

淘淘商城

第一天

1. 课程计划

第一天：

- 1、电商行业的背景。
- 2、淘宝商城的系统架构
 - a) 功能介绍
 - b) 架构讲解
- 3、工程搭建-后台工程
 - a) 使用 maven 搭建工程
 - b) 使用 maven 的 tomcat 插件启动工程
- 4、Svn 的使用。

第二天：

- 1、服务中间件 dubbo
- 2、SSM 框架整合。
- 3、整合测试
- 4、商品列表查询功能实现。

第三天：

- 1、商品类目选择
- 2、图片上传
 - a) 图片服务器 FastDFS
 - b) 图片上传功能实现
- 3、富文本编辑器的使用 KindEditor
- 4、商品添加功能完成

第四天：

- 1、前台系统搭建
- 2、CMS 系统的实现
 - a) 内容分类管理
 - b) 内容管理
- 3、前台内容动态展示

第五天：

- 1、Redis 服务器搭建
- 2、向业务逻辑中添加缓存。
- 3、使用 redis 做缓存
- 4、缓存同步。
- 5、Solr 服务器安装

第六天：

- 1、Solr 使用测试
- 2、把数据库中的数据导入索引库

3、搜索功能的实现

第七天：

- 1、solr 集群搭建
- 2、使用 solrj 管理 solr 集群
- 3、把搜索功能切换到集群版

第八天：

- 1、什么是 MQ
- 2、MQ 的应用场景
- 3、ActiveMQ 的使用方法。
- 4、使用消息队列实现商品同步。

第九天：

- 1、商品详情页面展示，动态展示 jsp + redis
- 2、使用 freemarker 实现网页静态化
- 3、ActiveMq 同步生成静态网页

第十天：

- 1、nginx 的安装
- 2、Nginx 配置虚拟机
- 3、Nginx 实现反向代理
- 4、Nginx 实现负载均衡
- 5、Sso 系统工程搭建

第十一天：

- 1、sso 注册功能实现
- 2、sso 登录功能实现
- 3、通过 token 获得用户信息
- 4、Ajax 跨域请求（jsonp）

第十二天：

- 1、购物车实现
- 2、订单确认页面展示

第十三天：

- 1、订单系统完成
- 2、系统部署

第十四天~十六天：

项目实战
项目总结

2. 电商行业背景介绍

近年来，中国的电子商务快速发展，交易额连创新高，电子商务在各领域的应用不断拓展和深化、相关服务业蓬勃发展、支撑体系不断健全完善、创新的动力和能力 不断增强。电子商务正在与实体经济深度融合，进入规模性发展阶段，对经济社会生活的影响不断增大，正成为我国经济发展的新引擎。

中国电子商务研究中心数据显示，截止到 2012 年底，中国电子商务市场交易规模达 7.85 万亿人民币，同比增长 30.83%。其中，B2B 电子商务交易额 达 6.25 万亿，同比增长 27%。而 2011 年全年，中国电子商务市场交易额达 6 万亿人民币，同比增长 33%，占 GDP 比重上升到 13%；2012 年，电 子商务占 GDP 的比重已经高达 15%。预计 2013 年我国电子商务规模将突破十万亿大关。



