

# 1. 课程目标

目标 1：了解电商行业特点以及理解电商的模式

目标 2：了解整体品优购的架构特点

目标 3：能够运用 Dubbo+SSM 搭建分布式应用

目标 4：搭建工程框架，完成品牌列表后端代码

## 2. 电商概述

### 2.1. 电商行业分析

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。





## 2.2. 电商行业技术特点

- ✧ 技术新
- ✧ 技术范围广
- ✧ 分布式（一件事情，拆开来做）
- ✧ 高并发、集群（同样的事情，大家一起做）、负载均衡、高可用
- ✧ 海量数据
- ✧ 业务复杂
- ✧ 系统安全

## 2.3. 主要电商模式

### 2.3.1. B2B--企业对企业

B2B（Business to Business）是指进行电子商务交易的供需双方都是商家（或企业、公司），她（他）们使用了互联网的技术或各种商务网络平台，完成商务交易的过程。电子商务是现代 B2B marketing 的一种具体主要的表现形式。



案例：阿里巴巴、慧聪网

### 2.3.2. C2C--个人对个人

C2C 即 Customer（Consumer） to Customer（Consumer），意思就是消费者个人间的电子商务行为。比如一个消费者有一台电脑，通过网络进行交易，把它出售给另外一个消费者，此种交易类型就称为 C2C 电子商务。



案例：淘宝、易趣、瓜子二手车

### 2.3.3. B2C--企业对个人

B2C 是 Business-to-Customer 的缩写，而其中文简称为“商对客”。“商对客”是电子商务的一种模式，也就是通常说的直接面向消费者销售产品和服务商业零售模式。这种形式的电子商务一般以网络零售业为主，主要借助于互联网开展在线销售活动。B2C 即企业通过互联网为消费者提供一个新型的购物环境——网上商店，消费者通过网络在网上购物、网上支付等消费行为。



案例：唯品会、乐蜂网

### 2.3.4. C2B--个人对企业

C2B (Consumer to Business, 即消费者到企业)，是互联网经济时代新的商业模式。这一模式改变了原有生产者（企业和机构）和消费者的关系，是一种消费者创造价值（Create Value），企业和机构消费价值（Consume Value）。

C2B 模式和我们熟知的供需模式（DSM, Demand Supply Model）恰恰相反，真正



的 C2B 应该先有消费者需求产生而后有企业生产，即先有消费者提出需求，后有生产企业按需求组织生产。通常情况为消费者根据自身需求定制产品价格，或主动参与产品设计、生产和定价，产品、价格等彰显消费者的个性化需求，生产企业进行定制化生产。



案例：海尔商城、尚品宅配

### 2.3.5. O2O--线上到线下

O2O 即 Online To Offline（在线离线/线上到线下），是指将线下的商务机会与互联网结合，让互联网成为线下交易的平台，这个概念最早来源于美国。O2O 的概念非常广泛，既可涉及到线上，又可涉及到线下,可以通称为 O2O。主流商业管理课程均对 O2O 这种新型的商业模式有所介绍及关注。



案例：美团、饿了么

### 2.3.6. F2C--工厂到个人

F2C 指的是 Factory to customer，即从厂商到消费者的电子商务模式。



### 2.3.7. B2B2C -企业-企业-个人

B2B2C 是一种电子商务类型的网络购物商业模式，B 是 BUSINESS 的简称，C 是 CUSTOMER 的简称，第一个 B 指的是商品或服务的供应商，第二个 B 指的是从事电子商务的企业，C 则表示消费者。

第一个 BUSINESS，并不仅仅局限于品牌供应商、影视制作公司和图书出版商，任何的商品供应商或服务供应商都能可以成为第一个 BUSINESS；第二 B 是 B2B2C 模式的电子商务企业，通过统一的经营管理对商品和服务、消费者终端同时进行整合，是广大供应商和消费者之间的桥梁，为供应商和消费者提供优质的服务，是互联网电子商务服务供应商。C 表示消费者，在第二个 B 构建的统一电子商务平台购物的消费者；

B2B2C 的来源于目前的 B2B、B2C 模式的演变和完善，把 B2C 和 C2C 完美地结合起来，通过 B2B2C 模式的电子商务企业构建自己的物流供应链系统，提供统一的服务。



案例：京东商城、天猫商城



## 3. 品优购需求分析与设计

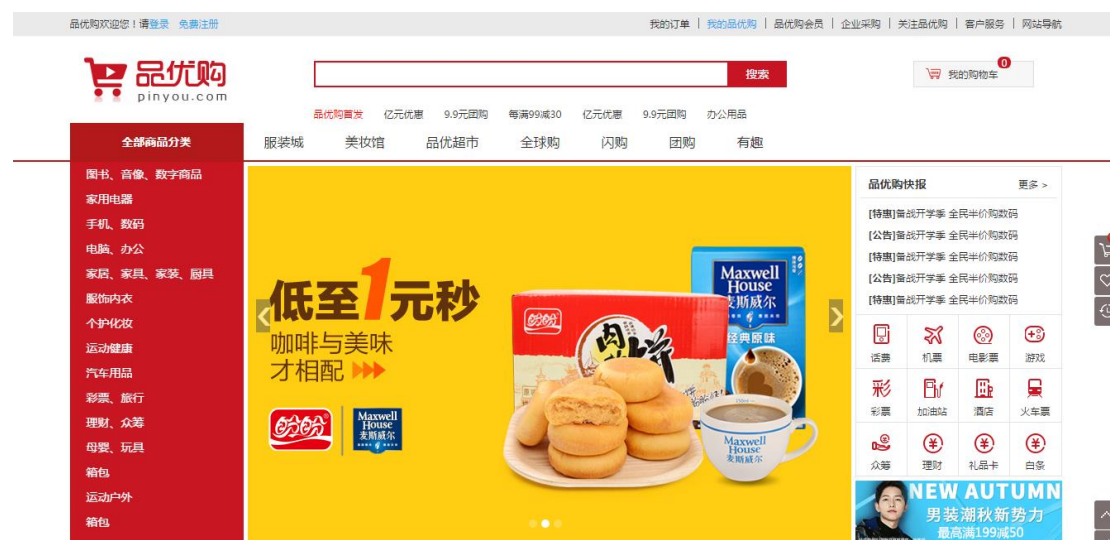
### 3.1. 品优购简介

品优购网上商城是一个综合性的 B2B2C 平台，类似京东商城、天猫商城。网站采用商家入驻的模式，商家入驻平台提交申请，有平台进行资质审核，审核通过后，商家拥有独立的管理后台录入商品信息。商品经过平台审核后即可发布。

品优购网上商城主要分为网站前台、运营商后台、商家管理后台三类系统。

#### 3.1.1. 网站前台

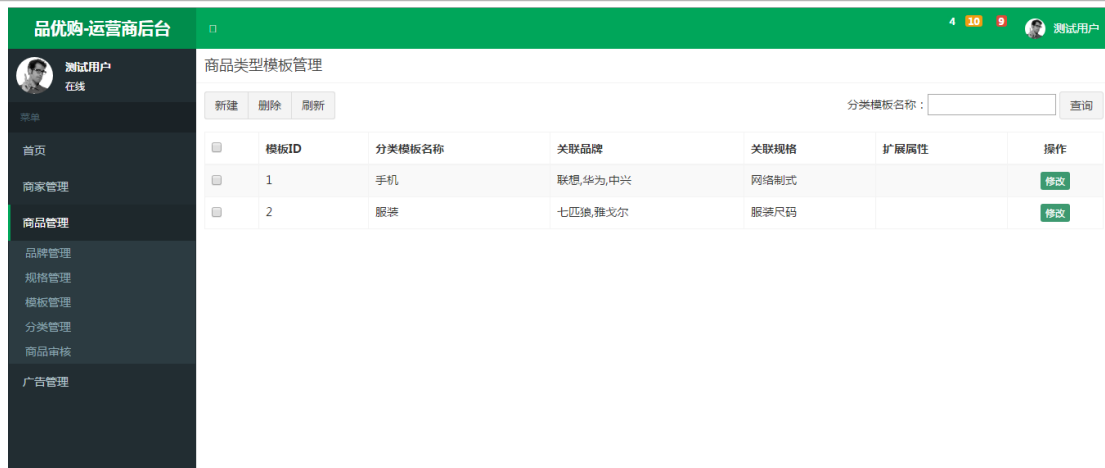
主要包括网站首页、商家首页、商品详情、搜索、会员中心、订单与支付相关页面、秒杀频道等。



#### 3.1.2. 运营商后台

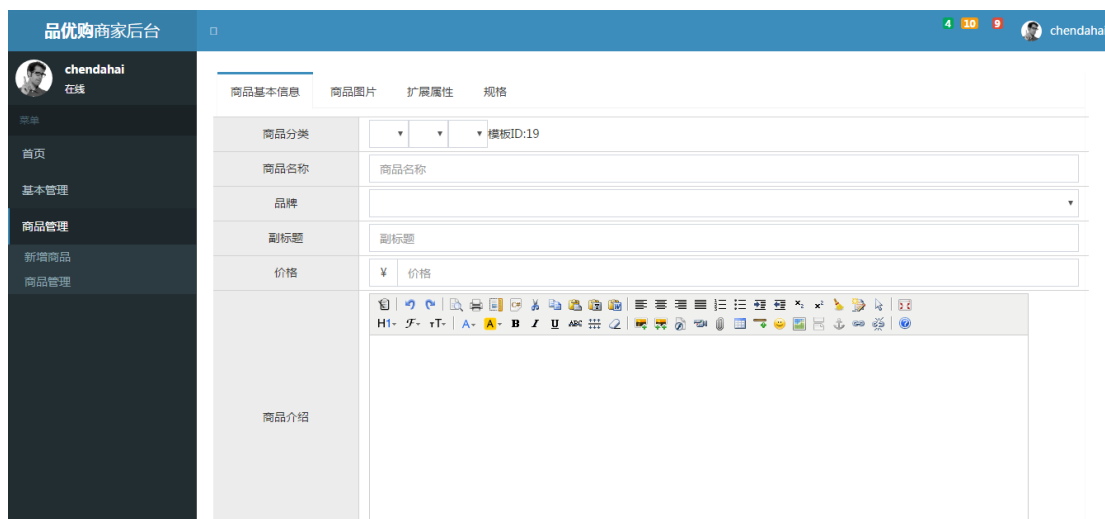
运营该电商平台的运营人员的管理后台。 主要包括商家审核、品牌管理、规格管理、模板管理、商品分类管理、商品审核、广告类型管理、广告管理、订单查询、商家结算等。





### 3.1.3. 商家管理后台

加入了该电商平台的商家进行管理的后台，主要功能是对商品的管理以及订单查询统计、资金结算等功能。



## 3.2. 系统架构

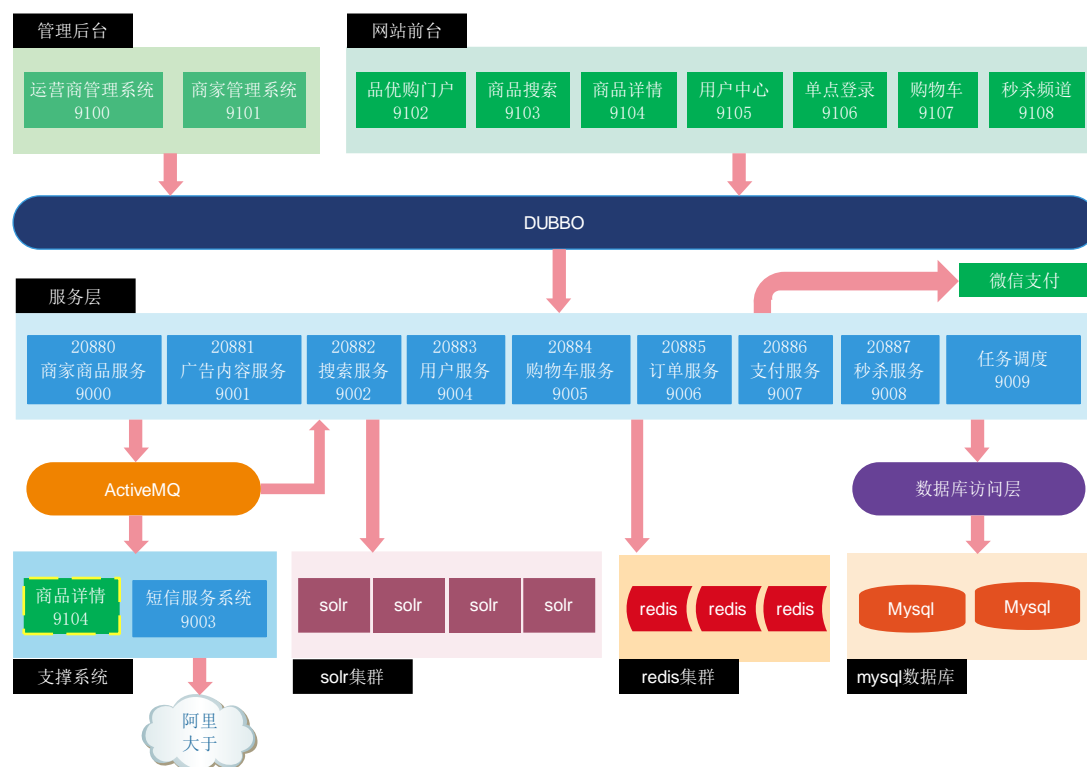
### 3.2.1. SOA 架构

SOA 是 Service-Oriented Architecture 的首字母简称，它是一种支持面向服务的架构样式，它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。服务层是 SOA 的基础，可以直接被应用调用，从而有效控制系统中与软件代理交互的人为依赖性。从服务、基于服务开发和服务的结果来看，

面向服务是一种思考方式。其实 SOA 架构更多应用于互联网项目开发。

为什么互联网项目会采用 SOA 架构呢？随着互联网的发展，网站应用的规模不断扩大，常规的垂直应用架构已无法应对，分布式服务架构以及流动计算架构势在必行，迫切需一个治理系统确保架构有条不紊的演进。

