 复制解压的maven到指定文件夹  cp –rf /root/apache-maven-3.5.3/developer  //配置环境变量  sudo vim /etc/profile  // 在文件最下方添加环境变量  export CATALINA\_HOME=/developer/apache-maven-3.5.3  export PATH=$PATH:$JAVA\_HOME/bin:$MAVEN\_HOME/bin  // 使环境变量生效  source /etc/profile |

#### linux下安装vsftpd

##### 1.6.1 安装vsftpd

|  |
| --- |
| yum –y install vsftpd  // 验证是否安装成功，配置文件在/etc/vsftpdvsftpd.conf  rpm –qa|grep vsftpd |

##### 1.6.2 创建虚拟用户

|  |
| --- |
| // 创建文件夹  mkdir /ftpfile  // 添加匿名用户  useradd ftpuser –d /ftpfile –s /sbin/nologin  // 修改ftpfile权限  chown –R ftpuser.ftpuser /ftpfile  // 重置ftpuser密码，密码为ftpuser  passwd ftpuser |

##### 1.6.3 配置vsftpd

|  |
| --- |
| vim /etc/vsftpd/vsftpd.conf  // 添加信息  anonymous\_enable=NO  ftpd\_banner=Welcome to tmall FTP service.  local\_root=/ftpfile  use\_localtime=yes  chroot\_list\_enable=YES  chroot\_lst\_file=/etc/vsftpd/chroot\_list  pasv\_min\_port=61001  pasv\_max\_port=62000  // 防火墙配置  -A INPUT -p TCP --dport 61001:62000 -j ACCEPT  -A OUTPUT -p TCP --sport 61001:62000 -j ACCEPT  -A INPUT -p TCP --dport 20 -j ACCEPT  -A OUTPUT -p TCP --sport 20 -j ACCEPT  -A INPUT -p TCP --dport 21 -j ACCEPT  -A OUTPUT -p TCP --sport 21 -j ACCEPT  // 使匿名用户可以创建文件  vim /etc/selinux/config  // 添加  SELINUX=disable  #SELINUX=enforcing  // 让命令生效  setenforce 0  // 重启ftp  service vsftpd restart  // 连接ftp  ftp 192.168.178.130 |

#### window下安装ftpserver

##### 1.7.1 安装配置ftpserver

下载解压缩FTPServer，并配置好用户名密码

#### linux下安装nginx

##### 1.8.1 安装nginx依赖

|  |
| --- |
| // 安装nginx的依赖  yum install gcc-c++  yum install pcre pcre-devel  yum install zlib zlib-devel  yum install openssl openssl-devel |

##### 1.8.2 下载nginx

|  |
| --- |
| wget ngin地址 |

##### 1.8.3 解压nginx

|  |
| --- |
| tar –zxvf nginx-1.12.2.tar.gz |

##### 1.8.4 安装nginx

|  |
| --- |
| 进入nginx目录执行./configure  make  make install  find –name nginx  nginx默认装载/usr/local |

##### 1.8.5 nginx常用命令

|  |
| --- |
| // 测试配置文件  安装路径下/nginx/sbin/nginx –t  // 启动命令  安装路径下/nginx/sbin/nginx  // 停止命令  安装路径下/nginx/sbin/nginx –stop 或者 nginx –s quit  // 重启命令  安装路径下/nginx/sbin/nginx –s reload  // 查看进程命令  ps –ef | grep nginx  // 平滑重启  kill –HUP nginx主进程号  // 增加防火墙访问权限  vim /etc/sysconfig/iptables  -A INPUT –p tcp –m state –state NEW –m tcp –dport 80 –j ACCEPT  Service iptables restart |

#### linux下nginx域名解析配置

##### 1.9.1 配置hosts文件

|  |
| --- |
| //修改hosts，window和linux都要修改  vim /etc/hosts  192.168.178.130 www.laughing.com  192.168.178.139 image.laughing.com  192.168.178.139 s.laughing.com |