009-2015 年中国电子商务市场交易规模（万亿元）

2.1.11.11



结论：

- 1、电商行业很挣钱，找互联网相关的工作。
- 2、电商行业技术要求很高、高可用、海量数据的存储。

2.2. 电商行业技术特点

- 技术新
- 技术范围广
- 分布式
- 高并发、集群、负载均衡、高可用
- 海量数据
- 业务复杂
- 系统安全

3. 淘淘商城介绍

淘淘网上商城是一个综合性的 B2C 平台，类似京东商城、天猫商城。会员可以在商城浏览商品、下订单，以及参加各种活动。

管理员、运营可以在平台后台管理系统中管理商品、订单、会员等。

客服可以在后台管理系统中处理用户的询问以及投诉。

电商模式：

B2B：商家到商家。阿里巴巴，慧聪网、铭万网。

B2C：商家到用户。京东。

C2C：用户到用户。淘宝。

B2B2C：商家到商家到用户。天猫。

O2O：线上到线下。百度外卖、美团、饿了么。

4. 淘淘商城架构

4.1. 功能列表



后台管理系统：管理商品、订单、类目、商品规格属性、用户管理以及内容发布等功能。

前台系统：用户可以在前台系统中进行注册、登录、浏览商品、首页、下单等操作。

会员系统：用户可以在该系统中查询已下的订单、收藏的商品、我的优惠券、团购等信息。

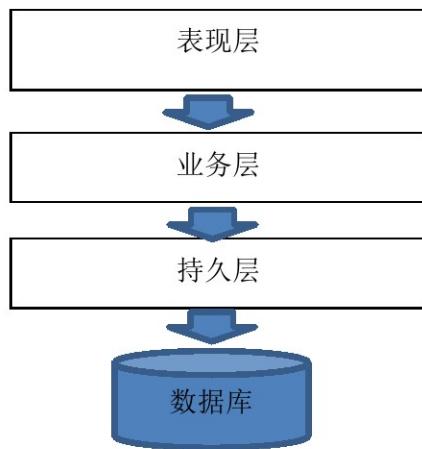
订单系统：提供下单、查询订单、修改订单状态、定时处理订单。

搜索系统：提供商品的搜索功能。

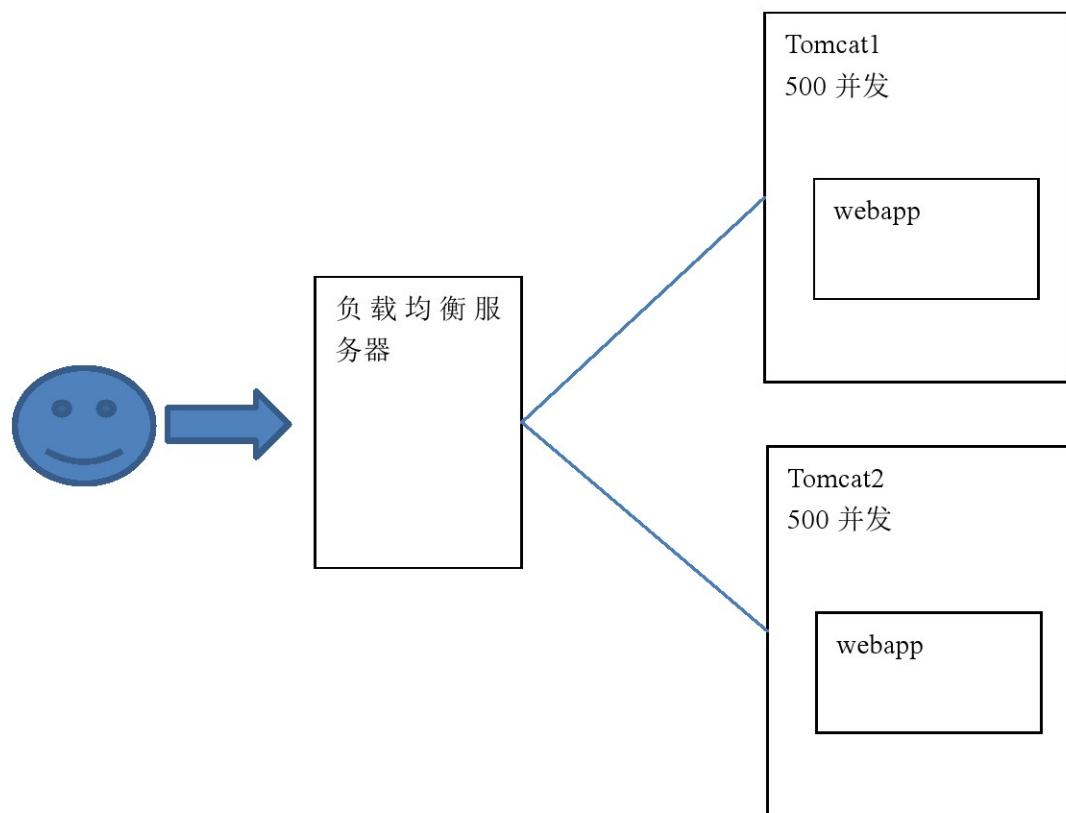
单点登录系统：为多个系统之间提供用户登录凭证以及查询登录用户的信息。

4.2. 系统架构

4.2.1. 传统架构



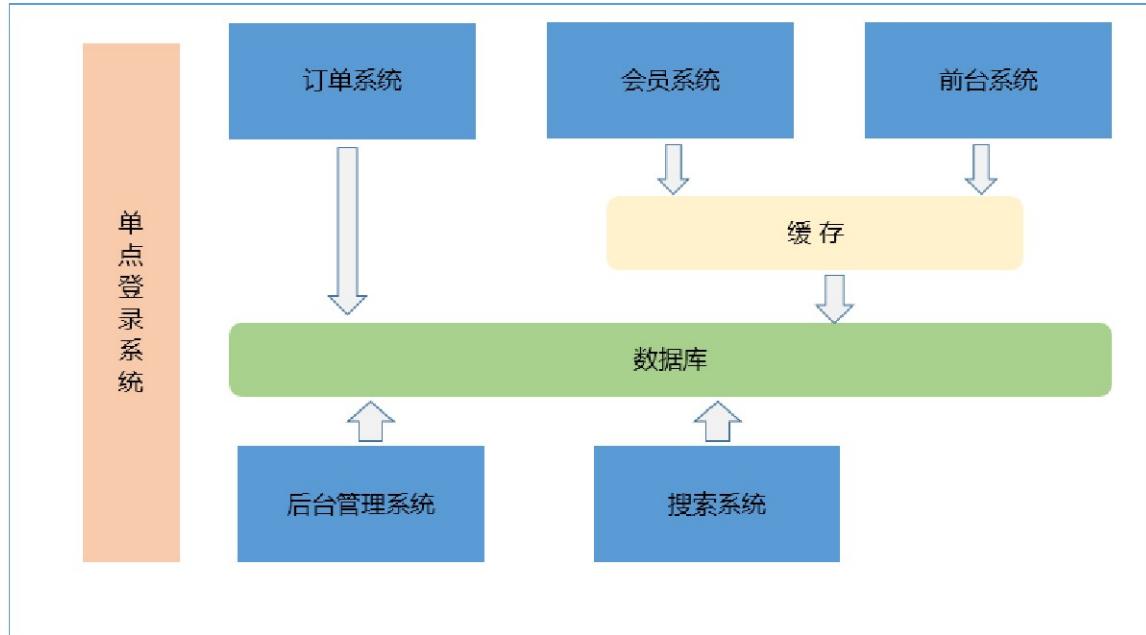
4.2.2. 1000 并发



需要 20 台服务器做 tomcat 集群。当 tomcat 集群中节点数量增加，服务能力先增加后下降。
所以集群中节点数量不能太多，一般也就 5 个左右。

4.2.3. 10000 并发

需要按照功能点把系统拆分，拆分成独立的功能。单独为某一个节点添加服务器。需要系统之间配合才能完成整个业务逻辑。叫做分布式。



分布式架构：多个子系统相互协作才能完成业务流程。系统之间需要进行通信。

集群：同一个工程部署到多台服务器上。

分布式架构：

把系统按照模块拆分成多个子系统。

优点：

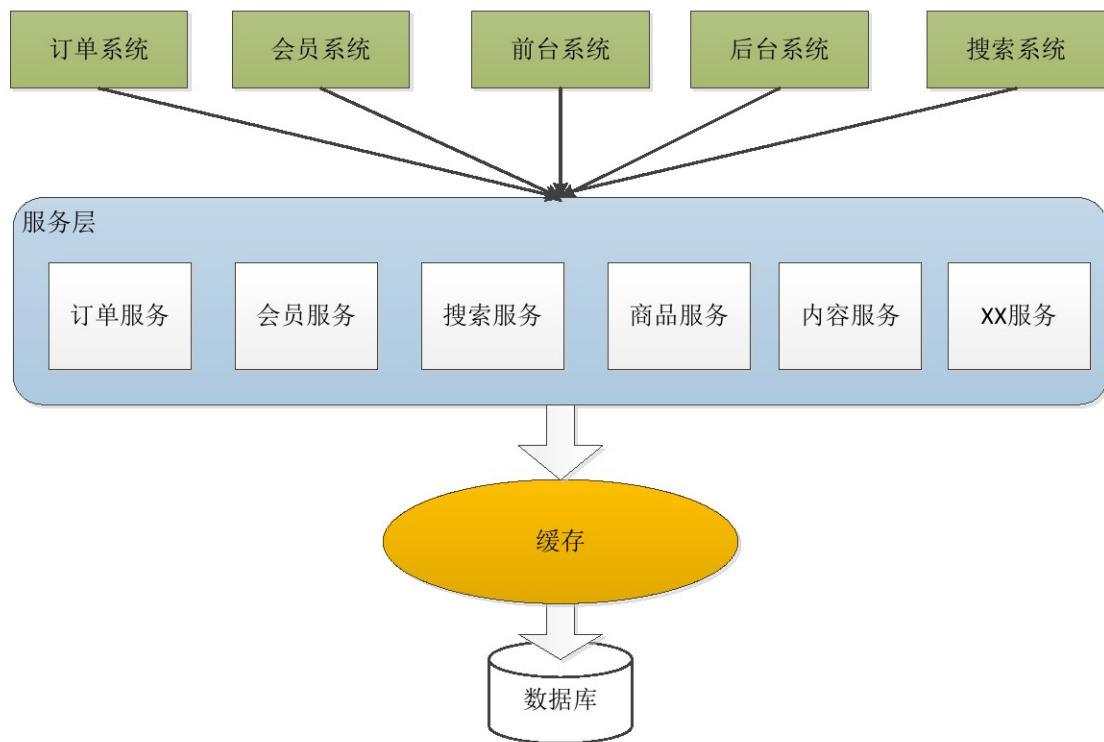
- 1、把模块拆分，使用接口通信，降低模块之间的耦合度。
- 2、把项目拆分成若干个子项目，不同的团队负责不同的子项目。
- 3、增加功能时只需要再增加一个子项目，调用其他系统的接口就可以。
- 4、可以灵活的进行分布式部署。

缺点：

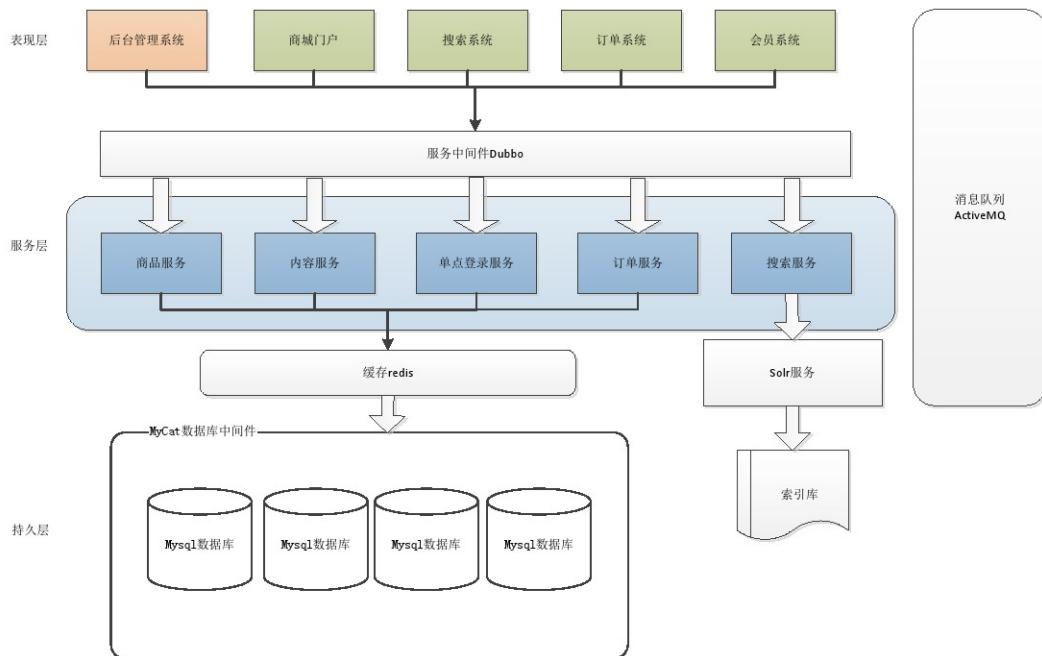
- 1、系统之间交互需要使用远程通信，接口开发增加工作量。
- 2、各个模块有一些通用的业务逻辑无法共用。

4.2.4. 基于 soa 的架构

SOA: Service Oriented Architecture 面向服务的架构。也就是把工程拆分成服务层、表现层两个工程。服务层中包含业务逻辑，只需要对外提供服务即可。表现层只需要处理和页面的交互，业务逻辑都是调用服务层的服务来实现。