### 3.2.2. 品优购架构分析



### 3.3. 数据库表结构

表名称	含义
tb_brand	品牌
tb_specification	规格
tb_specification_option	规格选项
tb_type_template	类型模板：用于关联品牌和规格
tb_item_cat	商品分类
tb_seller	商家
tb_goods	商品
tb_goods_desc	商品详情



tb_item	商品明细
tb_content	内容（广告）
tb_content_category	内容（广告）类型
tb_user	用户
tb_order	订单
tb_order_item	订单明细
tb_pay_log	支付日志
tb_address	用户收货地址
tb_seckill_goods	秒杀商品
tb_seckill_order	秒杀订单

### 3.4. 项目框架

品优购采用当前流行的前后端分离架构。后端框架采用 Spring + Spring MVC + Mybatis +Dubbo。前端采用 angularJS + Bootstrap。

## 4. 分布式服务框架 Dubbo

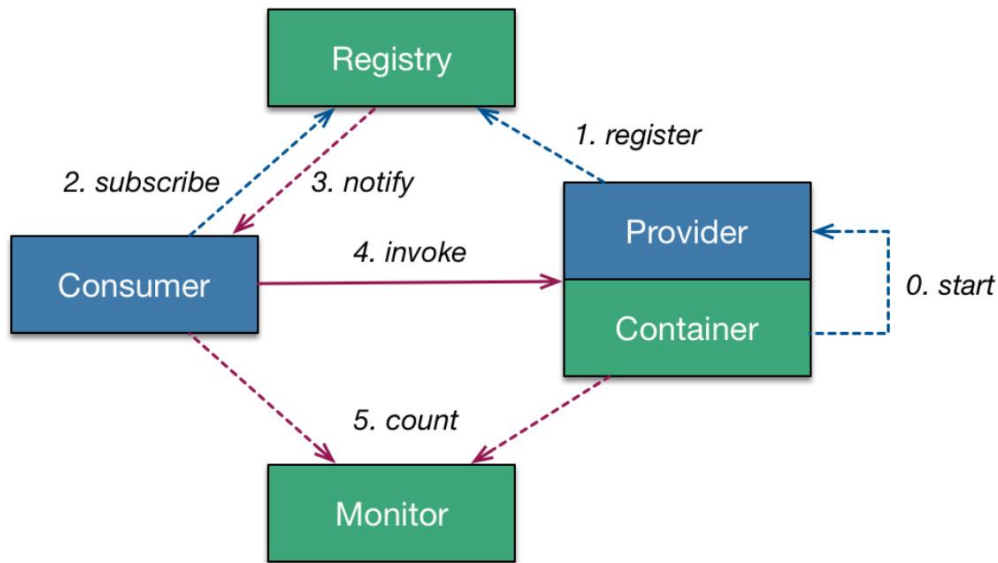
### 4.1. DUBBO 简介

DUBBO 是一个分布式服务框架，致力于提供高性能和透明化的 RPC 远程服务调用方案，是阿里巴巴 SOA (Service-Oriented Architecture, SOA, 面向服务架构) 服务化治理方案的核心框架，每天为 2,000+ 个服务提供 3,000,000,000+ 次访问量支持，并被广泛应用于阿里巴巴集团的各成员站点。官网地址：<http://dubbo.io/>



## Dubbo Architecture

-----> init    -.-.-.-> async    -> sync



### 节点角色说明：

Provider: 暴露服务的服务提供方。

Consumer: 调用远程服务的服务消费方。

Registry: 服务注册与发现的注册中心。

Monitor: 统计服务的调用次数和调用时间的监控中心。

Container: 服务运行容器。

### 调用关系说明：

0. 服务容器负责启动，加载，运行服务提供者。
1. 服务提供者在启动时，向注册中心注册自己提供的服务。
2. 服务消费者在启动时，向注册中心订阅自己所需的服务。
3. 注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。
4. 服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。
5. 服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。

【软负载均衡】软件负载均衡则是通过在服务器上安装的特定的负载均衡软件或是自带负载均衡模块完成对请求的分配派发。如：轮询法、随机法、源地址哈希法、最小连接数法等。在消费方中声明服务的时候可以指定负载均衡的策略，dubbo 在返回的服务地址列表中使用负载均衡策略选择一个服务地址；默认是使用随机法。

## 4.2. 注册中心 zookeeper

### 4.2.1. zookeeper 简介

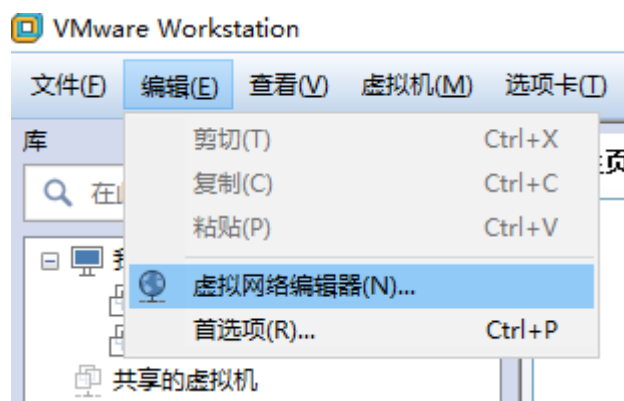
官方推荐使用 zookeeper 注册中心。注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者和消费者只在启动时与注册中心交互，注册中心不转发请求，压力较小。

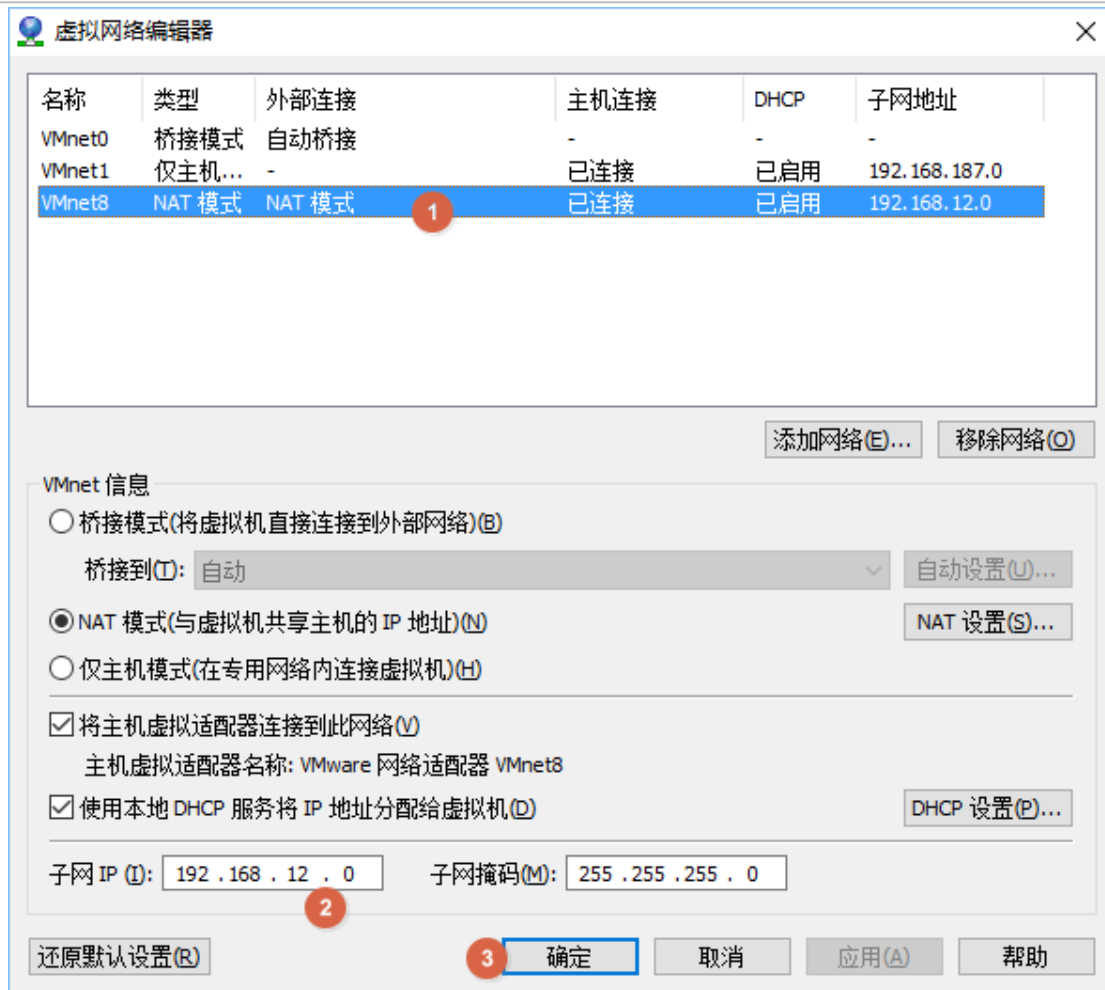
zookeeper 是 Apache Hadoop 的子项目，是一个树型的目录服务，支持变更推送，适合作为 Dubbo 服务的注册中心，工业强度较高，可用于生产环境。

zooKeeper 用来做 dubbo 的注册中心，也即哪一个服务由哪一台机器来提供必需让调用者知道，简单来说就是 ip 地址和服务名称的对应关系。

### 4.2.2. 准备 Linux 环境

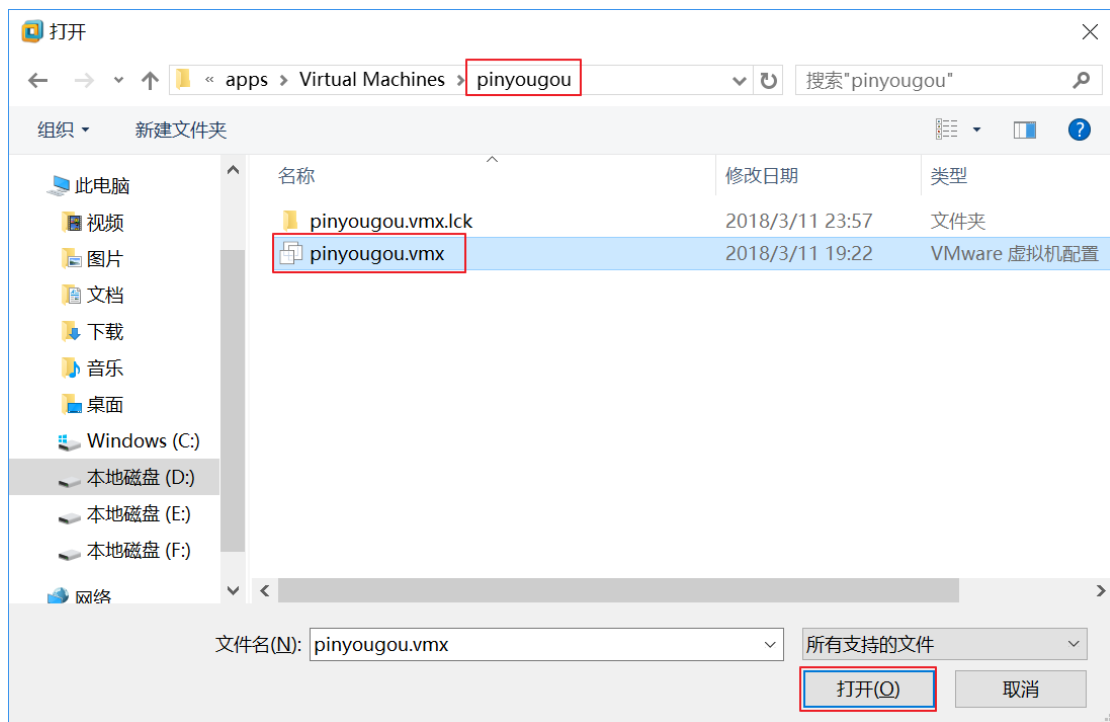
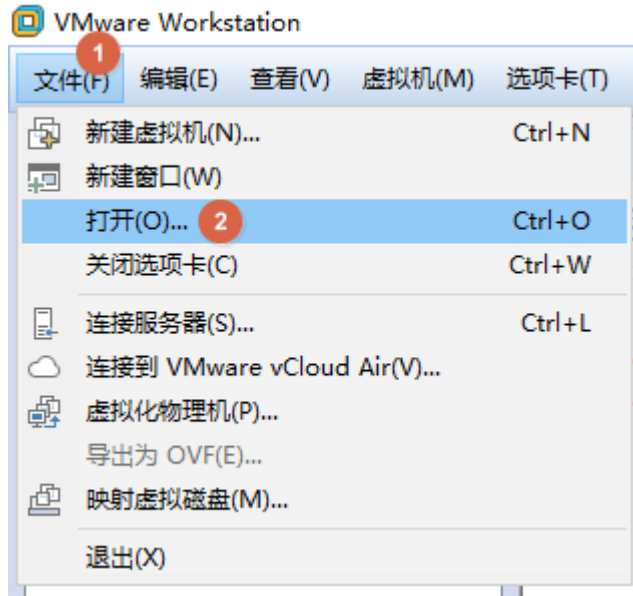
导入的虚拟机必须设置 ip 为 192.168.12.168 才可以使用 FastDFS(后面的课程会用)；如果虚拟机不是 12 网段，需要手动设置网段，设置方式如下图：