##### 1.9.2 配置nginx域名

|  |
| --- |
| // 进入nginx目录  cd /usr/local/nginx/conf  mkdir vhost  vim nginx.conf  添加include vhost/\*.conf;  // 在vshost下新建文件  vim www.laughing.com.conf  server {  listen 80;  autoindex on;  server\_name www.laughing.com;  access\_log /usr/local/nginx/logs/access.log combined;  index index.html index.htm index.jsp index.php;  #error\_page 404 /404.html;  if ( $query\_string ~\* ".\*[\;'\<\>].\*" ){  return 404;  }  location / {  proxy\_pass http://127.0.0.1:8080;  add\_header Access-Control-Allow-Origin \*;  }  }  //  vim image.laughing.com.conf  server {  listen 80;  autoindex off;  server\_name image.laughing.com;  access\_log /usr/local/nginx/logs/access.log combined;  index index.html index.htm index.jsp index.php;  #error\_page 404 /404.html;  if ( $query\_string ~\* ".\*[\;'\<\>].\*" ){  return 404;  }  location ~ /(mmall\_fe|mmall\_admin\_fe)/dist/view/\* {  deny all;  }  location / {  root /ftpfile/;  add\_header Access-Control-Allow-Origin \*;  }  }  // 访问测试  www.laughing.com  image.laughing.com |

#### windows下安装nginx

##### 1.10.1 下载解压nginx

在nginx官网下载稳定版本，放在本地任意位置并解压

##### 1.10.2配置nginx

（1）在nginx目录conf文件夹下的ngin.conf中http结束标签后添加include vhost/\*.conf;

（2）在conf文件夹下创建vhost文件夹

（3）新建image.laughing.com.conf文件

|  |
| --- |
| server {  listen 80;  autoindex off;  server\_name image.laughing.com;  access\_log c:/access.log combined;  index index.html index.htm index.jsp index.php;  #error\_page 404 /404.html;  if ( $query\_string ~\* ".\*[\;'\<\>].\*" ){  return 404;  }  location ~ /(mmall\_fe|mmall\_admin\_fe)/dist/view/\* {  deny all;  }  location / {  root F:\开发\Share\img;  add\_header Access-Control-Allow-Origin \*;  }  } |

（5）测试

在cmd窗口执行nginx –t

（6）新建tomcat.laughing.com.conf文件

|  |
| --- |
| server {  listen 80;  autoindex on;  server\_name tomcat.laughing.com;  access\_log c:/access.log combined;  index index.html index.htm index.jsp index.php;  #error\_page 404 /404.html;  if ( $query\_string ~\* ".\*[\;'\<\>].\*" ){  return 404;  }  location / {  proxy\_pass http://127.0.0.1:8080;  add\_header Access-Control-Allow-Origin \*;  }  } |

（7）修改hosts文件

|  |
| --- |
| 127.0.0.1 image.laughing.com  127.0.0.1 tomcat.laughing.com |

（8）重启nginx

|  |
| --- |
| nginx -s reload |

（9）访问域名

|  |
| --- |
| tomcat.laughing.com  image.laughing.com |

#### linux下安装mysql

##### 1.11.1 安装mysql

|  |
| --- |
| yum –y install mysql-server  rpm –qa|grep mysql-server |

##### 1.11.2 配置字符集

|  |
| --- |
| default-character-set=utf8  character-set-server=utf8 |

##### 1.11.3 配置mysql自启动

|  |
| --- |
| chkconfig mysqld on  chkconfig –list mysqld  2-5位为on状态 |

##### 1.11.4 配置防火墙

|  |
| --- |
| vim /etc/sysconfig/iptables  -A INPUT –p tcp –m tcp –dport 3306 –j ACCEPT  service iptables restart |

##### 1.11.5 创建mysql帐号

|  |
| --- |
| insert into mysql.user(Host,User,Password) values("localhost","laughing",password("laughing"));  select user,host from mysql.user;  // 删除匿名用户  Delete from mysql.user where user=’’;  Flush privileges; |

##### 1.11.6 创建数据库

|  |
| --- |
| create database `tmall` default character set utf8 collate utf8\_general\_ci;  // 查看权限  select \* from mysql.user \G  // 赋予本地用户所有权限  grant select,delete,create on tmall.\* to laughing@’localhost’ identified by ‘laughing’  // 赋予帐号外网所有权限  grant all privileges on tmall.\* to laughing@’%’ identified by ‘laughing’ with grant option |

#### linux下安装git

##### 1.12.1 下载git

|  |
| --- |
| wget https://github.com/git/git/archive/v2.8.0.tar.gz |

##### 1.12.2 安装依赖

|  |
| --- |
| yum -y install zlib-devel openssl-devel cpio expat-devel gettext-devel curl-devel perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker |

##### 1.12.3 解压git

|  |
| --- |
| tar –zxvf v2.8.0.tar.gz |

##### 1.12.4 编译git

|  |
| --- |
| cd git-2.8.0  make prefix=/usr/local all |

##### 1.12.5 安装git

|  |
| --- |
| make prefix=/usr/local install  git --version |

##### 1.12.6 配置私钥

|  |
| --- |
| ssh-keygen –t rsa –C “425984393@qq.com”  ssh-agent bash  ssh-add ~/.ssh/id\_rsa（/前面是数字1左边的波浪线）  cat ~/.ssh/id\_rsa.pub  复制key的内容到github |