4.2.5. 淘淘商城系统架构



5. 工程搭建

5.1. 使用 maven 的好处

使用 maven 管理工程。

Jar 包的管理

工程之间的依赖管理

自动打包

5.2. 后台工程搭建分析

Maven 的常见打包方式：jar、war、pom

Pom 工程一般都是父工程，管理 jar 包的版本、maven 插件的版本、统一的依赖管理。聚合工程。

Taotao-parent: 父工程，打包方式 pom，管理 jar 包的版本号。

| 项目中所有工程都应该继承父工程。

|--Taotao-common: 通用的工具类通用的 pojo。打包方式 jar

|--Taotao-manager: 服务层工程。聚合工程。Pom 工程

 |--taotao-manager-dao: 打包方式 jar

 |--taotao-manager-pojo: 打包方式 jar

 |--taotao-manager-interface: 打包方式 jar

 |--taotao-manager-service: 打包方式: war

 |--taotao-manager-web: 表现层工程。打包方式 war

5.3. 工程搭建

5.3.1. Taotao-parent

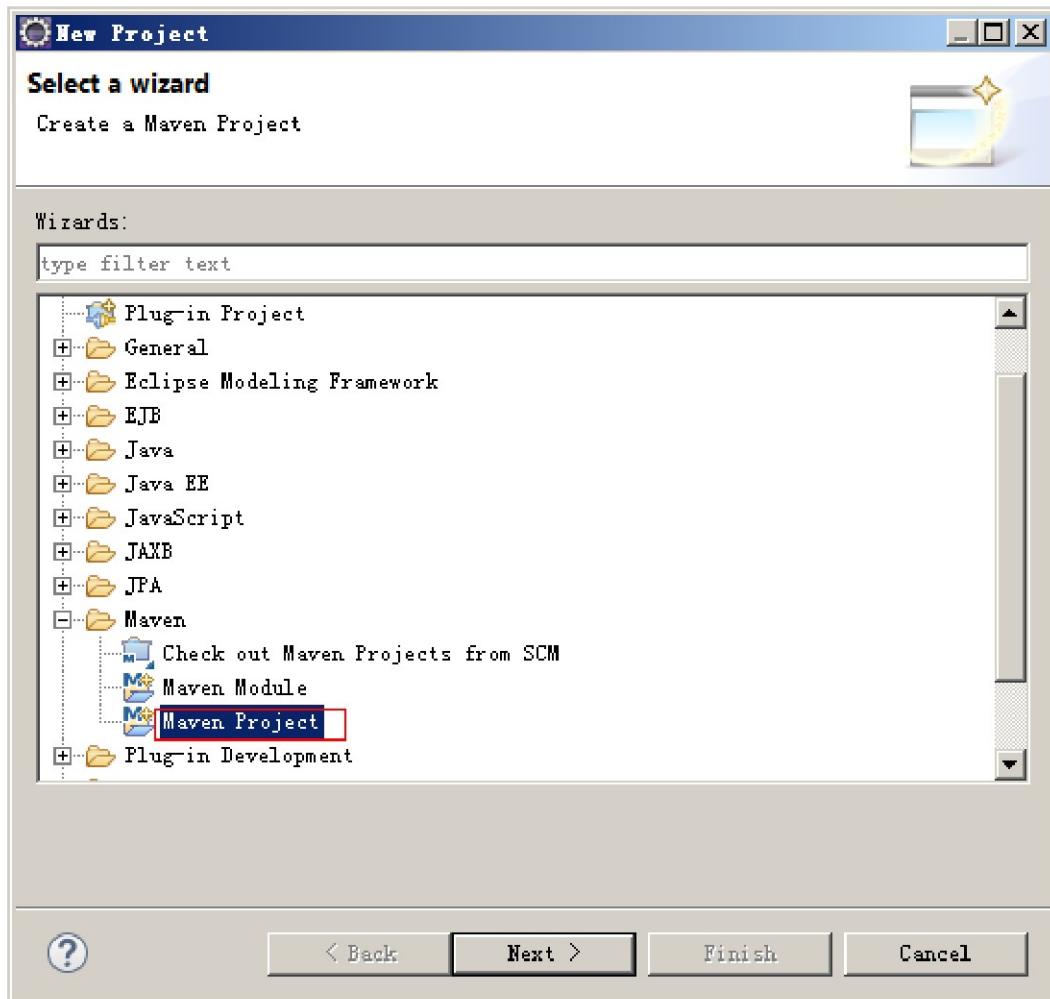
Pom 工程。

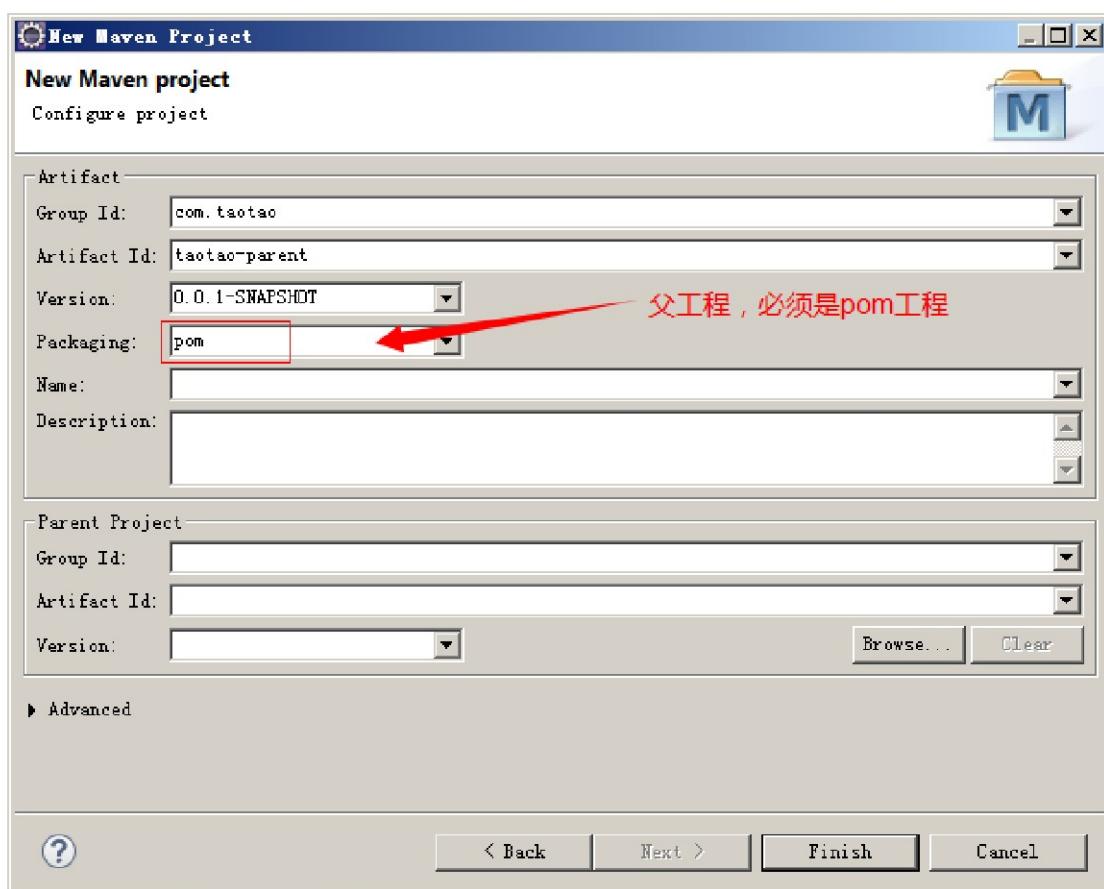
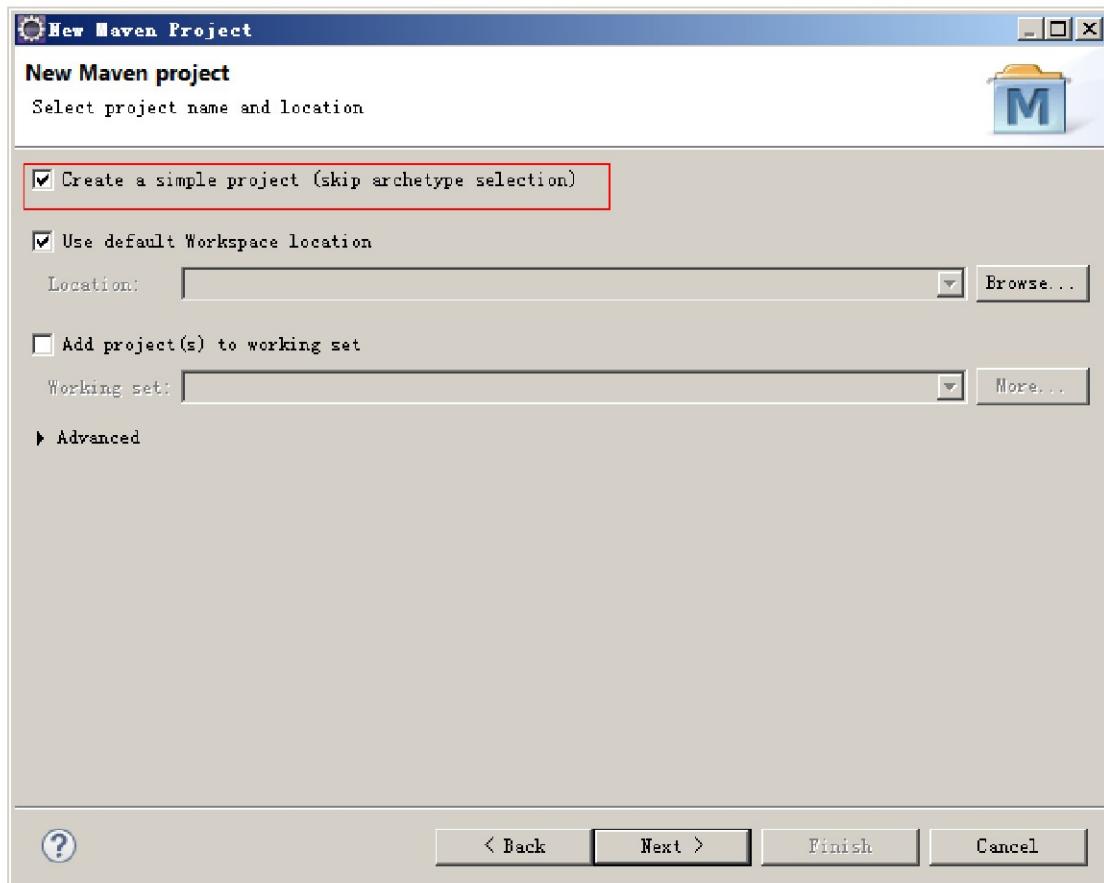
开发环境：eclipse mars2