### 4.2.3. VMWare 中打开虚拟机

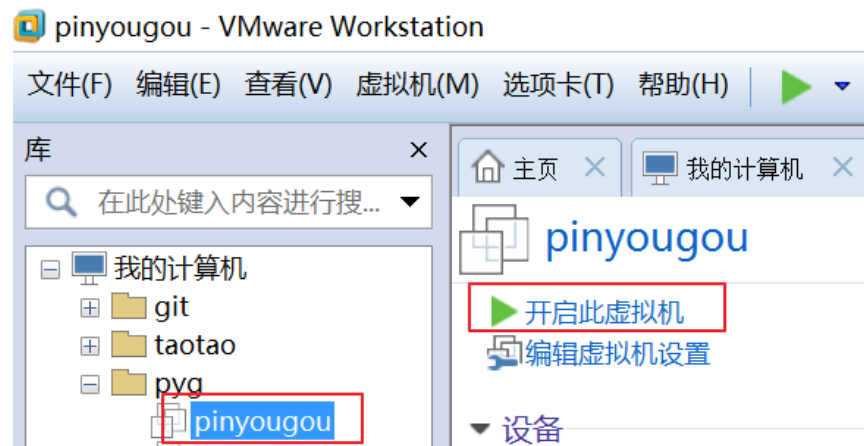
找到下发的虚拟机文件夹“pinyougou”；导入 VMWare 内，步骤如下：



如果出现弹出框：那么选择 我已移动该虚拟机



## 4.2.4. 启动虚拟机



启动成功；则如下

```
CentOS Linux 7 (Core)
Kernel 3.10.0-693.el7.x86_64 on an x86_64

pinyougou login: _
```

上述安装的 CentOS7 64 位系统；

帐号：root，密码：123456

【注意】如果启动后无法在 windows 中使用工具连接（systemctl start network.service 命令可以查看状态）；那么可以在 centos 中重启网络；输入如下命令：

```
systemctl stop NetworkManager
systemctl restart network.service
```

## 4.2.5. Linux 中安装 zookeeper

详见“资料\zookeeper\ Linux 安装 zookeeper.docx”

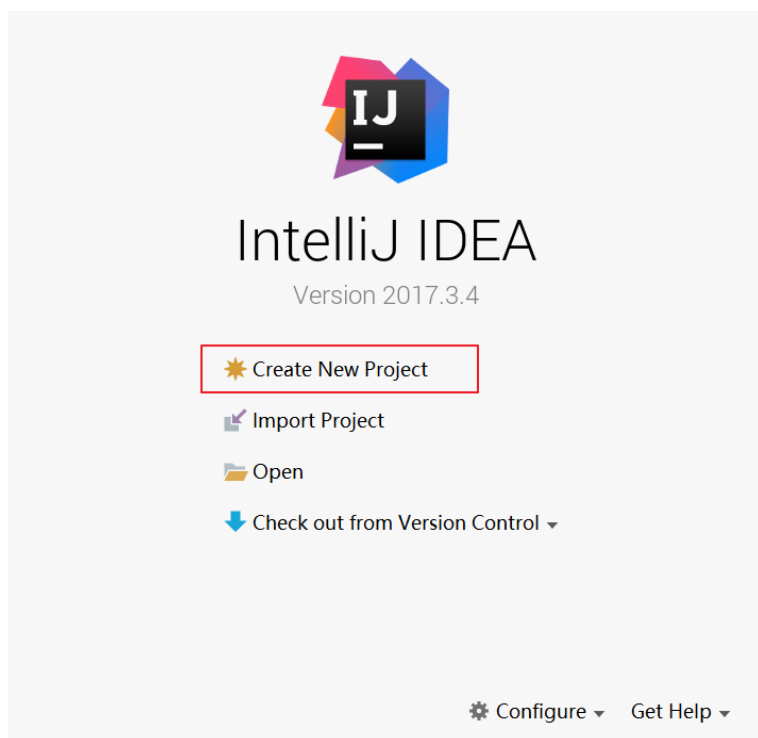
## 4.3. 监控中心 Monitor

在开发时，需要知道注册中心都注册了哪些服务，以便我们开发和测试。可以通过部署一个监控中心来实现。其实监控中心就是一个 web 应用，部署到 tomcat 即可。

监控中心 Monitor 的安装详见“资料\dubbo\安装部署 dubbo 的监控中心 monitor.docx”

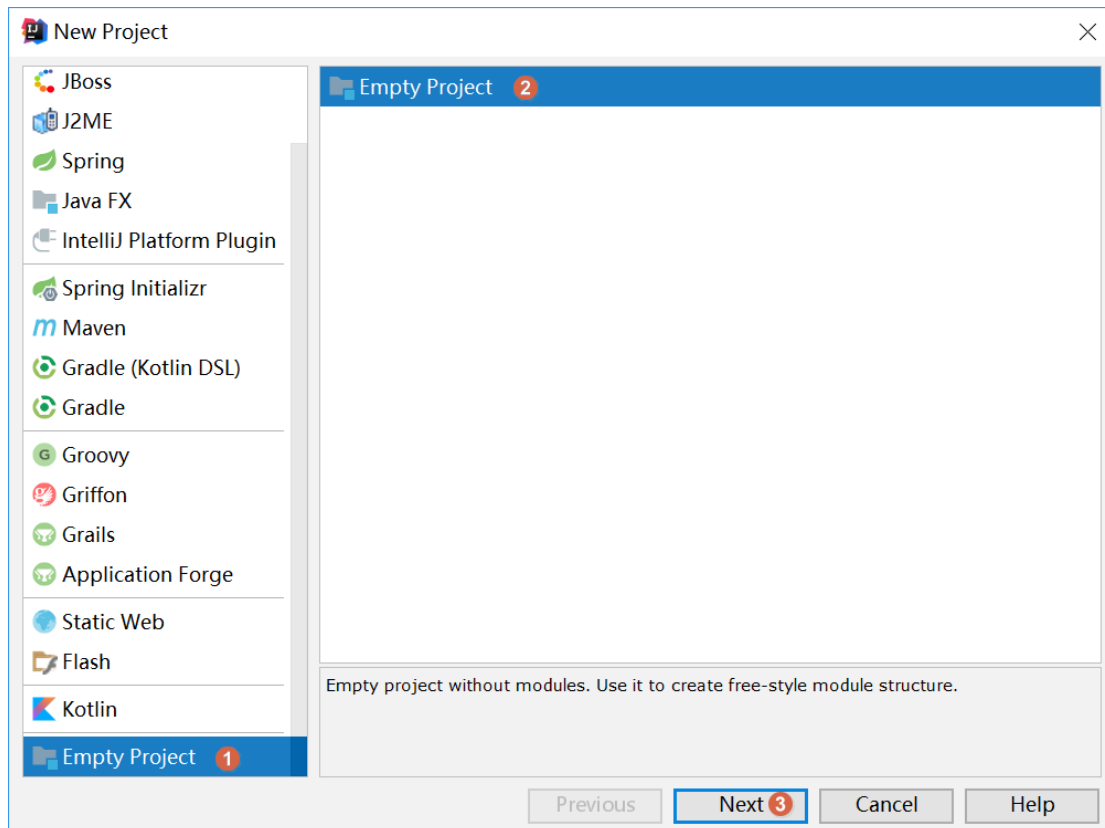
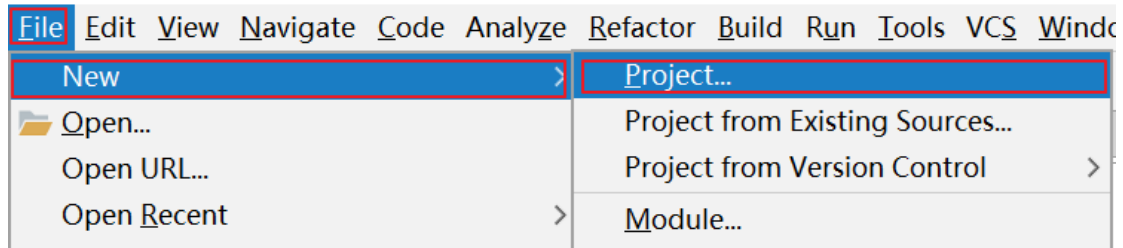
# 5. IDEA 开发环境设置

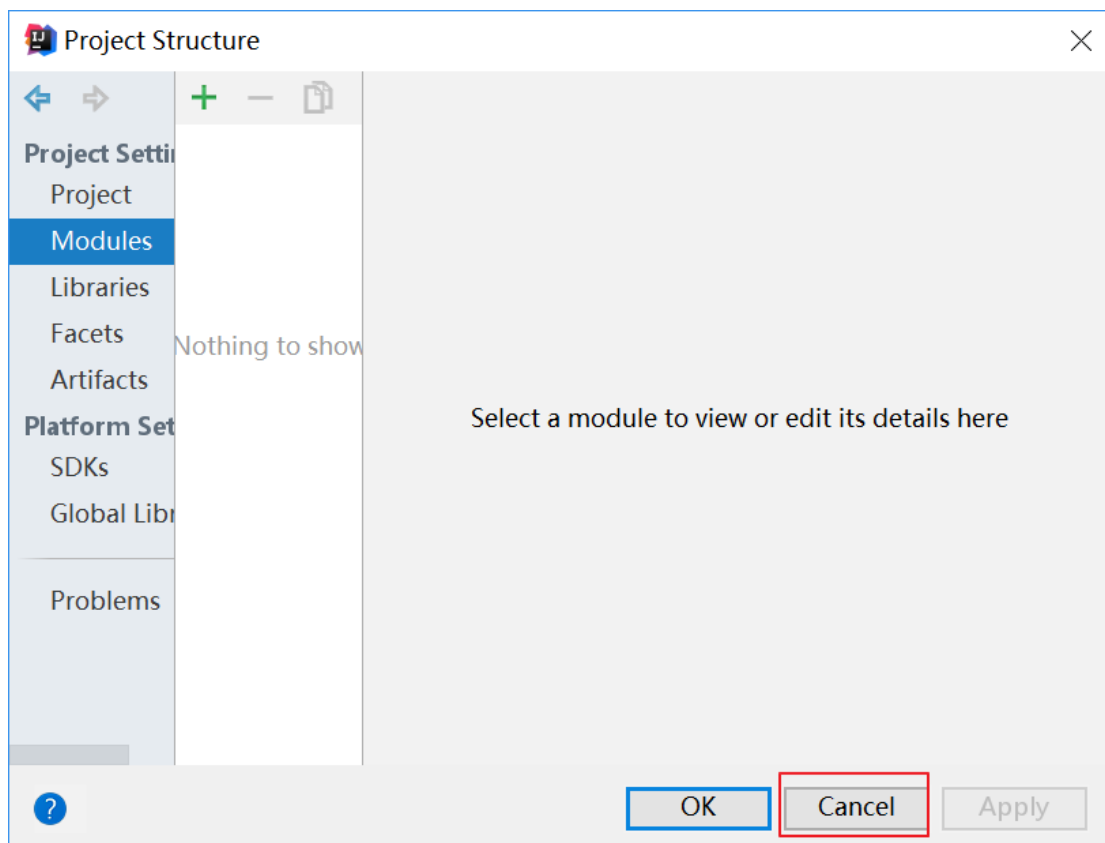
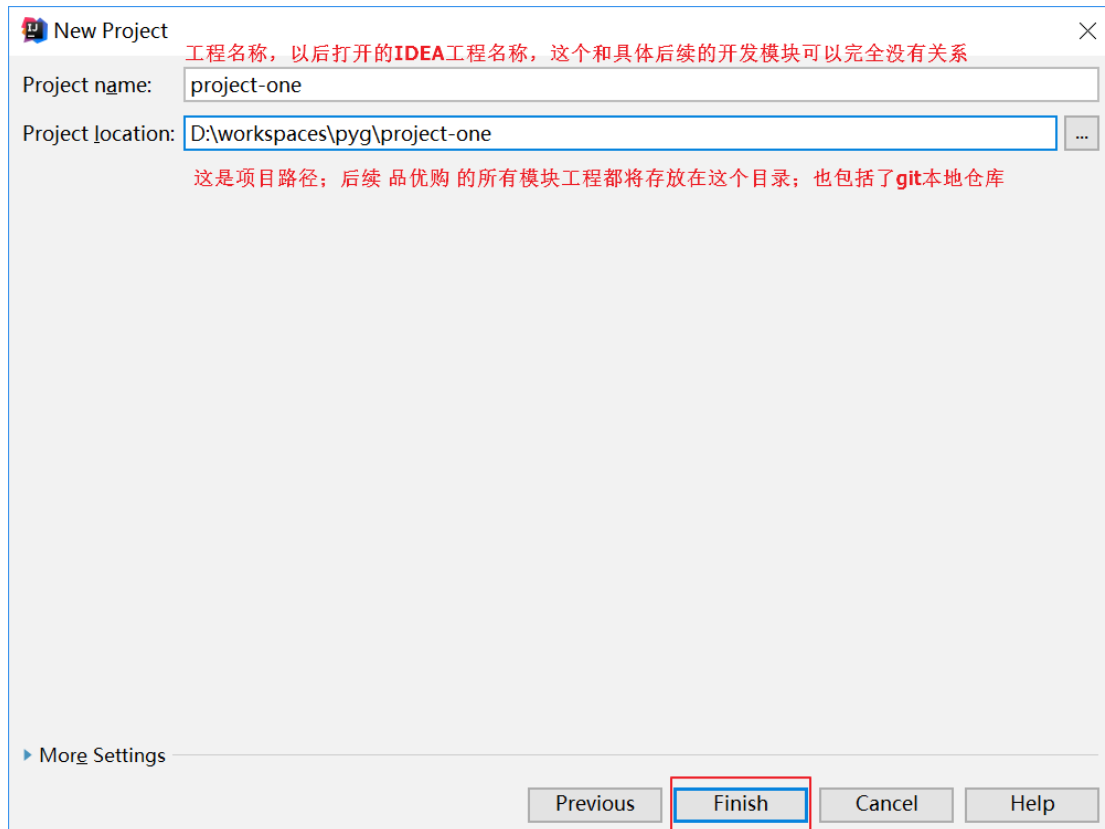
## 5.1. 创建 IDEA 工程