### 二 项目初始化

#### 2.1 配置IDEA

##### 2.1.1 配置JDK

Configure->Project Defaults->Project Structure

##### 2.1.2 配置Maven

Configure->Project Defaults->Settings

#### 2.2 git初始化

##### 2.2.1 创建README.md文件

在项目根目录下创建README.md文件

##### 2.2.2 配置.gitignore文件

在项目根目录下创建.gitignore文件

|  |
| --- |
| \*.class  #package file \*.war \*.ear  #kdiff3 ignore \*.orig  #maven ignore target/  #eclipse ignore .settings/ .project .classpath  #idea .idea/ /idea/ \*.ipr \*.iml \*.iws  #temp file \*.log \*.cache \*.diff \*.patch \*.tmp  #system ignore .DS\_Store Thumbs.db |

##### 2.2.3 提交到git仓库

|  |
| --- |
| git add .  git status  git commit -m "first commit init project"  git remote add origin [git@github.com:Lee-Shawn/tmall.git](mailto:git@github.com:Lee-Shawn/tmall.git)  git branch  git pull  git push -u -f origin master |

##### 2.2.4 创建分支

|  |
| --- |
| // 查看当前分支  git branch –r  // 创建新分支v1.0  git checkout -b v1.0 origin/master  // 推送到远程  git push origin HEAD -u |