Maven: 3.3.9

本地仓库：默认位置在~/.m2/repository

Eclipse 不需要配置，只需要把本地仓库放到当前用户目录下的.m2 下。





Pom 文件

```
<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.taobao</groupId>
  <artifactId>taobao-parent</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>pom</packaging>
  <!-- 集中定义依赖版本号 -->
  <properties>
    <junit.version>4.12</junit.version>
    <spring.version>4.2.4.RELEASE</spring.version>
    <mybatis.version>3.2.8</mybatis.version>
    <mybatis.spring.version>1.2.2</mybatis.spring.version>
    <mybatis.paginator.version>1.2.15</mybatis.paginator.version>
    <mysql.version>5.1.32</mysql.version>
    <slf4j.version>1.6.4</slf4j.version>
    <jackson.version>2.4.2</jackson.version>
    <druid.version>1.0.9</druid.version>
    <httpclient.version>4.3.5</httpclient.version>
    <jstl.version>1.2</jstl.version>
    <servlet-api.version>2.5</servlet-api.version>
    <jsp-api.version>2.0</jsp-api.version>
    <joda-time.version>2.5</joda-time.version>
    <commons-lang3.version>3.3.2</commons-lang3.version>
    <commons-io.version>1.3.2</commons-io.version>
    <commons-net.version>3.3</commons-net.version>
    <pagehelper.version>3.4.2-fix</pagehelper.version>
    <jsqlparser.version>0.9.1</jsqlparser.version>
    <commons-fileupload.version>1.3.1</commons-fileupload.version>
    <jedis.version>2.7.2</jedis.version>
    <solrj.version>4.10.3</solrj.version>
    <dubbo.version>2.5.3</dubbo.version>
    <zookeeper.version>3.4.7</zookeeper.version>
    <zkclient.version>0.1</zkclient.version>
    <activemq.version>5.11.2</activemq.version>
    <freemarker.version>2.3.23</freemarker.version>
    <quartz.version>2.2.2</quartz.version>
```

```
</properties>

<dependencyManagement>
    <dependencies>
        <!-- 时间操作组件 -->
        <dependency>
            <groupId>joda-time</groupId>
            <artifactId>joda-time</artifactId>
            <version>${joda-time.version}</version>
        </dependency>
        <!-- Apache 工具组件 -->
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-lang3</artifactId>
            <version>${commons-lang3.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-io</artifactId>
            <version>${commons-io.version}</version>
        </dependency>
        <dependency>
            <groupId>commons-net</groupId>
            <artifactId>commons-net</artifactId>
            <version>${commons-net.version}</version>
        </dependency>
        <!-- Jackson Json 处理工具包 -->
        <dependency>
            <groupId>com.fasterxml.jackson.core</groupId>
            <artifactId>jackson-databind</artifactId>
            <version>${jackson.version}</version>
        </dependency>
        <!-- httpclient -->
        <dependency>
            <groupId>org.apache.httpcomponents</groupId>
            <artifactId>httpclient</artifactId>
            <version>${httpclient.version}</version>
        </dependency>
        <!-- quartz 任务调度框架 -->
        <dependency>
            <groupId>org.quartz-scheduler</groupId>
            <artifactId>quartz</artifactId>
            <version>${quartz.version}</version>
        </dependency>
        <!-- 单元测试 -->
    </dependencies>
</dependencyManagement>
```

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
</dependency>
<!-- 日志处理 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${slf4j.version}</version>
</dependency>
<!-- Mybatis -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>${mybatis.version}</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>${mybatis.spring.version}</version>
</dependency>
<dependency>
    <groupId>com.github.miemiedev</groupId>
    <artifactId>mybatis-paginator</artifactId>
    <version>${mybatis.paginator.version}</version>
</dependency>
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
    <version>${pagehelper.version}</version>
</dependency>
<!-- MySql -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>${mysql.version}</version>
</dependency>
<!-- 连接池 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
    <version>${druid.version}</version>
</dependency>
```

```
</dependency>
<!-- Spring -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-beans</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jms</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
    <version>${spring.version}</version>
</dependency>
<!-- JSP 相关 -->
<dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>${jstl.version}</version>
</dependency>
<dependency>
```



```
<groupId>javax.servlet</groupId>
<artifactId>servlet-api</artifactId>
<version>${servlet-api.version}</version>
<scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jsp-api</artifactId>
    <version>${jsp-api.version}</version>
    <scope>provided</scope>
</dependency>
<!-- 文件上传组件 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>${commons-fileupload.version}</version>
</dependency>
<!-- Redis 客户端 -->
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>${jedis.version}</version>
</dependency>
<!-- solr 客户端 -->
<dependency>
    <groupId>org.apache.solr</groupId>
    <artifactId>solr-solrj</artifactId>
    <version>${solrj.version}</version>
</dependency>
<!-- dubbo 相关 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>dubbo</artifactId>
    <version>${dubbo.version}</version>
</dependency>
<dependency>
    <groupId>org.apache.zookeeper</groupId>
    <artifactId>zookeeper</artifactId>
    <version>${zookeeper.version}</version>
</dependency>
<dependency>
    <groupId>com.github.sgroschupf</groupId>
    <artifactId>zkclient</artifactId>
    <version>${zkclient.version}</version>
</dependency>
```

```
</dependency>

<dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-all</artifactId>
    <version>${activemq.version}</version>
</dependency>

<dependency>
    <groupId>org.freemarker</groupId>
    <artifactId>freemarker</artifactId>
    <version>${freemarker.version}</version>
</dependency>

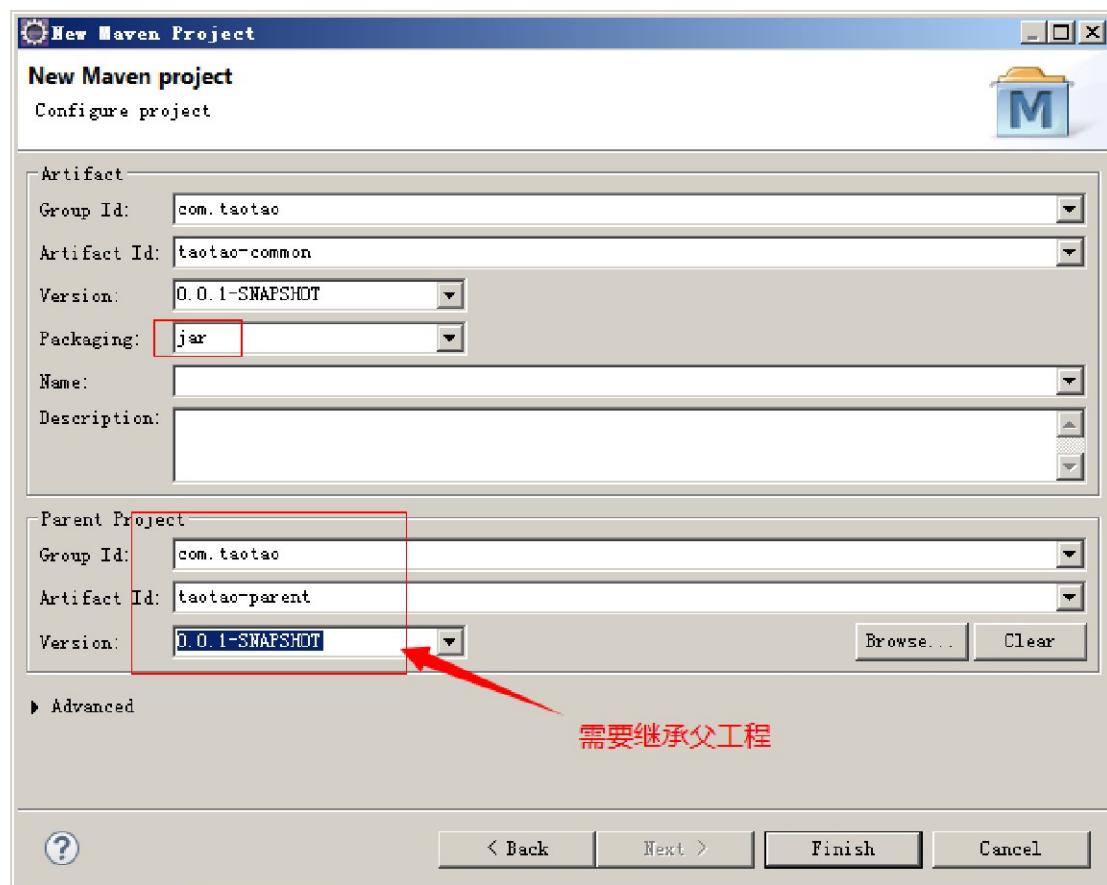
</dependencies>
</dependencyManagement>

<build>
    <finalName>${project.artifactId}</finalName>
    <plugins>
        <!-- 资源文件拷贝插件 -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <version>2.7</version>
            <configuration>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
        <!-- java 编译插件 -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.2</version>
            <configuration>
                <source>1.7</source>
                <target>1.7</target>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
    </plugins>
    <pluginManagement>
        <plugins>
            <!-- 配置 Tomcat 插件 -->
            <plugin>
                <groupId>org.apache.tomcat.maven</groupId>
```

```
<artifactId>tomcat7-maven-plugin</artifactId>
<version>2.2</version>
</plugin>
</plugins>
</pluginManagement>
</build>
</project>
```

5.3.2. Taotao-common

通用的工具类、通用的 pojo。打包方式 jar。需要继承父工程。



Pom 文件