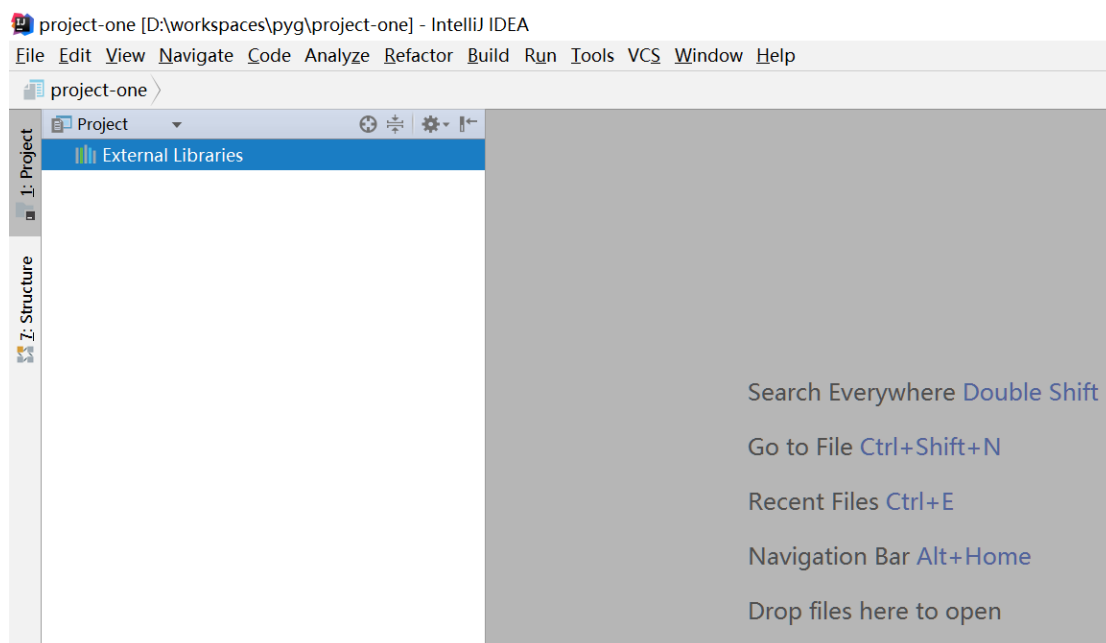
或者







创建空模块的项目效果如下：



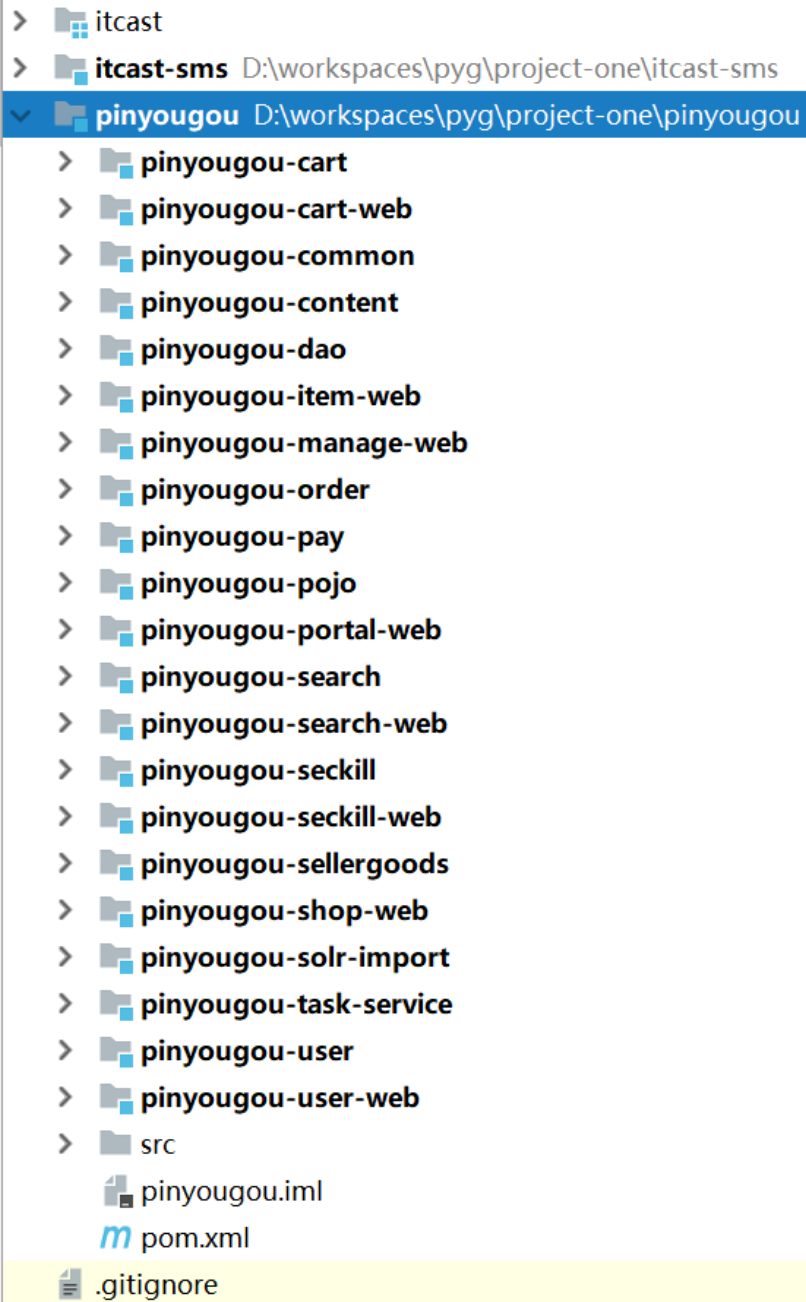
## 5.2. IDEA 工程环境设置

打开“资料\idea\IntelliJ IDEA 常用操作说明.pdf”文档；请按照第 2、3、12 章节设置自己的 IDEA 开发环境。

# 6. 品优购-框架搭建

## 6.1. 工程结构说明

开发完成后的工程结构如下：



工程说明：

pinyougou-parent 聚合父工程

pinyougou-pojo 通用实体类

pinyougou-dao 通用数据访问

pinyougou-xxx 业务服务层工程

pinyougou-xxx-interface 业务服务层接口

pinyougou-xxx-service 业务服务层实现（web 工程）



pinyougou-xxx-web     展现（控制）层工程（web 工程）

## 6.2. 创建数据库表

在导入的虚拟机里面已经有使用 mysql 5.7 的数据库名称为：pinyougodb，mysql5.7 的用户名和密码都为：root。

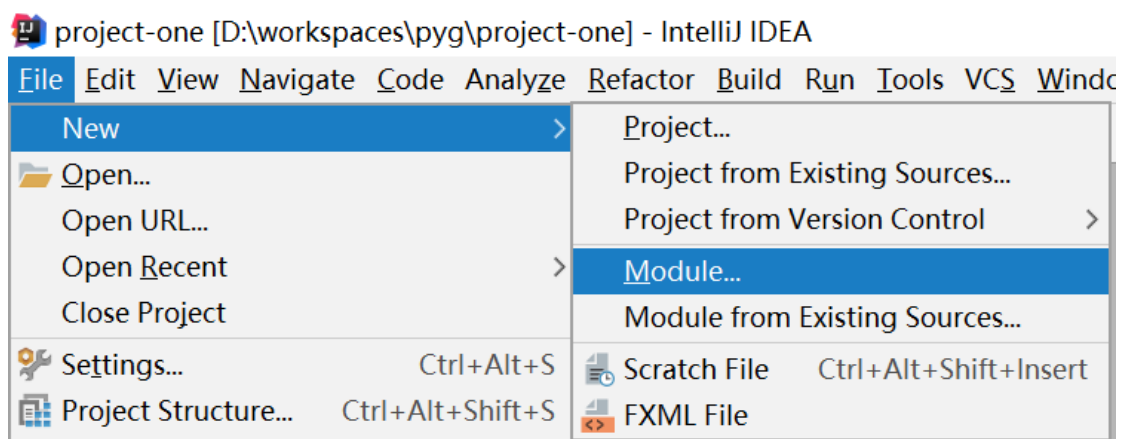
另外；也可以自行创建数据库，将“资料/database/pinyougodb.sql”导入。

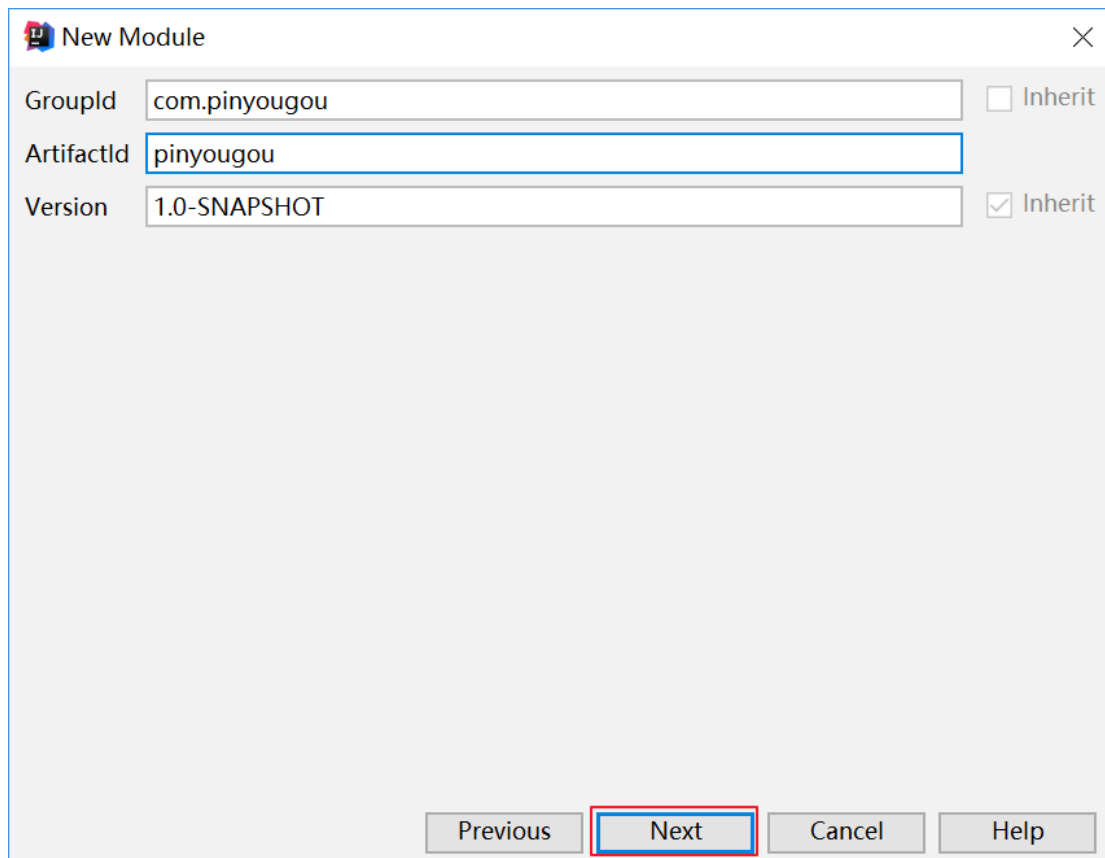
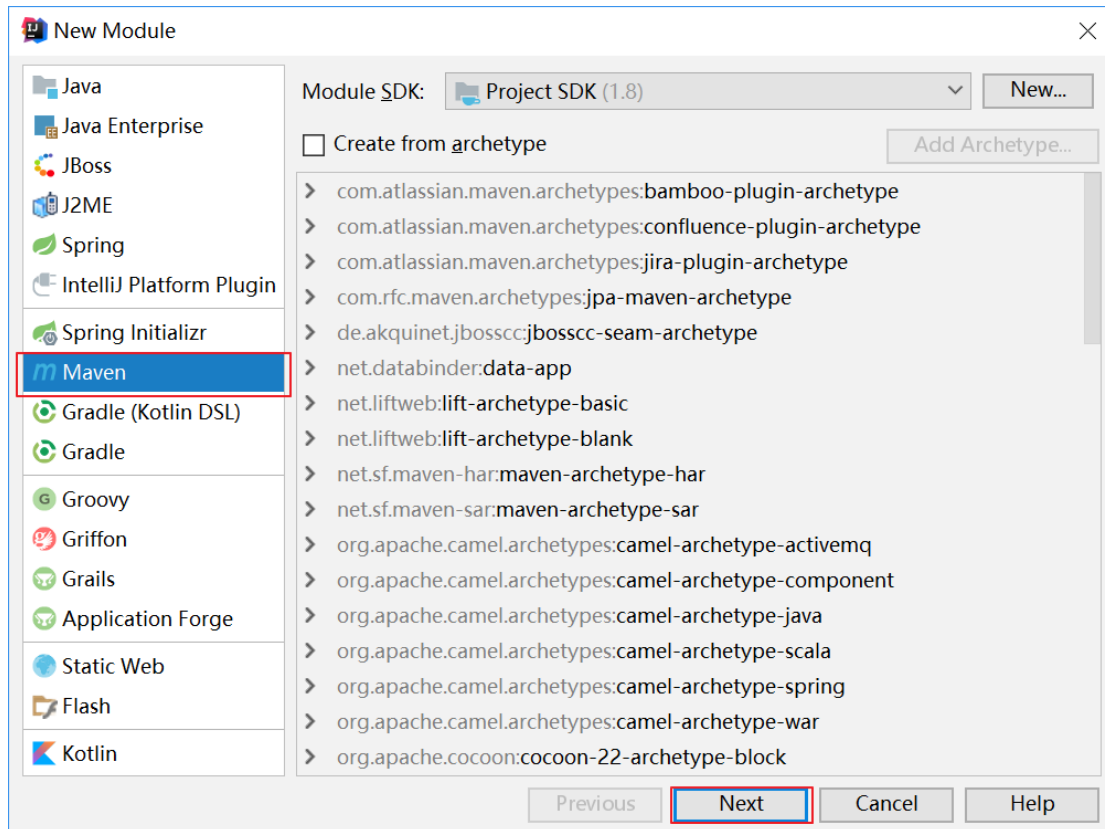
## 6.3. 创建工程&添加依赖