##### 2.2.5 合并分支到master

|  |
| --- |
| git checkout master  git pull origin master  git merge v1.0 |

#### 2.3 配置pom.xml

##### 2.3.1 配置maven的pom.xml文件

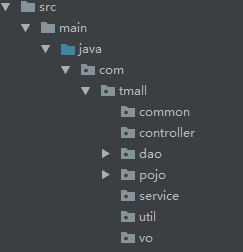
在pom.xml中配置项目所需要的依赖

|  |
| --- |
| <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/maven-v4\_0\_0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com</groupId>  <artifactId>tmall</artifactId>  <packaging>war</packaging>  <version>1.0-SNAPSHOT</version>  <name>tmall Maven Webapp</name>  <url>http://maven.apache.org</url>   <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <maven.compiler.encoding>UTF-8</maven.compiler.encoding>   <org.springframework.version>4.3.10.RELEASE</org.springframework.version>  <org.mybatis.version>3.4.1</org.mybatis.version>  <org.mybatis.spring.version>1.3.0</org.mybatis.spring.version>  </properties>   <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.12</version>  <scope>test</scope>  </dependency>   <dependency>  <groupId>org.apache.tomcat</groupId>  <artifactId>tomcat-servlet-api</artifactId>  <version>7.0.64</version>  </dependency>   <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <version>${org.springframework.version}</version>  </dependency>   <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-oxm</artifactId>  <version>${org.springframework.version}</version>  </dependency>   <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-jdbc</artifactId>  <version>${org.springframework.version}</version>  </dependency>   <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-tx</artifactId>  <version>${org.springframework.version}</version>  </dependency>   <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-test</artifactId>  <version>${org.springframework.version}</version>  </dependency>   <dependency>  <groupId>org.aspectj</groupId>  <artifactId>aspectjweaver</artifactId>  <version>1.8.1</version>  </dependency>   <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis-spring</artifactId>  <version>${org.mybatis.spring.version}</version>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>${org.mybatis.version}</version>  </dependency>   <dependency>  <groupId>org.aspectj</groupId>  <artifactId>aspectjrt</artifactId>  <version>1.8.13</version>  </dependency>   <dependency>  <groupId>org.codehaus.jackson</groupId>  <artifactId>jackson-mapper-asl</artifactId>  <version>1.9.13</version>  </dependency>   <dependency>  <groupId>commons-dbcp</groupId>  <artifactId>commons-dbcp</artifactId>  <version>1.4</version>  <!--<scope>runtime</scope>-->  </dependency>   <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-classic</artifactId>  <version>1.1.8</version>  <scope>compile</scope>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-core</artifactId>  <version>1.1.8</version>  <scope>compile</scope>  </dependency>   <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.38</version>  </dependency>   <dependency>  <groupId>com.google.guava</groupId>  <artifactId>guava</artifactId>  <version>20.0</version>  </dependency>   <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>3.1</version>  </dependency>   <dependency>  <groupId>commons-collections</groupId>  <artifactId>commons-collections</artifactId>  <version>3.2.1</version>  </dependency>   <dependency>  <groupId>joda-time</groupId>  <artifactId>joda-time</artifactId>  <version>2.3</version>  </dependency>   <!-- id加密解密 -->  <dependency>  <groupId>org.hashids</groupId>  <artifactId>hashids</artifactId>  <version>1.0.1</version>  </dependency>   <!-- ftpclient -->  <dependency>  <groupId>commons-net</groupId>  <artifactId>commons-net</artifactId>  <version>3.1</version>  </dependency>   <!-- file upload -->   <!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->  <dependency>  <groupId>commons-fileupload</groupId>  <artifactId>commons-fileupload</artifactId>  <version>1.2.2</version>  </dependency>   <dependency>  <groupId>commons-io</groupId>  <artifactId>commons-io</artifactId>  <version>2.0.1</version>  </dependency>   <!-- mybatis pager -->   <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper</artifactId>  <version>4.1.0</version>  </dependency>   <dependency>  <groupId>com.github.miemiedev</groupId>  <artifactId>mybatis-paginator</artifactId>  <version>1.2.17</version>  </dependency>   <dependency>  <groupId>com.github.jsqlparser</groupId>  <artifactId>jsqlparser</artifactId>  <version>0.9.4</version>  </dependency>   <!-- alipay -->  <dependency>  <groupId>commons-codec</groupId>  <artifactId>commons-codec</artifactId>  <version>1.10</version>  </dependency>  <dependency>  <groupId>commons-configuration</groupId>  <artifactId>commons-configuration</artifactId>  <version>1.10</version>  </dependency>  <dependency>  <groupId>commons-lang</groupId>  <artifactId>commons-lang</artifactId>  <version>2.6</version>  </dependency>  <dependency>  <groupId>commons-logging</groupId>  <artifactId>commons-logging</artifactId>  <version>1.1.1</version>  </dependency>  <dependency>  <groupId>com.google.zxing</groupId>  <artifactId>core</artifactId>  <version>2.1</version>  </dependency>  <dependency>  <groupId>com.google.code.gson</groupId>  <artifactId>gson</artifactId>  <version>2.3.1</version>  </dependency>  <dependency>  <groupId>org.hamcrest</groupId>  <artifactId>hamcrest-core</artifactId>  <version>1.3</version>  </dependency>   <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  <version>2.9.0</version>  </dependency>   </dependencies>   <build>  <finalName>tmall</finalName>  <plugins>  <plugin>  <groupId>org.mybatis.generator</groupId>  <artifactId>mybatis-generator-maven-plugin</artifactId>  <version>1.3.2</version>  <configuration>  <verbose>true</verbose>  <overwrite>true</overwrite>  </configuration>  </plugin>   <!-- geelynote maven的核心插件之-complier插件默认只支持编译Java 1.4，因此需要加上支持高版本jre的配置，在pom.xml里面加上 增加编译插件 -->  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <configuration>  <source>1.8</source>  <target>1.8</target>  <encoding>UTF-8</encoding>  <compilerArguments>  <extdirs>${project.basedir}/src/main/webapp/WEB-INF/lib</extdirs>  </compilerArguments>  </configuration>  </plugin>  </plugins>   </build> </project> |

#### 2.4 项目包结构初始化

##### 2.4.1 新建包结构

（1）在src文件夹先新建main/java/com/tmall，然后再分别建各个功能文件夹。



（2）在src文件夹下新建与main同级的test文件夹。

（3）在java文件夹上右键选择make directory as将其设为Resource Root，在test文件夹上右键设为Test Resource Root。

#### 2.5 配置mybaits-generator

##### 2.5.1 配置pom.xml

在pom.xml中配置依赖，引入jar包

|  |
| --- |
| <build>  <plugins>  <plugin>  <groupId>org.mybatis.generator</groupId>  <artifactId>mybatis-generator-maven-plugin</artifactId>  <version>1.3.2</version>  <configuration>  <verbose>true</verbose>  <overwrite>true</overwrite>  </configuration>  </plugin>  </plugins> </build> |