```
<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>
```

```
<parent>
    <groupId>com.taobao</groupId>
    <artifactId>taobao-parent</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</parent>
<groupId>com.taobao</groupId>
<artifactId>taobao-common</artifactId>
<version>0.0.1-SNAPSHOT</version>
<!-- 依赖工具类的 jar 包 -->
<dependencies>
    <!-- 时间操作组件 -->
    <dependency>
        <groupId>joda-time</groupId>
        <artifactId>joda-time</artifactId>
    </dependency>
    <!-- Apache 工具组件 -->
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-lang3</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-io</artifactId>
    </dependency>
    <dependency>
        <groupId>commons-net</groupId>
        <artifactId>commons-net</artifactId>
    </dependency>
    <!-- Jackson Json 处理工具包 -->
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
    </dependency>
    <!-- httpclient -->
    <dependency>
        <groupId>org.apache.httpcomponents</groupId>
        <artifactId>httpclient</artifactId>
    </dependency>
    <!-- quartz 任务调度框架 -->
    <dependency>
        <groupId>org.quartz-scheduler</groupId>
        <artifactId>quartz</artifactId>
    </dependency>
    <!-- 单元测试 -->
```

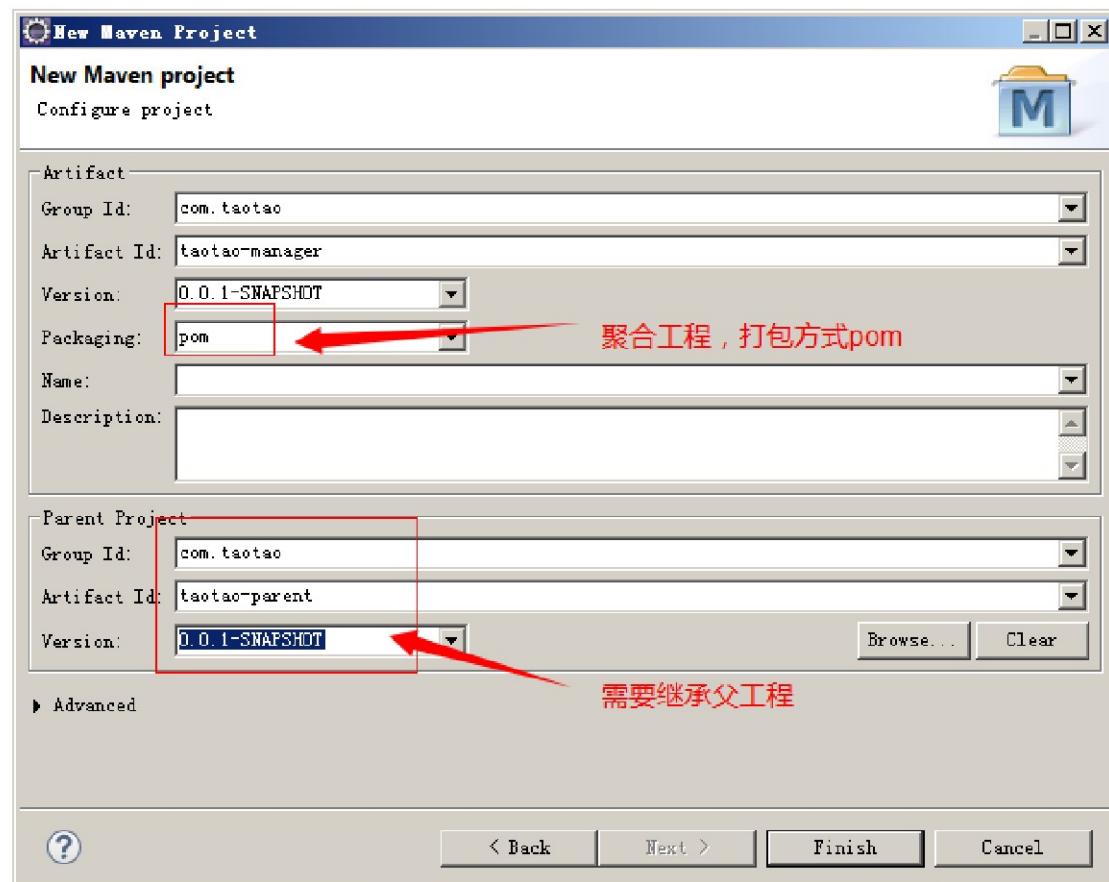
```

<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
</dependency>
<!-- 日志处理 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
</dependency>
</dependencies>
</project>

```

5.3.3. Taobao-manager

聚合工程。打包方式 pom。



Pom 文件

```

<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>

<parent>

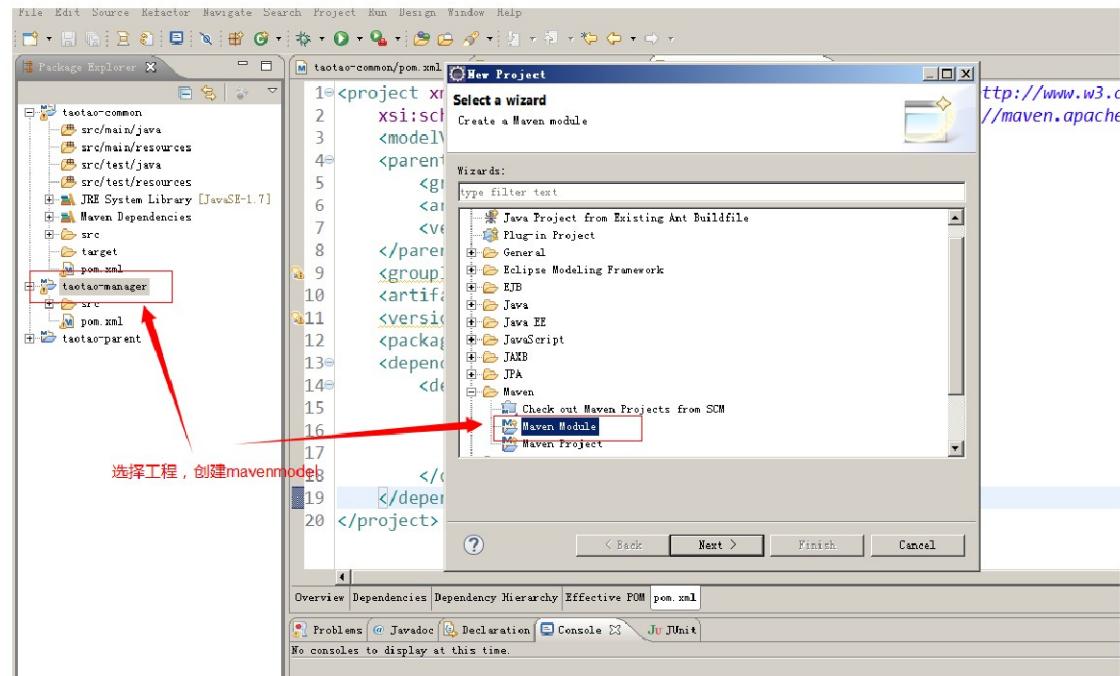
    <groupId>com.taobao</groupId>
    <artifactId>taobao-parent</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</parent>

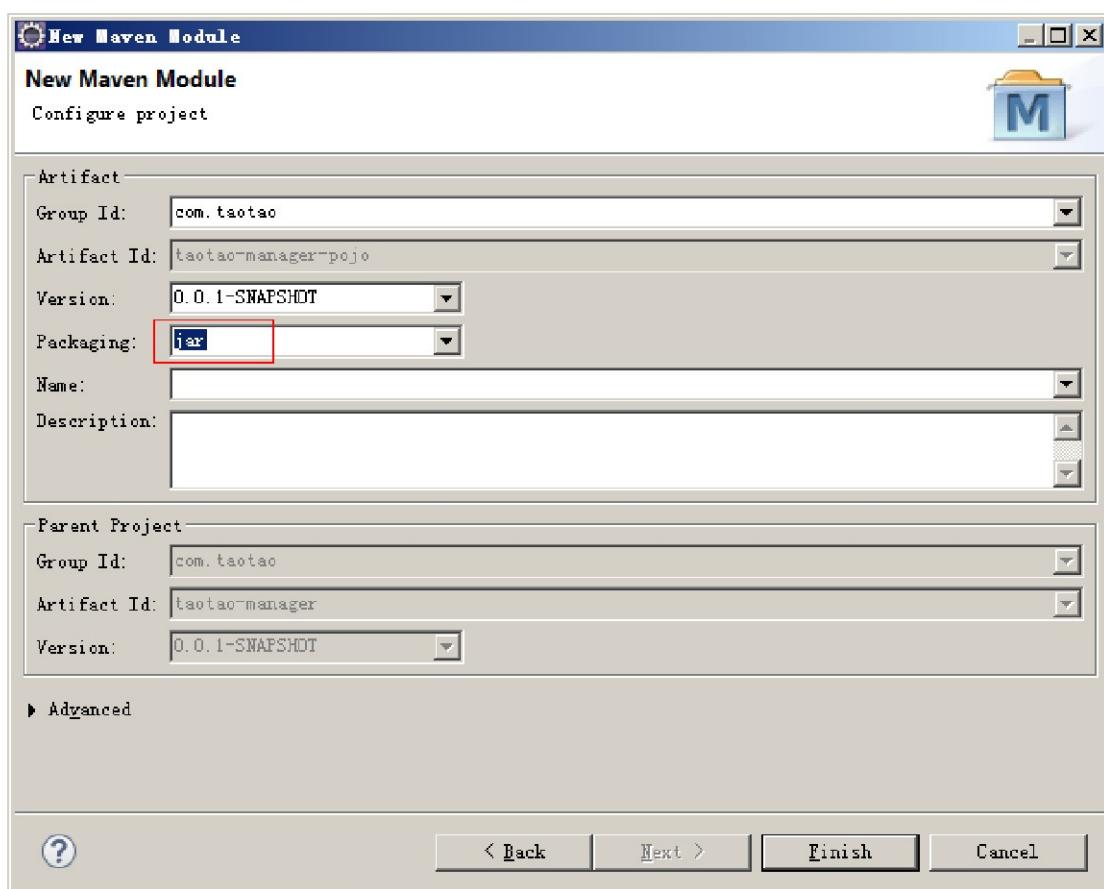
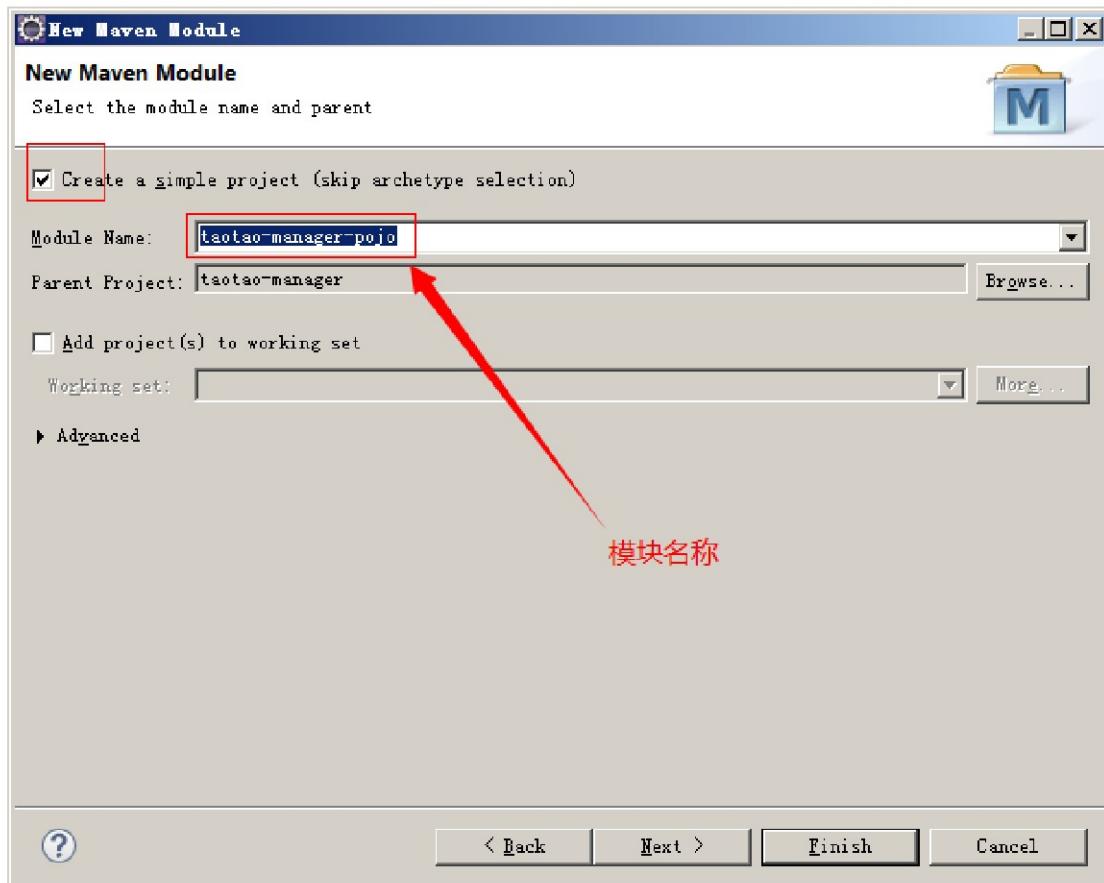
<groupId>com.taobao</groupId>
<artifactId>taobao-manager</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>pom</packaging>
<dependencies>