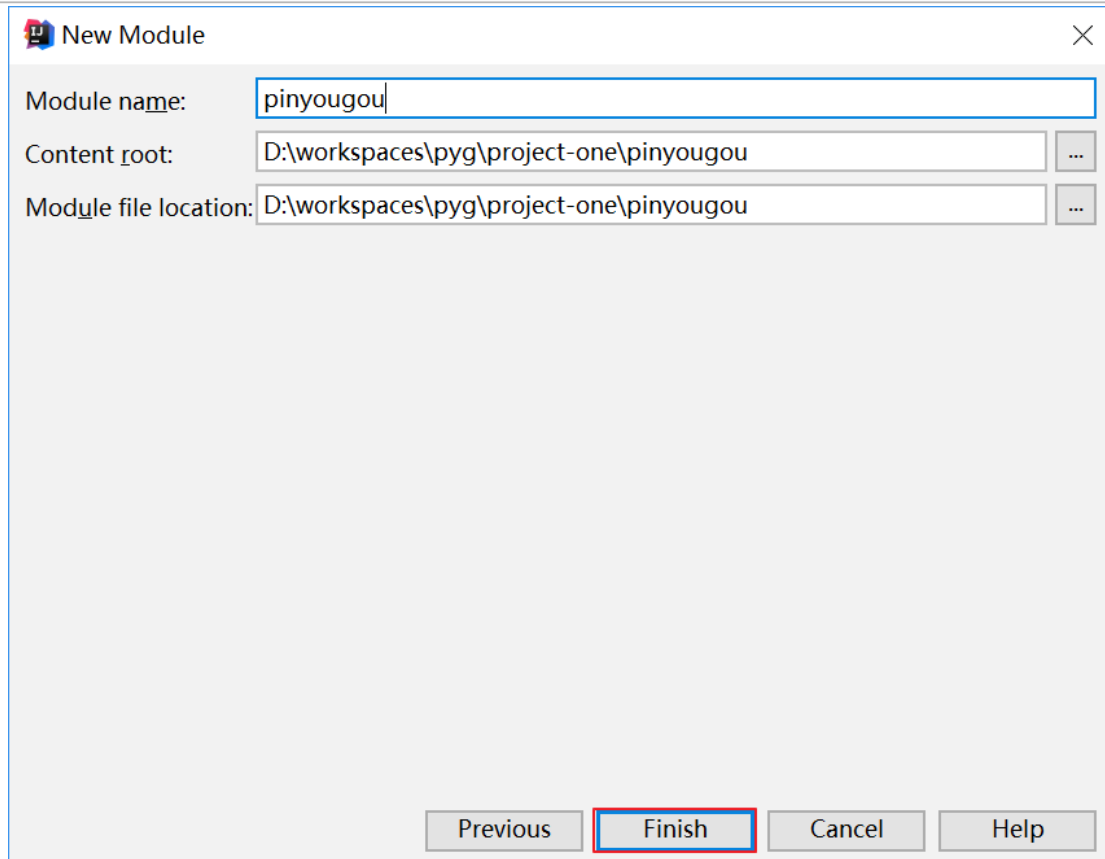
### 6.3.1. pinyougou

品优购聚合父工程；包括后续开发的所有品优购聚合子工程。主要功能是进行依赖、插件统一管理。

#### 6.3.1.1. 创建







### 6.3.1.2. 依赖

后续的文中出现依赖表示修改当前项目的 pom.xml 文件

```
<?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.pinyougou</groupId>
    <artifactId>pinyougou</artifactId>
    <packaging>pom</packaging>
    <version>1.0-SNAPSHOT</version>

    <!-- 集中定义依赖版本号 -->
    <properties>
        <junit.version>4.12</junit.version>
        <spring.version>4.3.10.RELEASE</spring.version>
        <spring.security.version>4.2.3.RELEASE</spring.security.version>
```





```
<mybatis.version>3.4.5</mybatis.version>
<mybatis.spring.version>1.3.1</mybatis.spring.version>
<mysql.version>5.1.32</mysql.version>
<druid.version>1.1.5</druid.version>
<pagehelper.version>5.1.2</pagehelper.version>
<mapper.version>3.5.0</mapper.version>
<dubbo.version>2.6.0</dubbo.version>
<slf4j.version>1.7.12</slf4j.version>
<javax.persistence.version>1.0</javax.persistence.version>
<fastjson.version>1.2.44</fastjson.version>
<zookeeper.version>3.4.9</zookeeper.version>
<zkclient.version>0.1</zkclient.version>
<jedis.version>2.9.0</jedis.version>
<spring.data.redis.version>1.8.9.RELEASE</spring.data.redis.version>
<servlet.api.version>2.5</servlet.api.version>
<commons.fileupload.version>1.3.1</commons.fileupload.version>
<fastdfs.client.version>1.25</fastdfs.client.version>
<freemarker.version>2.3.23</freemarker.version>
<activemq.version>5.14.3</activemq.version>
<cas.client.core.version>3.4.1</cas.client.core.version>
<spring.data.solr.version>1.5.5.RELEASE</spring.data.solr.version>
<httpclient.version>4.5.3</httpclient.version>
<wxpay.sdk.version>0.0.3</wxpay.sdk.version>
<httpcore.version>4.4.8</httpcore.version>
<javassist.version>3.12.1.GA</javassist.version>
<commons.codec.version>1.11</commons.codec.version>
<dom4j.version>1.6.1</dom4j.version>
<xml.apis.version>2.0.2</xml.apis.version>
<kaptcha.version>2.3.2</kaptcha.version>
</properties>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>${junit.version}</version>
    </dependency>

    <!-- Spring -->
    <dependency>
      <groupId>org.springframework</groupId>
```



```
<artifactId>spring-beans</artifactId>
<version>${spring.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>${spring.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</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-jdbc</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-jms</artifactId>
<version>${spring.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context-support</artifactId>
<version>${spring.version}</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-test</artifactId>
<version>${spring.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>

<!-- 日志 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${slf4j.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>

<!-- json 的解析 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
```



```
<version>${fastjson.version}</version>
</dependency>

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>${servlet.api.version}</version>
  <scope>provided</scope>
</dependency>

<!-- mybaitis 分页助手 -->
<dependency>
  <groupId>com.github.pagehelper</groupId>
  <artifactId>pagehelper</artifactId>
  <version>${pagehelper.version}</version>
</dependency>

<!-- 通用 Mapper -->
<dependency>
  <groupId>tk.mybatis</groupId>
  <artifactId>mapper</artifactId>
  <version>${mapper.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>

<!-- jpa 注解 -->
<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>${javax.persistence.version}</version>
```



```
</dependency>

<!-- redis 缓存 -->
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>${jedis.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-redis</artifactId>
    <version>${spring.data.redis.version}</version>
</dependency>

<!-- 文件上传组件 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>${commons.fileupload.version}</version>
</dependency>

<!--fastDFS 客户端工具类包-->
<dependency>
    <groupId>fastdfs_client</groupId>
    <artifactId>fastdfs_client</artifactId>
    <version>${fastdfs.client.version}</version>
</dependency>

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

<!-- ActiveMQ -->
<dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-client</artifactId>
    <version>${activemq.version}</version>
</dependency>
```



```
<!-- 身份验证 -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>${spring.security.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>${spring.security.version}</version>
</dependency>

<!-- cas -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-cas</artifactId>
    <version>${spring.security.version}</version>
</dependency>
<dependency>
    <groupId>org.jasig.cas.client</groupId>
    <artifactId>cas-client-core</artifactId>
    <version>${cas.client.core.version}</version>
    <!-- 排除 log4j 包冲突 -->
    <exclusions>
        <exclusion>
            <groupId>org.slf4j</groupId>
            <artifactId>log4j-over-slf4j</artifactId>
        </exclusion>
    </exclusions>
</dependency>

<!-- solr 相关 -->
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-solr</artifactId>
    <version>${spring.data.solr.version}</version>
</dependency>

<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>${httpclient.version}</version>
```



```
</dependency>

<!-- 微信支付 sdk -->
<dependency>
    <groupId>com.github.wxpay</groupId>
    <artifactId>wxpay-sdk</artifactId>
    <version>${wxpay.sdk.version}</version>
</dependency>

<!-- 验证码工具类包-->
<dependency>
    <groupId>com.github.penggle</groupId>
    <artifactId>kaptcha</artifactId>
    <version>${kaptcha.version}</version>
    <exclusions>
        <exclusion>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
        </exclusion>
    </exclusions>
</dependency>

<!-- 加密解密 -->
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>${commons.codec.version}</version>
</dependency>

<dependency>
    <groupId>javassist</groupId>
    <artifactId>javassist</artifactId>
    <version>${javassist.version}</version>
</dependency>

<dependency>
    <groupId>dom4j</groupId>
    <artifactId>dom4j</artifactId>
    <version>${dom4j.version}</version>
</dependency>

<dependency>
```





```
<groupId>xml-apis</groupId>
<artifactId>xml-apis</artifactId>
<version>${xml.apis.version}</version>
</dependency>
</dependencies>
</dependencyManagement>

<build>
  <plugins>
    <!-- java 编译插件 -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.6.0</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>UTF-8</encoding>
      </configuration>
    </plugin>
  </plugins>

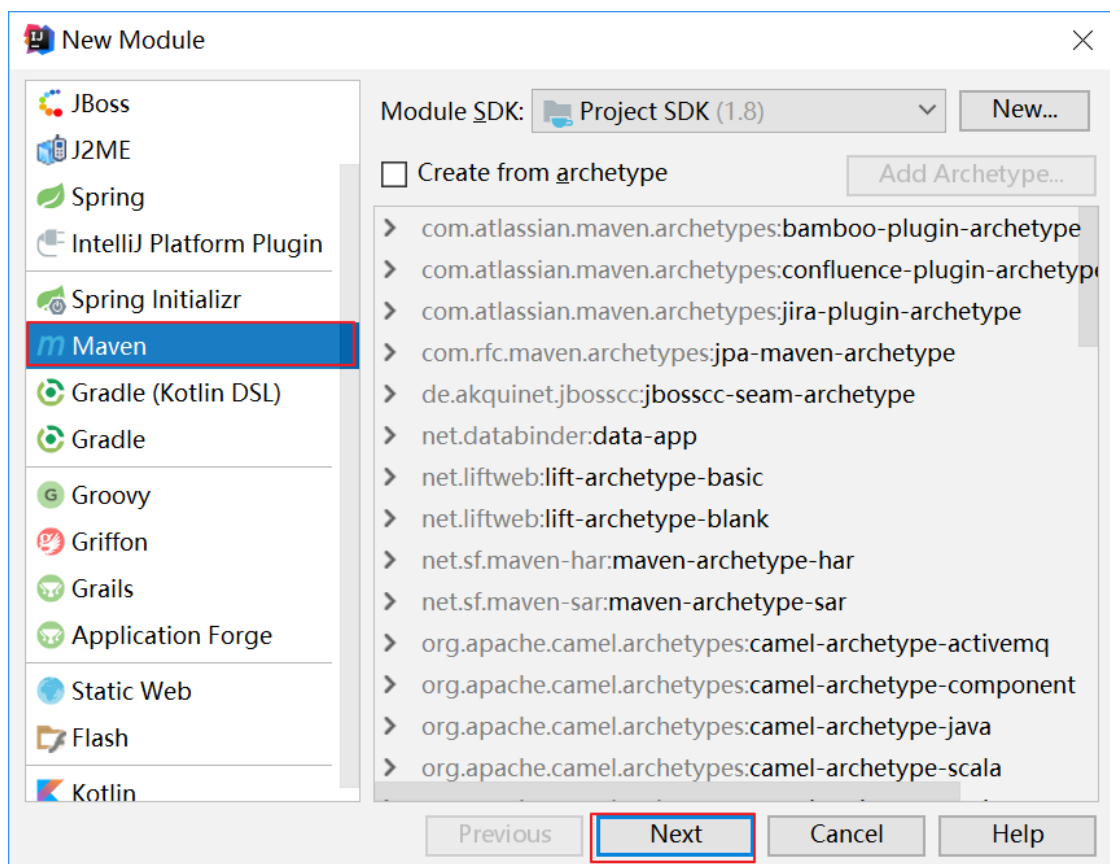
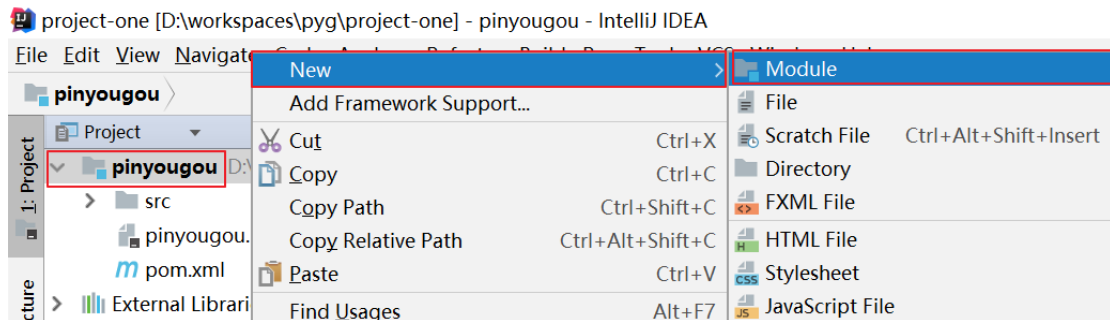
  <pluginManagement>
    <plugins>
      <!-- tomcat 插件 -->
      <plugin>
        <groupId>org.apache.tomcat.maven</groupId>
        <artifactId>tomcat7-maven-plugin</artifactId>
        <version>2.2</version>
      </plugin>
    </plugins>
  </pluginManagement>
</build>
</project>
```

### 6.3.2. pinyougou-common

通用工具类工程：放置其它工程经常使用的工具类或者组件。



### 6.3.2.1. 创建





**New Module** ×

表示当前模块输入那个模块的子模块；可以通过右边 ... 按钮 修改

Add as module to com.pinyougou:pinyougou:1.0-SNAPSHOT ...