##### 2.5.2 配置generator.xml

在resources目录下新建generator.xml文件，并进行配置

|  |
| --- |
| <?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>  <!--导入属性配置-->  <properties resource="datasource.properties"></properties>   <!--指定特定数据库的jdbc驱动jar包的位置-->  <classPathEntry location="${db.driverLocation}"/>   <context id="default" targetRuntime="MyBatis3">   <!-- optional，旨在创建class时，对注释进行控制 -->  <commentGenerator>  <property name="suppressDate" value="true"/>  <property name="suppressAllComments" value="true"/>  </commentGenerator>   <!--jdbc的数据库连接 -->  <jdbcConnection  driverClass="${db.driverClassName}"  connectionURL="${db.url}"  userId="${db.username}"  password="${db.password}">  </jdbcConnection>   <!-- 非必需，类型处理器，在数据库类型和java类型之间的转换控制-->  <javaTypeResolver>  <property name="forceBigDecimals" value="false"/>  </javaTypeResolver>    <!-- Model模型生成器,用来生成含有主键key的类，记录类 以及查询Example类  targetPackage 指定生成的model生成所在的包名  targetProject 指定在该项目下所在的路径  -->  <!--<javaModelGenerator targetPackage="com.tmall.pojo" targetProject=".\src\main\java">-->  <javaModelGenerator targetPackage="com.tmall.pojo" targetProject="./src/main/java">  <!-- 是否允许子包，即targetPackage.schemaName.tableName -->  <property name="enableSubPackages" value="false"/>  <!-- 是否对model添加 构造函数 -->  <property name="constructorBased" value="true"/>  <!-- 是否对类CHAR类型的列的数据进行trim操作 -->  <property name="trimStrings" value="true"/>  <!-- 建立的Model对象是否 不可改变 即生成的Model对象不会有 setter方法，只有构造方法 -->  <property name="immutable" value="false"/>  </javaModelGenerator>   <!--mapper映射文件生成所在的目录 为每一个数据库的表生成对应的SqlMap文件 -->  <!--<sqlMapGenerator targetPackage="mappers" targetProject=".\src\main\resources">-->  <sqlMapGenerator targetPackage="mappers" targetProject="./src/main/resources">  <property name="enableSubPackages" value="false"/>  </sqlMapGenerator>   <!-- 客户端代码，生成易于使用的针对Model对象和XML配置文件 的代码  type="ANNOTATEDMAPPER",生成Java Model 和基于注解的Mapper对象  type="MIXEDMAPPER",生成基于注解的Java Model 和相应的Mapper对象  type="XMLMAPPER",生成SQLMap XML文件和独立的Mapper接口  -->   <!-- targetPackage：mapper接口dao生成的位置 -->  <!--<javaClientGenerator type="XMLMAPPER" targetPackage="com.tmall.dao" targetProject=".\src\main\java">-->  <javaClientGenerator type="XMLMAPPER" targetPackage="com.tmall.dao" targetProject="./src/main/java">  <!-- enableSubPackages:是否让schema作为包的后缀 -->  <property name="enableSubPackages" value="false" />  </javaClientGenerator>    <table tableName="tmall\_shipping" domainObjectName="Shipping" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_cart" domainObjectName="Cart" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_cart\_item" domainObjectName="CartItem" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_category" domainObjectName="Category" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_order" domainObjectName="Order" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_order\_item" domainObjectName="OrderItem" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_pay\_info" domainObjectName="PayInfo" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>  <table tableName="tmall\_product" domainObjectName="Product" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false">  <columnOverride column="detail" jdbcType="VARCHAR" />  <columnOverride column="sub\_images" jdbcType="VARCHAR" />  </table>  <table tableName="tmall\_user" domainObjectName="User" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" selectByExampleQueryId="false"></table>    <!-- geelynote mybatis插件的搭建 -->  </context> </generatorConfiguration> |

##### 2.5.3 配置datasource.properties

在resources目录下新建datesource.properties文件并配置

|  |
| --- |
| db.driverLocation=D:/PrograSoftware/Maven/repository/mysql/mysql-connector-java/5.1.38/mysql-connector-java-5.1.38.jar db.driverClassName=com.mysql.jdbc.Driver db.url=jdbc:mysql://localhost:3306/test\_tmall?characterEncoding=utf-8 db.username=root db.password=root   db.initialSize = 20 db.maxActive = 50 db.maxIdle = 20 db.minIdle = 10 db.maxWait = 10 db.defaultAutoCommit = true db.minEvictableIdleTimeMillis = 3600000 |

##### 2.5.4 执行脚本生成文件

点击idea右侧maven project，选择mybatis-generator，双击执行，生成pojo和dao相关类以及xml文件

#### 2.6 配置mybatis-plugin

##### 2.6.1 下载安装mybatis plugin插件

File->settings->plugin->browse repositories，搜索mybatis plugin并安装