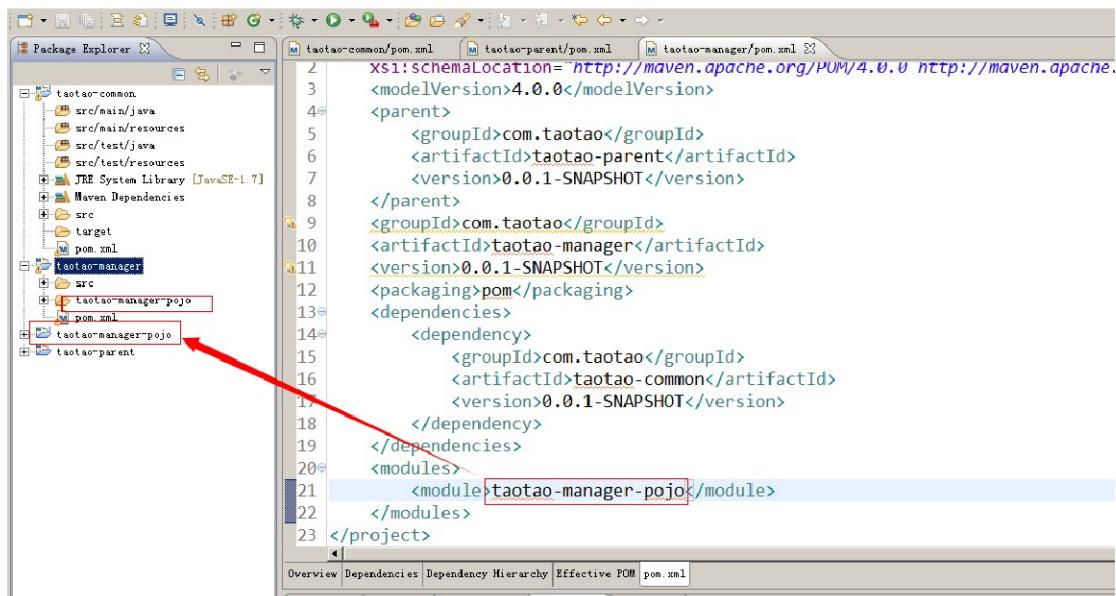
    <dependency>
        <groupId>com.taobao</groupId>
        <artifactId>taobao-common</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
</dependencies>
</project>
```

5.3.4. Taobao-manager-pojo

是一个 maven 模块，打包方式 jar。





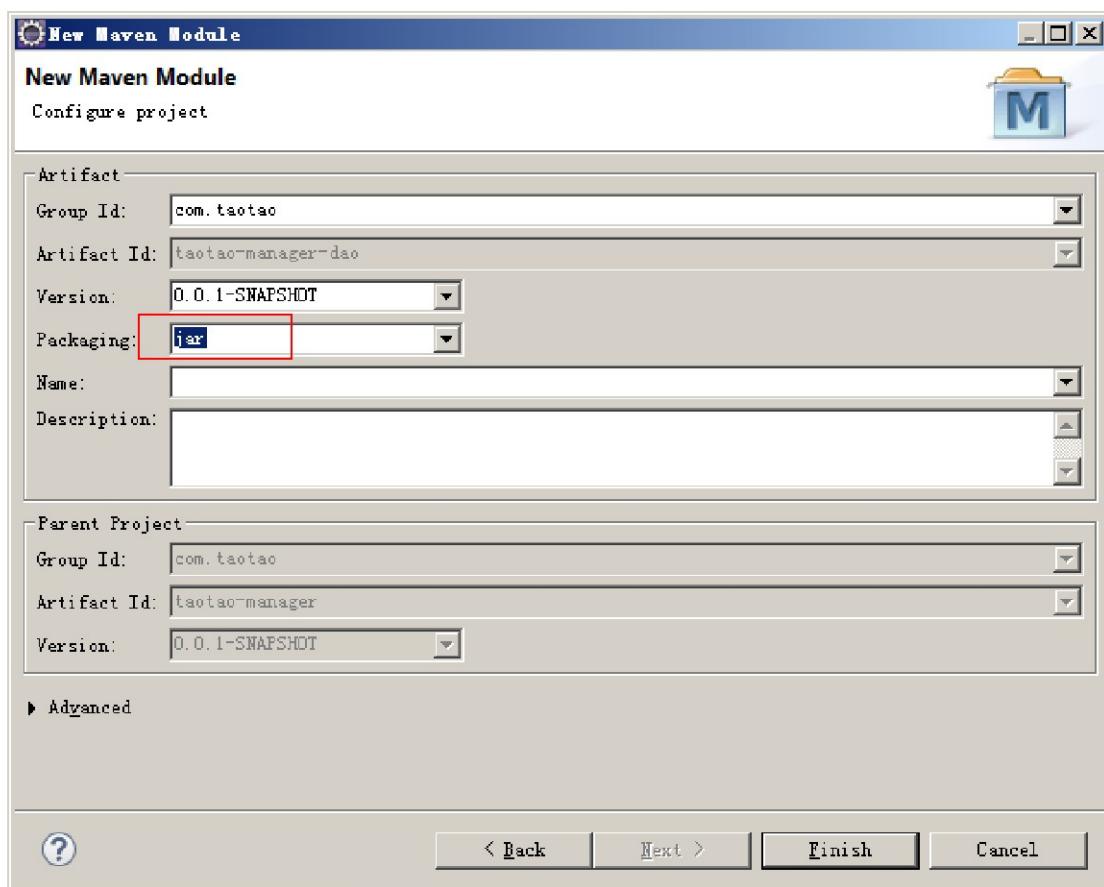
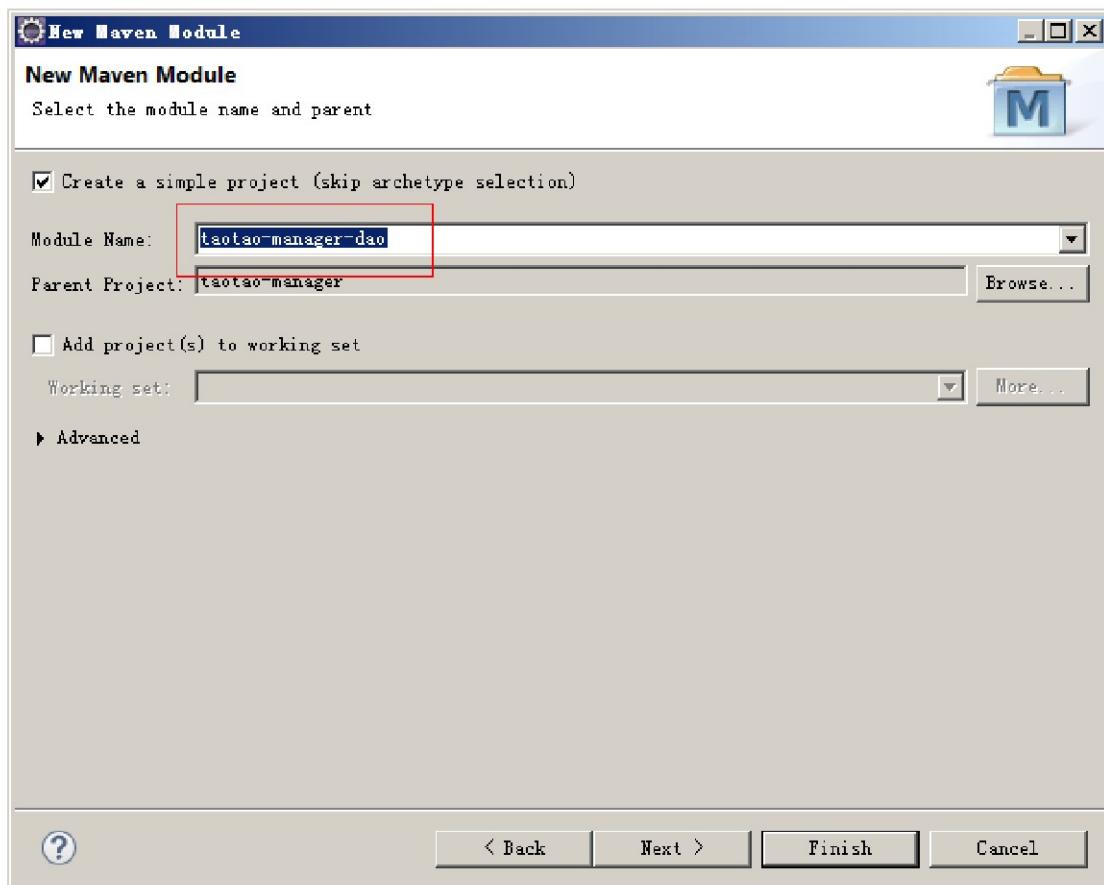


Pom 文件