Parent com.pinyougou:pinyougou:1.0-SNAPSHOT ... 父工程，右边按钮可修改

GroupId com.pinyougou ☒ Inherit

ArtifactId pinyougou-common ① ☒ Inherit

Version 1.0-SNAPSHOT ☒ Inherit

是否要继承父工程的部分坐标  
如果要修改则去掉勾即可修改

Previous Next ② Cancel Help

**New Module** ×

Module name: pinyougou-common ①

Content root: D:\workspaces\pyg\project-one\pinyougou\pinyougou-common ② ...

Module file location: D:\workspaces\pyg\project-one\pinyougou\pinyougou-common ③ ...

注意：  
**Module name** 默认会把 - 去掉，所以请自行重写

小窍门：直接修改**Content root** 的值其它的会跟着变化；如：写了  
**pinyougou-common**后，**Module name** 和 **Module file location**会跟着变为  
**pinyougou-common**

Previous Finish ④ Cancel Help



### 6.3.2.2. 依赖

```
<?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">
  <parent>
    <artifactId>pinyougou</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>pinyougou-common</artifactId>

</project>
```

### 6.3.3. pinyougou-pojo

实体类工程；统一放置品优购的所有数据库实体类或者其它领域模型。

#### 6.3.3.1. 创建

创建 pinyougou 的子模块工程，与 pinyougou-common 的创建方式类似；后面的若是 pinyougou 的子模块工程的创建也都如此；不再每一步都截图了。



**New Module** [X]

Add as module to com.pinyougou:pinyougou:1.0-SNAPSHOT [...]

Parent com.pinyougou:pinyougou:1.0-SNAPSHOT [...]

GroupId com.pinyougou ☒ Inherit

ArtifactId pinyougou-pojo

Version 1.0-SNAPSHOT ☒ Inherit

Previous Next Cancel Help

**New Module** [X]

Module name: pinyougou-pojo

Content root: D:\workspaces\pyg\project-one\pinyougou\pinyougou-pojo [...]

Module file location: D:\workspaces\pyg\project-one\pinyougou\pinyougou-pojo [...]

Previous Finish Cancel Help



### 6.3.3.2. 依赖

```
<?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">
  <parent>
    <artifactId>pinyougou</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>pinyougou-pojo</artifactId>

  <dependencies>
    <!-- jpa 注解 -->
    <dependency>
      <groupId>javax.persistence</groupId>
      <artifactId>persistence-api</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.data</groupId>
      <artifactId>spring-data-solr</artifactId>
    </dependency>
  </dependencies>
</project>
```

### 6.3.4. pinyougou-dao

数据库通用访问层工程；放置 Mybatis 中对应于数据库操作的所有 Mapper 接口类。



### 6.3.4.1. 创建

The 'New Module' dialog box shows the configuration for a new module. The 'Add as module to' field is set to 'com.pinyougou:pinyougou:1.0-SNAPSHOT'. The 'Parent' field is also set to 'com.pinyougou:pinyougou:1.0-SNAPSHOT'. The 'GroupId' field is 'com.pinyougou'. The 'ArtifactId' field is 'pinyougou-dao'. The 'Version' field is '1.0-SNAPSHOT'. There are checkboxes for 'Inherit' next to the 'ArtifactId' and 'Version' fields, both of which are checked. At the bottom, there are buttons for 'Previous', 'Next', 'Cancel', and 'Help'.

New Module

Add as module to com.pinyougou:pinyougou:1.0-SNAPSHOT

Parent com.pinyougou:pinyougou:1.0-SNAPSHOT

GroupId com.pinyougou

ArtifactId pinyougou-dao

Version 1.0-SNAPSHOT

Previous Next Cancel Help

The 'New Module' dialog box shows the configuration for a new module. The 'Module name' field is 'pinyougou-dao'. The 'Content root' field is 'D:\workspaces\pyg\project-one\pinyougou\pinyougou-dao'. The 'Module file location' field is 'D:\workspaces\pyg\project-one\pinyougou\pinyougou-dao'. At the bottom, there are buttons for 'Previous', 'Finish', 'Cancel', and 'Help'.

New Module

Module name: pinyougou-dao

Content root: D:\workspaces\pyg\project-one\pinyougou\pinyougou-dao

Module file location: D:\workspaces\pyg\project-one\pinyougou\pinyougou-dao

Previous Finish Cancel Help





### 6.3.4.2. 依赖

```
<?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">
  <parent>
    <artifactId>pinyougou</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>pinyougou-dao</artifactId>

  <dependencies>
    <dependency>
      <groupId>com.pinyougou</groupId>
      <artifactId>pinyougou-pojo</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-beans</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-jdbc</artifactId>
    </dependency>

    <!-- Mybatis -->
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
    </dependency>
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis-spring</artifactId>
```



```

</dependency>

<!-- mybaitis 分页助手 -->
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
</dependency>

<!-- 通用 Mapper -->
<dependency>
    <groupId>tk.mybatis</groupId>
    <artifactId>mapper</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>

<!-- 日志 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
</dependency>
</dependencies>

</project>

```

### 6.3.5. pinyougou-sellergoods

商家商品服务工程；该工程主要是对应于商品、商家、品牌、规格、商品分类等的服务提供，由 pinyougou-sellergoods-interface 和 pinyougou-sellergoods-service



构成。

### 6.3.5.1. 创建

New Module

Add as module to com.pinyougou:pinyougou:1.0-SNAPSHOT

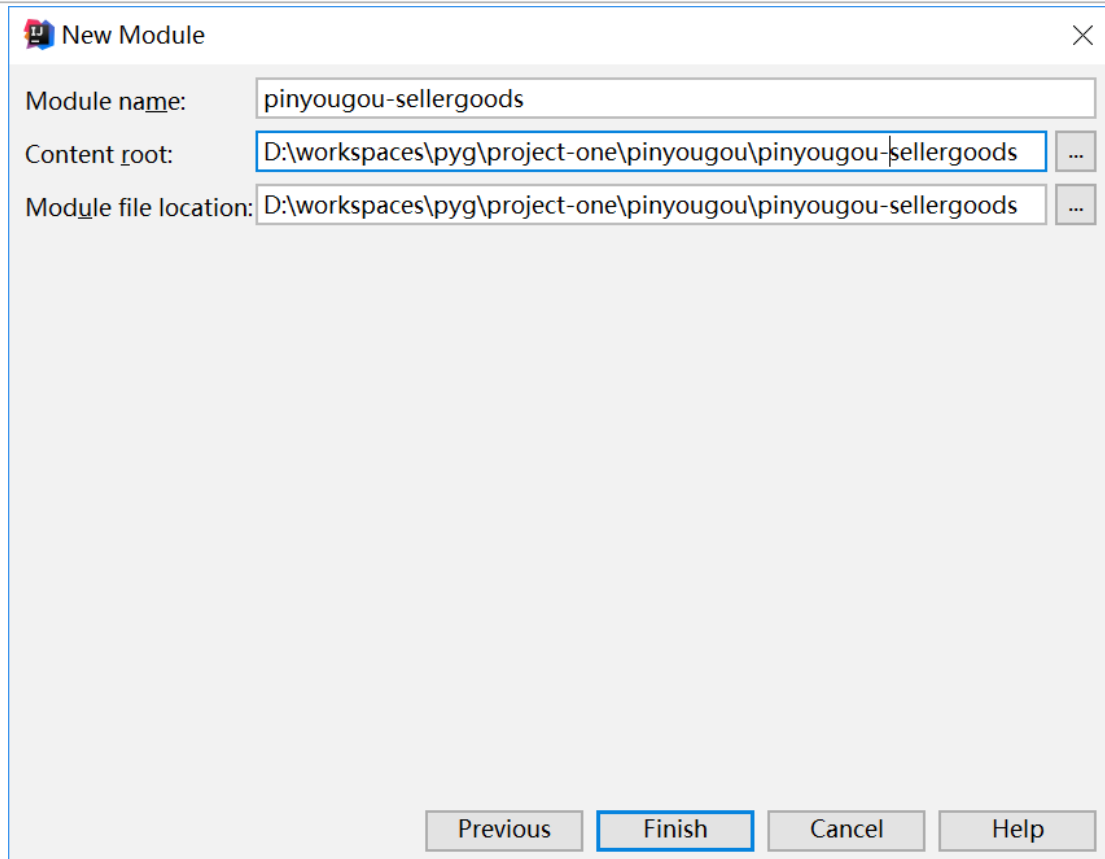
Parent com.pinyougou:pinyougou:1.0-SNAPSHOT

GroupId com.pinyougou ☒ Inherit

ArtifactId pinyougou-sellergoods

Version 1.0-SNAPSHOT ☒ Inherit

Previous Next Cancel Help



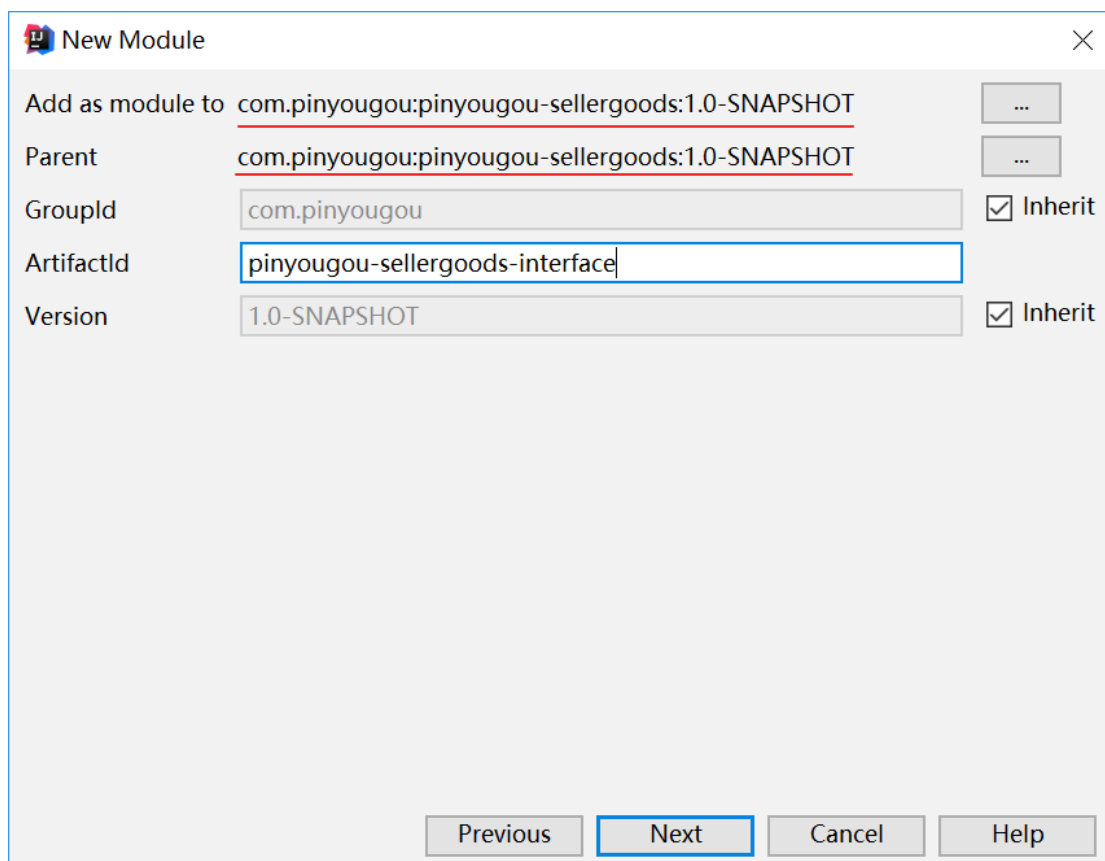
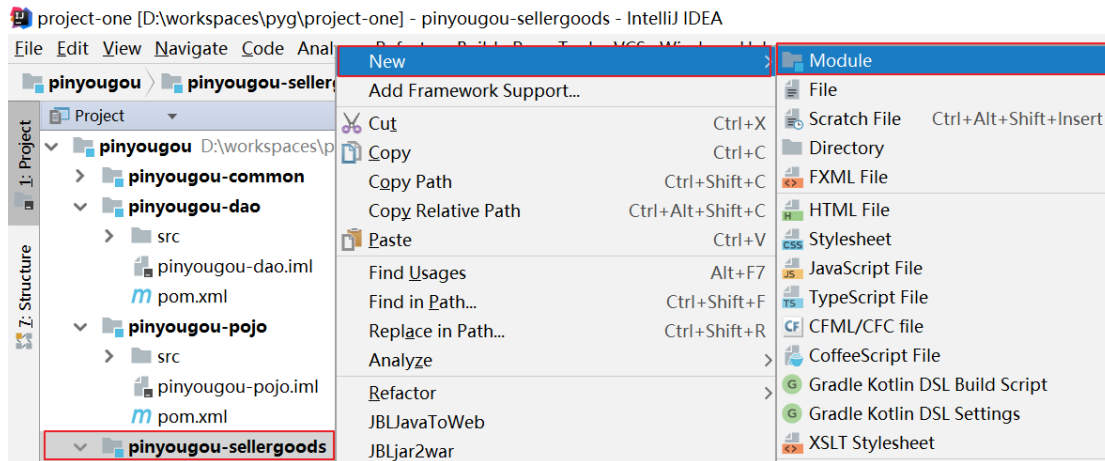
#### 6.3.5.2. 依赖

无需改动。

### 6.3.6. pinyougou-sellergoods-interface

#### 6.3.6.1. 创建

注意这个工程是 pinyougou-sellergoods 的聚合子工程。





**New Module** [X]

Module name:

Content root:  ...

Module file location:  ...

Previous Finish Cancel Help

### 6.3.6.2. 依赖

```
<?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">
  <parent>
    <artifactId>pinyougou-sellergoods</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>pinyougou-sellergoods-interface</artifactId>

  <dependencies>
    <dependency>
      <groupId>com.pinyougou</groupId>
      <artifactId>pinyougou-pojo</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>
  </dependencies>
</project>
```



```
</dependency>  
</dependencies>  
</project>
```

## 6.3.7. pinyougou-sellergoods-service

### 6.3.7.1. 创建

注意这个工程是 pinyougou-sellergoods 的聚合子工程。

New Module

Add as module to com.pinyougou:pinyougou-sellergoods:1.0-SNAPSHOT ...

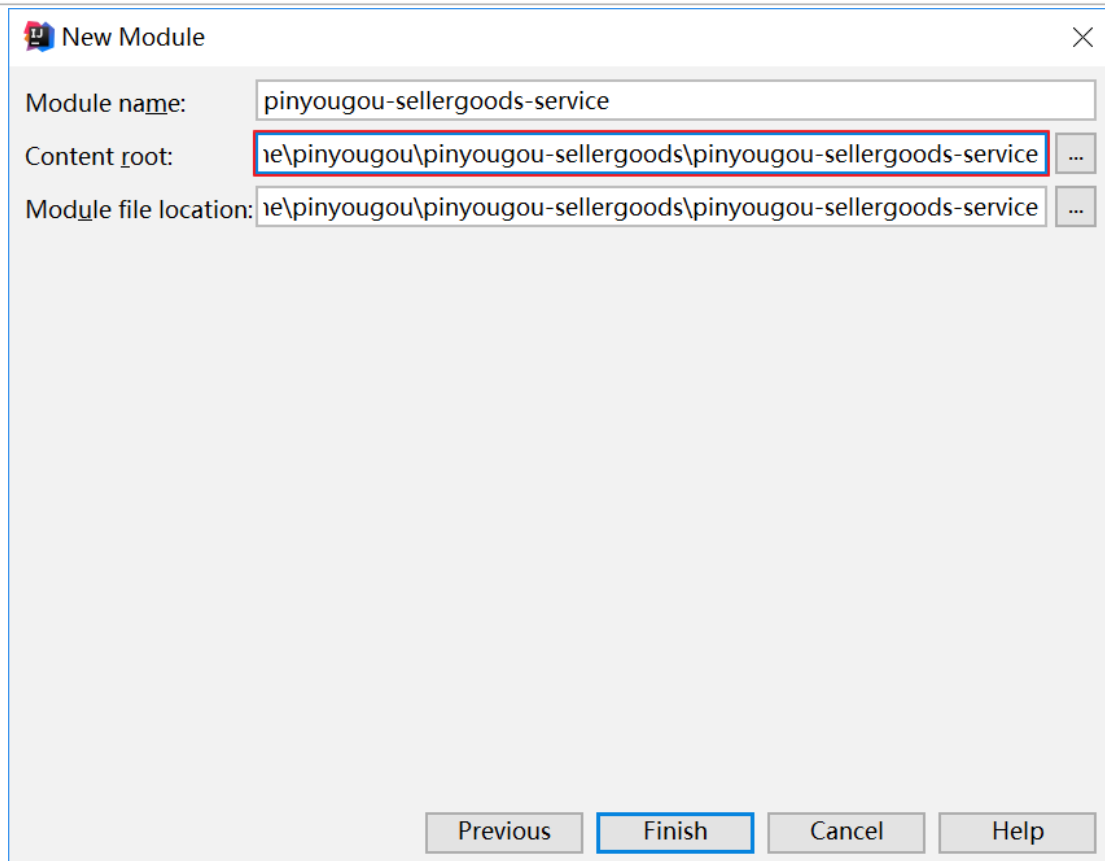
Parent com.pinyougou:pinyougou-sellergoods:1.0-SNAPSHOT ...

GroupId com.pinyougou ☒ Inherit

ArtifactId pinyougou-sellergoods-service ☒ Inherit

Version 1.0-SNAPSHOT ☒ Inherit

Previous Next Cancel Help

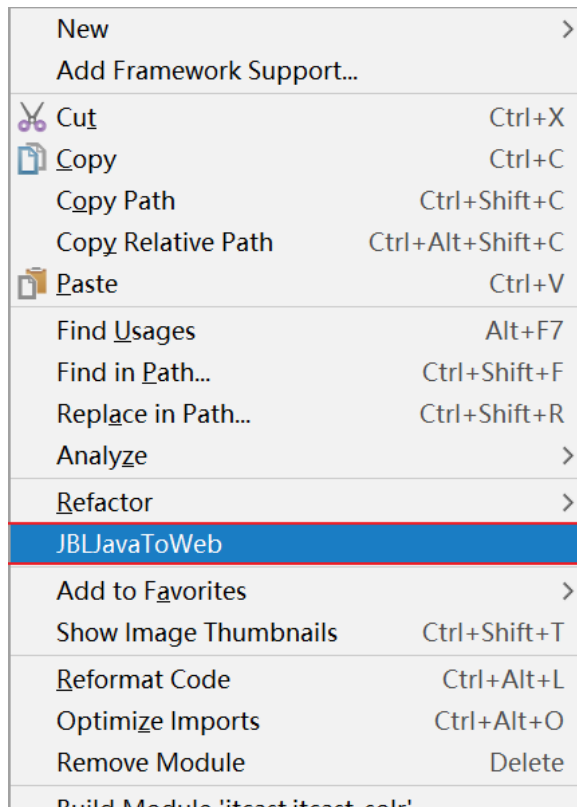


#### 6.3.7.2. 转换为 web 工程

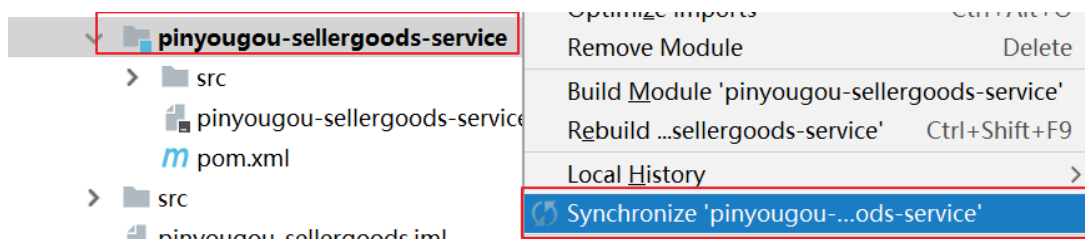
查看“资料\idea\IntelliJ IDEA 常用操作说明.pdf”文档的插件安装章节；安装 JBLJavaToWeb（**推荐。也可以转非 Maven 工程**）。进行如下转换操作：

右击要转换的模块工程，在出现的右击菜单中点击如下：

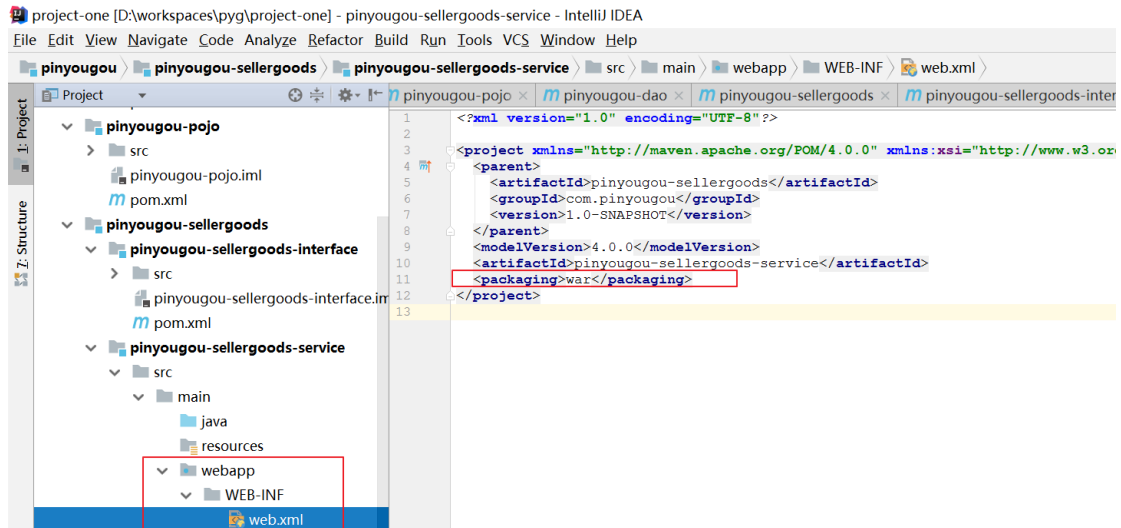




转换完后可以如下更新工程（一般设置过自动检测 maven 的话是不需要如下操作的）



转换后效果如下：



### 6.3.7.3. 依赖

```
<?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">
  <parent>
    <artifactId>pinyougou-sellergoods</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>pinyougou-sellergoods-service</artifactId>
  <packaging>war</packaging>

  <dependencies>

    <dependency>
      <groupId>com.pinyougou</groupId>
      <artifactId>pinyougou-sellergoods-interface</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>

    <dependency>
      <groupId>com.pinyougou</groupId>
      <artifactId>pinyougou-dao</artifactId>
      <version>1.0-SNAPSHOT</version>
```



```
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-beans</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
</dependency>

<!-- dubbo 相关 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>dubbo</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.zookeeper</groupId>
    <artifactId>zookeeper</artifactId>
</dependency>
<dependency>
    <groupId>com.github.sgroschupf</groupId>
    <artifactId>zkclient</artifactId>
</dependency>

<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
</dependency>
<dependency>
    <groupId>com.pinyougou</groupId>
    <artifactId>pinyougou-common</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>
```



```
</dependencies>

<build>
  <plugins>
    <!-- tomcat 插件 -->
    <plugin>
      <groupId>org.apache.tomcat.maven</groupId>
      <artifactId>tomcat7-maven-plugin</artifactId>
      <configuration>
        <path>/</path>
        <port>9000</port>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```



## 6.3.8. pinyougou-manage-web

### 6.3.8.1. 创建

New Module

Add as module to com.pinyougou:pinyougou:1.0-SNAPSHOT

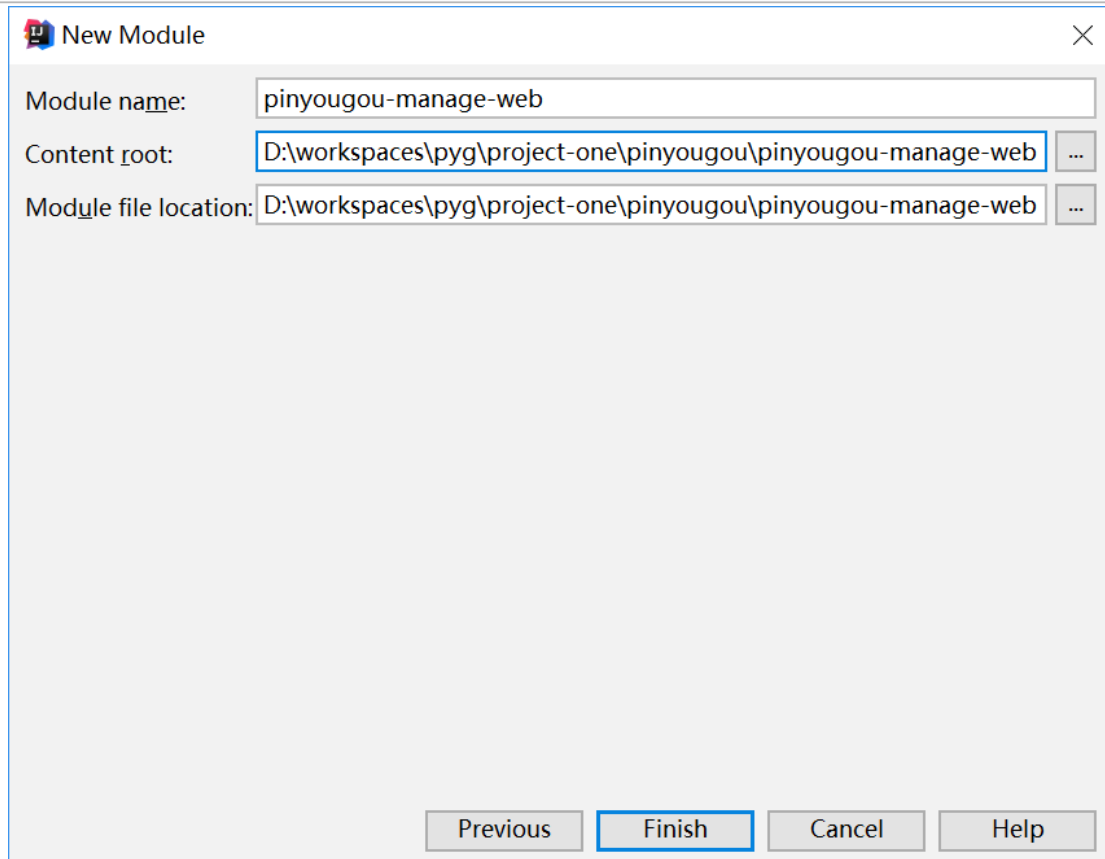
Parent com.pinyougou:pinyougou:1.0-SNAPSHOT

GroupId com.pinyougou ☒ Inherit

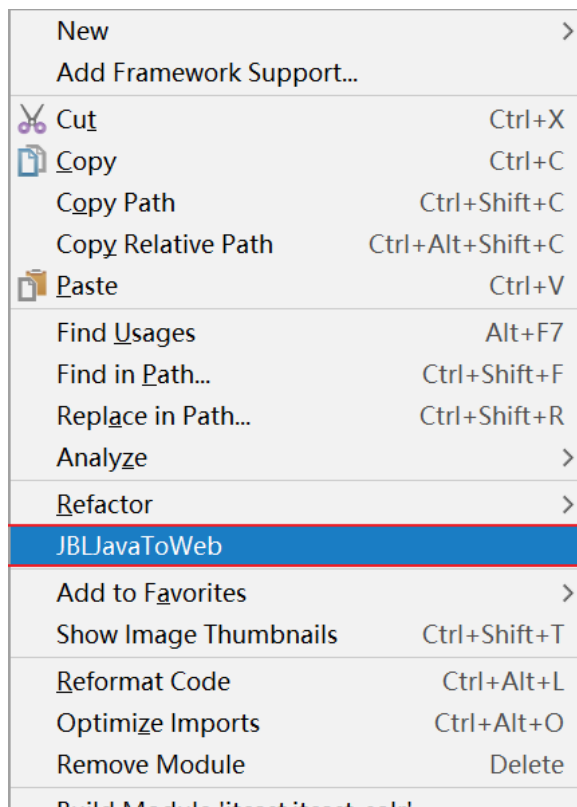
ArtifactId pinyougou-manage-web

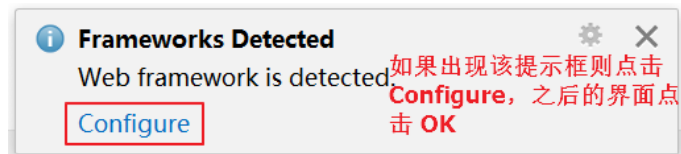
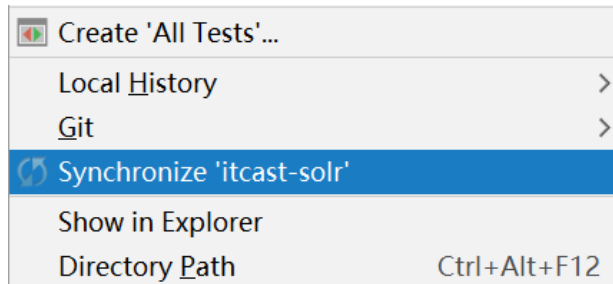
Version 1.0-SNAPSHOT ☒ Inherit

Previous Next Cancel Help



### 6.3.8.2. 转换为 web 工程





### 6.3.8.3. 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">
  <parent>
    <artifactId>pinyougou</artifactId>
    <groupId>com.pinyougou</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <artifactId>pinyougou-manage-web</artifactId>
  <packaging>war</packaging>
  <dependencies>
    <dependency>
      <groupId>com.pinyougou</groupId>
      <artifactId>pinyougou-sellergoods-interface</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
    </dependency>

    <dependency>
      <groupId>org.slf4j</groupId>
```



```
<artifactId>slf4j-log4j12</artifactId>
</dependency>

<!-- dubbo 相关 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>dubbo</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.zookeeper</groupId>
    <artifactId>zookeeper</artifactId>
</dependency>
<dependency>
    <groupId>com.github.sgroschupf</groupId>
    <artifactId>zkclient</artifactId>
</dependency>

<!-- json 的解析 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
</dependency>

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <scope>provided</scope>
</dependency>
</dependencies>
<build>
    <plugins>
        <!-- tomcat 插件 -->
        <plugin>
            <groupId>org.apache.tomcat.maven</groupId>
            <artifactId>tomcat7-maven-plugin</artifactId>
            <configuration>
                <path></path>
                <port>9100</port>
            </configuration>
        </plugin>
    </plugins>
```





```
</build>  
</project>
```