```
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>com.taotao</groupId>
    <artifactId>taotao-manager</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <artifactId>taotao-manager-pojo</artifactId>
</project>
```

5.3.5. Taotao-manager-dao

Maven 模块，打包方式 jar。



Pom 文件

```
<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">

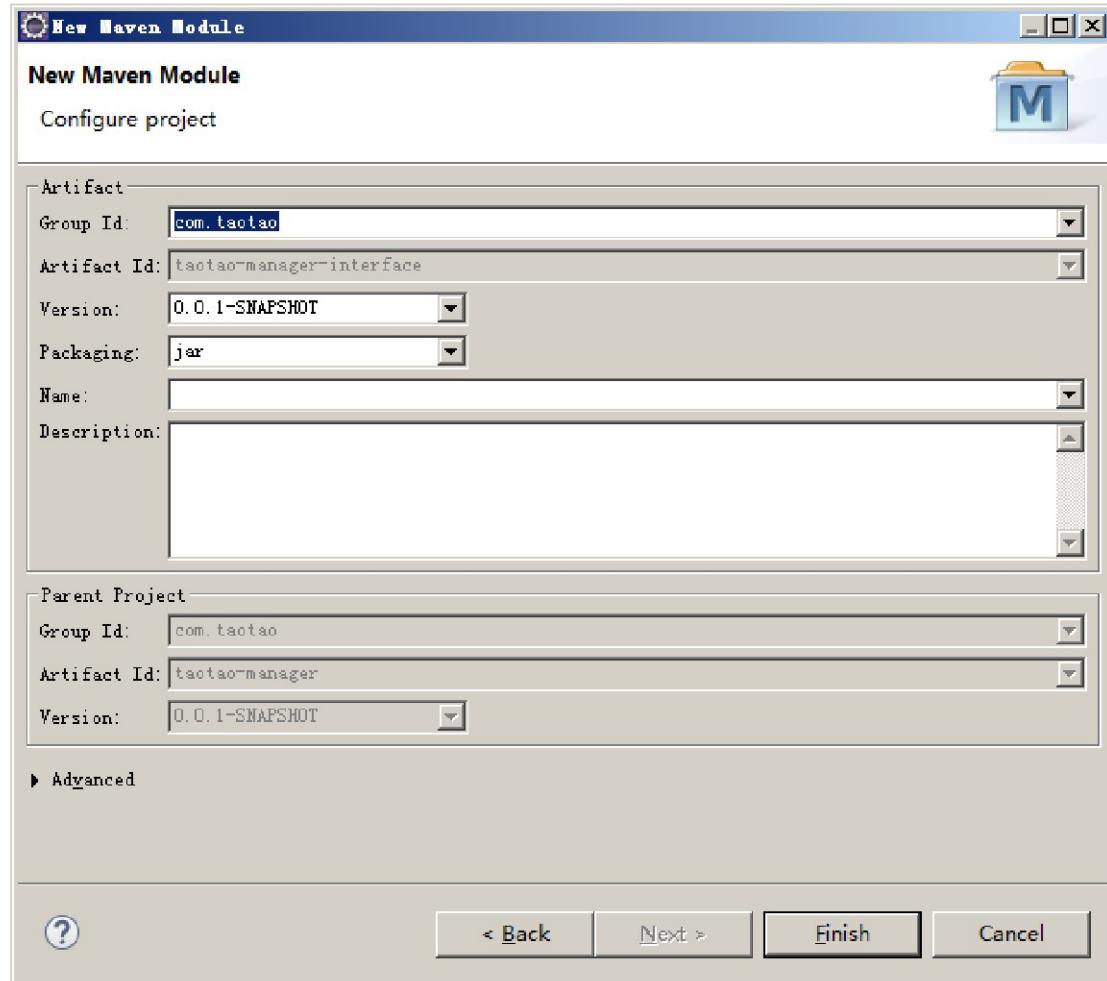
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>com.taobao</groupId>
    <artifactId>taobao-manager</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <artifactId>taobao-manager-dao</artifactId>
  <!-- 依赖的 jar 包 -->
  <dependencies>

    <dependency>
      <groupId>com.taobao</groupId>
      <artifactId>taobao-manager-pojo</artifactId>
      <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <!-- Mybatis -->
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
    </dependency>
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis-spring</artifactId>
    </dependency>
    <dependency>
      <groupId>com.github.miemiedev</groupId>
      <artifactId>mybatis-paginator</artifactId>
    </dependency>
    <dependency>
      <groupId>com.github.pagehelper</groupId>
      <artifactId>pagehelper</artifactId>
    </dependency>
    <!-- MySql -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
    </dependency>
    <!-- 连接池 -->
  
```

```
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
</dependency>
</dependencies>
</project>
```

5.3.6. Taotao-manager-interface



Pom 文件

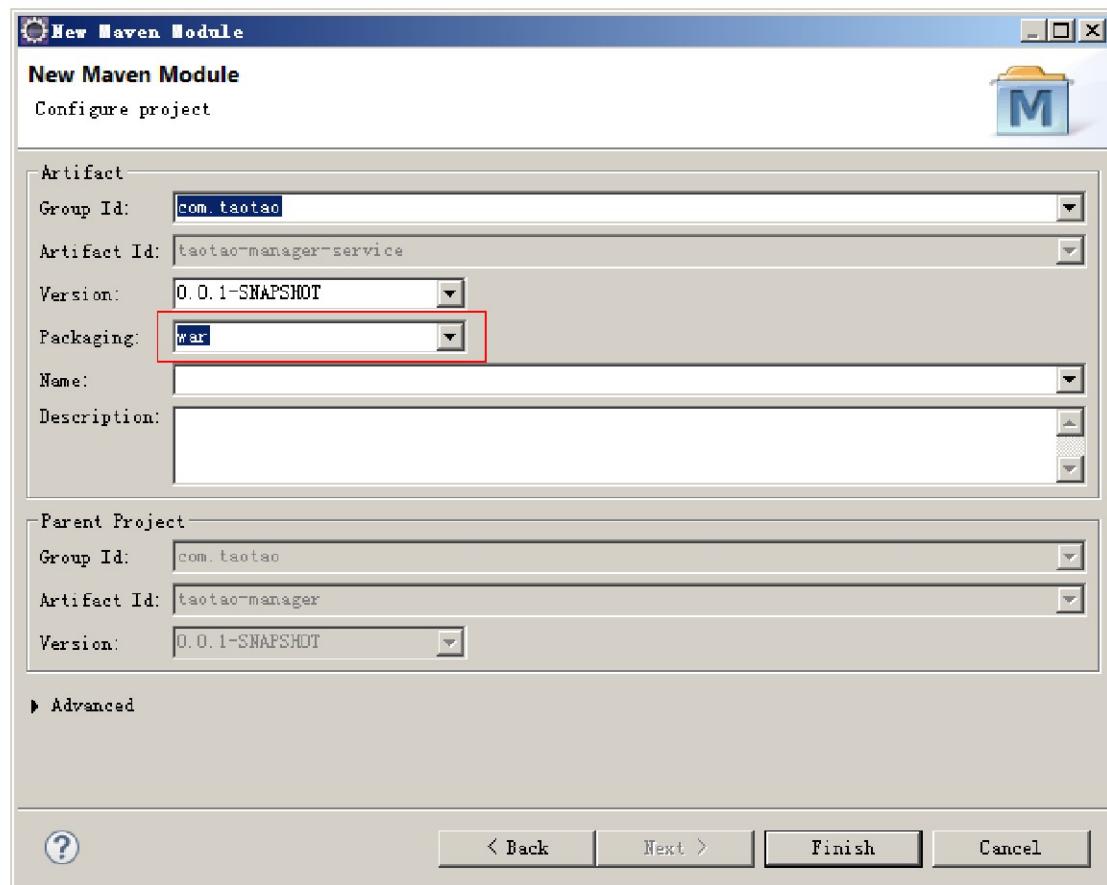
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.taotao</groupId>
```



```
<artifactId>taotao-manager</artifactId>
<version>0.0.1-SNAPSHOT</version>
</parent>
<artifactId>taotao-manager-interface</artifactId>
<dependencies>
    <dependency>
        <groupId>com.taotao</groupId>
        <artifactId>taotao-manager-pojo</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
</dependencies>
</project>
```

5.3.7. Taotao-manager-service

打包方式 war 包。



Pom 文件

```
<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>

<parent>
    <groupId>com.taobao</groupId>
    <artifactId>taobao-manager</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</parent>
<artifactId>taobao-manager-service</artifactId>
<packaging>war</packaging>
<dependencies>
    <dependency>
        <groupId>com.taobao</groupId>
        <artifactId>taobao-manager-dao</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <!-- spring 的依赖 -->
    <!-- Spring -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-beans</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aspects</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jms</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>

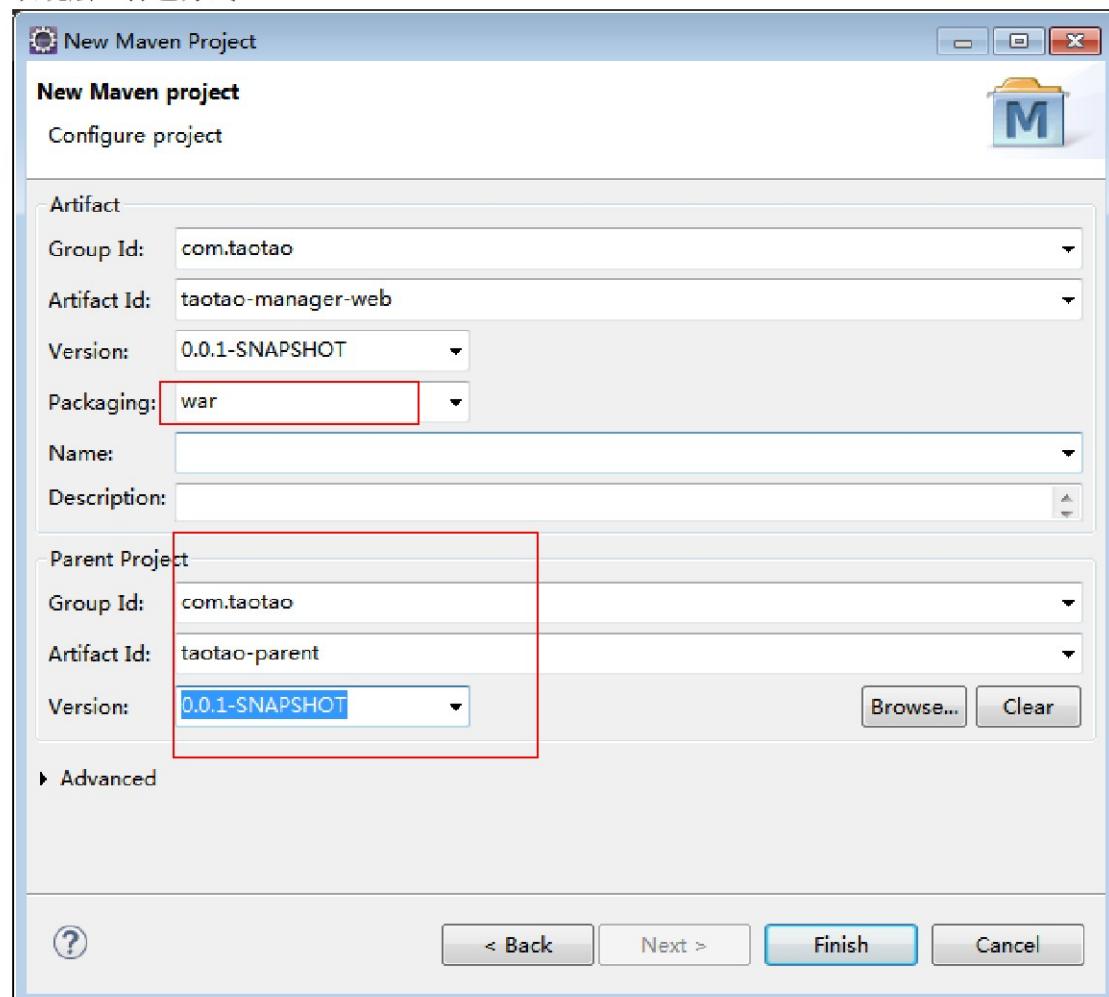
```



```
<artifactId>spring-context-support</artifactId>
</dependency>
</dependencies>
</project>
```

5.3.8. Taotao-manager-web

表现层，打包方式 war。



Pom 文件

```
<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>
  <parent>
    <groupId>com.taotao</groupId>
    <artifactId>taotao-parent</artifactId>
    <version>0.0.1-SNAPSHOT</version>
```

```
</parent>

<artifactId>taotao-manager-web</artifactId>

<packaging>war</packaging>

<dependencies>

    <!-- JSP 相关 -->

    <dependency>

        <groupId>jstl</groupId>
        <artifactId>jstl</artifactId>

    </dependency>

    <dependency>

        <groupId>javax.servlet</groupId>
        <artifactId>servlet-api</artifactId>
        <scope>provided</scope>

    </dependency>

    <dependency>

        <groupId>javax.servlet</groupId>
        <artifactId>jsp-api</artifactId>
        <scope>provided</scope>

    </dependency>

</dependencies>

</project>
```

5.4. 启动工程

启动 taotao-manager 工程。

需要在 taotao-manager 的 pom 工程中，配置 tomcat 插件。启动的端口号，和工程名称。

在 taotao-manager 的 pom 文件中添加如下配置：

```
<!-- 配置 tomcat 插件 -->

<build>

    <plugins>

        <plugin>

            <groupId>org.apache.tomcat.maven</groupId>
            <artifactId>tomcat7-maven-plugin</artifactId>

            <configuration>

                <port>8080</port>
                <path>/</path>

            </configuration>

        </plugin>

    </plugins>

</build>
```

启动 tomcat 插件：

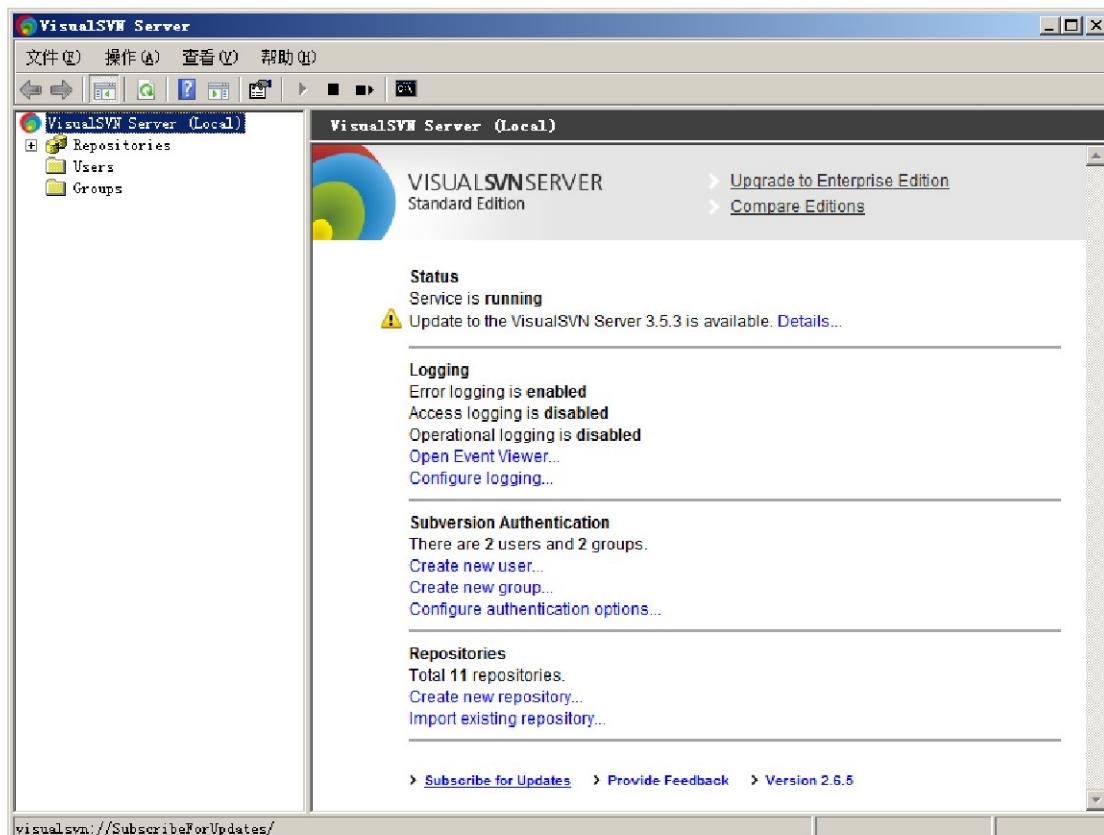
clean tomcat7:run

先把 taotao-parent、taotao-common 安装到本地仓库。然后再启动。

6. Svn 的使用

6.1. 服务端

服务端使用 VisualSVN，一般一个项目组只有一个 svn，并不是每个开发者都需要安装服务端。



6.2. 客户端

客户端使用 Eclipse 的 svn 插件，在提供的 Eclipse 中已经安装好，直接使用即可。

<https://{svn 服务 ip 地址}/{仓库名称}/>

用户名: user

密码: user

<https://PC-201311301552/svn/taotao-javaee25/>