## 6.4. 整合工程

### 6.4.1. pinyougou-dao

#### 6.4.1.1. applicationContext-dao.xml

添加

pinyougou\pinyougou-dao\src\main\resources\spring\applicationContext-dao.xml

内容如下：

```
<?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:context="http://www.springframework.org/schema/context"  
    xsi:schemaLocation="http://www.springframework.org/schema/beans  
        http://www.springframework.org/schema/beans/spring-beans.xsd  
        http://www.springframework.org/schema/context  
        http://www.springframework.org/schema/context/spring-context.xsd">  
  
    <!-- 加载配置文件 -->  
    <context:property-placeholder  
        location="classpath*:properties/*.properties"/>  
  
    <!-- 数据库连接池 -->  
    <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"  
        destroy-method="close">  
        <property name="url" value="${jdbc.url}"/>  
        <property name="username" value="${jdbc.username}"/>  
        <property name="password" value="${jdbc.password}"/>  
        <property name="driverClassName" value="${jdbc.driver}"/>  
        <property name="maxActive" value="10"/>  
        <property name="minIdle" value="5"/>  
    </bean>  
  
    <!-- 配置 sqlSessionFactory -->  
    <bean id="sqlSessionFactory"  
        class="org.mybatis.spring.SqlSessionFactoryBean">
```



```
<!-- 数据源 -->
<property name="dataSource" ref="dataSource"/>
<!-- mybatis 总配置文件 -->
<property name="configLocation" value="classpath:mybatis/mybatis-
config.xml"/>
<!-- 配置实体类别名 -->
<property name="typeAliasesPackage" value="com.pinyougou.pojo"/>
</bean>

<!-- 扫描 mapper -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="com.pinyougou.mapper"/>
</bean>

</beans>
```

#### 6.4.1.2. mybatis-config.xml

添加

pinyougou\pinyougou-dao\src\main\resources\mybatis\mybatis-config.xml

内容如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>

  <settings>
    <!-- 开启驼峰规则自动映射字段属性值；如字段为 user_name 的可以映射到 userName
    属性中 -->
    <setting name="mapUnderscoreToCamelCase" value="true"/>
  </settings>

</configuration>
```

#### 6.4.1.3. db.properties

添加



pinyougou\pinyougou-dao\src\main\resources\properties\db.properties

内容如下：

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://192.168.12.168:3306/pinyougoudb?useUnicode=true&characterEncoding=utf8&autoReconnect=true
jdbc.username=root
jdbc.password=root
```

#### 6.4.1.4.log4j.properties

添加

pinyougou\pinyougou-dao\src\main\resources\log4j.properties

内容如下：

```
log4j.rootLogger=DEBUG,stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss,SSS} [%t] [%c]-[%p] %m%n
```

### 6.4.2. pinyougou-sellergoods-service

#### 6.4.2.1.applicationContext-service.xml

添加

pinyougou-sellergoods-service\src\main\resources\spring\applicationContext-service.xml

内容如下：

```
<?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:dubbo="http://code.alibabatech.com/schema/dubbo"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://code.alibabatech.com/schema/dubbo
http://code.alibabatech.com/schema/dubbo/dubbo.xsd">
```



```
<dubbo:protocol name="dubbo" port="20880"/>
<dubbo:application name="pinyougou-sellergoods-service"/>
<dubbo:registry address="zookeeper://192.168.12.168:2181"/>
<dubbo:annotation package="com.pinyougou.sellergoods.service.impl"/>

</beans>
```

### 6.4.2.2. web.xml

修改 pinyougou-sellergoods-service\src\main\webapp\WEB-INF\web.xml 内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-
class>
    </listener>
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath*:spring/applicationContext*.xml</param-value>
    </context-param>
</web-app>
```

### 6.4.3. pinyougou-manage-web

#### 6.4.3.1. springmvc.xml

添加

pinyougou\pinyougou-manage-web\src\main\resources\spring\springmvc.xml

内容如下：

```
<?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:dubbo="http://code.alibabatech.com/schema/dubbo"
```



```
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://code.alibabatech.com/schema/dubbo
http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

<mvc:annotation-driven>
    <mvc:message-converters register-defaults="true">
        <bean
class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter">
            <property name="supportedMediaTypes"
value="application/json"/>
        </bean>
    </mvc:message-converters>
</mvc:annotation-driven>

<!-- 引用 dubbo 服务 -->
<dubbo:application name="pinyougou-manage-web" />
<dubbo:registry address="zookeeper://192.168.12.168:2181"/>
<dubbo:annotation package="com.pinyougou.manage.controller" />
</beans>
```

### 6.4.3.2.log4j.properties

添加

pinyougou\pinyougou-manage-web\src\main\resources\log4j.properties

内容如下：

```
log4j.rootLogger=DEBUG,stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%-d{yyyy-MM-dd
HH:mm:ss,SSS} [%t] [%c]-[%p] %m%n
```

### 6.4.3.3.web.xml

修改



pinyougou\pinyougou-manage-web\src\main\webapp\WEB-INF\web.xml

内容如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5">
    <!-- 解决 post 乱码 -->
    <filter>
        <filter-name>CharacterEncodingFilter</filter-name>
        <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-
class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>utf-8</param-value>
        </init-param>
        <init-param>
            <param-name>forceEncoding</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>CharacterEncodingFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

    <servlet>
        <servlet-name>springmvc</servlet-name>
        <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <!-- 指定加载的配置文件，通过参数 contextConfigLocation 加载-->
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:spring/springmvc.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>springmvc</servlet-name>
```



```
<url-pattern>*.do</url-pattern>
</servlet-mapping>

</web-app>
```

#### 6.4.4. 引入品优购实体类

在 pinyougou-pojo 工程创建 com.pinyougou.pojo 包；然后将“资料/pojo”下的所有实体类复制到上述创建的包中。

## 7. 查询品牌列表数据

### 7.1. 需求分析

在浏览器上访问：<http://localhost:9100/brand/findAll.do> 把品牌数据(json 格式字符串)加载并显示。

### 7.2. 数据库表

```
CREATE TABLE `tb_brand` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL COMMENT '品牌名称',
  `first_char` varchar(1) DEFAULT NULL COMMENT '品牌首字母',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8;
```

### 7.3. 后端代码

#### 7.3.1. BrandMapper 接口

添加

pinyougou-dao\src\main\java\com\pinyougou\mapper\BrandMapper.java

内容如下：



```
1 package com.pinyougou.mapper;
2
3 import com.pinyougou.pojo.TbBrand;
4
5 import java.util.List;
6
7 public interface BrandMapper {
8
9     public List<TbBrand> queryAll();
10
11 }
```

### 7.3.2. BrandMapper.xml 配置文件

添加

pinyougou-dao\src\main\resources\com\pinyougou\mapper\BrandMapper.xml

内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.pinyougou.mapper.BrandMapper">
    <select id="queryAll" resultType="TbBrand">
        SELECT * FROM tb_brand
    </select>
</mapper>
```

### 7.3.3. BrandService 接口

添加

pinyougou-sellergoods\pinyougou-sellergoods-

interface\src\main\java\com\pinyougou\sellergoods\service\BrandService.java

内容如下：

```
1 package com.pinyougou.sellergoods.service;
2
3 import com.pinyougou.pojo.TbBrand;
4
5 import java.util.List;
6
7 public interface BrandService {
8     public List<TbBrand> queryAll();
9 }
```





### 7.3.4. BrandServiceImpl 实现类

添加

pinyougou-sellergoods\pinyougou-sellergoods-

service\src\main\java\com\pinyougou\sellergoods\service\impl\BrandServiceImpl.java

va

内容如下：

```
1 package com.pinyougou.sellergoods.service.impl;
2
3 import com.alibaba.dubbo.config.annotation.Service;
4 import com.pinyougou.mapper.BrandMapper;
5 import com.pinyougou.pojo.TbBrand;
6 import com.pinyougou.sellergoods.service.BrandService;
7 import org.springframework.beans.factory.annotation.Autowired;
8
9 import java.util.List;    注意：该Service注解是阿里巴巴的，不是spring的
10
11 @Service(interfaceClass = BrandService.class)
12 public class BrandServiceImpl implements BrandService {
13
14     @Autowired
15     private BrandMapper brandMapper;
16
17     @Override
18     public List<TbBrand> queryAll() {
19         return brandMapper.queryAll();
20     }
21 }
```

### 7.3.5. BrandController 处理器

添加

pinyougou-manage-

web\src\main\java\com\pinyougou\manage\controller\BrandController.java

内容如下：



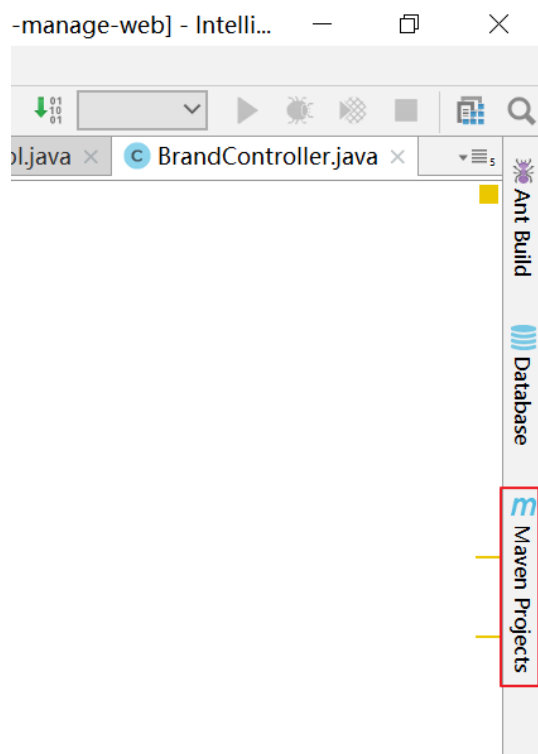
```
1 package com.pinyougou.manage.controller;
2
3 import com.alibaba.dubbo.config.annotation.Reference;
4 import com.pinyougou.pojo.TbBrand;
5 import com.pinyougou.sellergoods.service.BrandService;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import java.util.List;
11
12 @RequestMapping("/brand")
13 @RestController
14 public class BrandController {
15
16     @Reference
17     private BrandService brandService;
18
19     @GetMapping("/findAll")
20     public List<TbBrand> findAll() {
21         return brandService.queryAll();
22     }
23 }
```

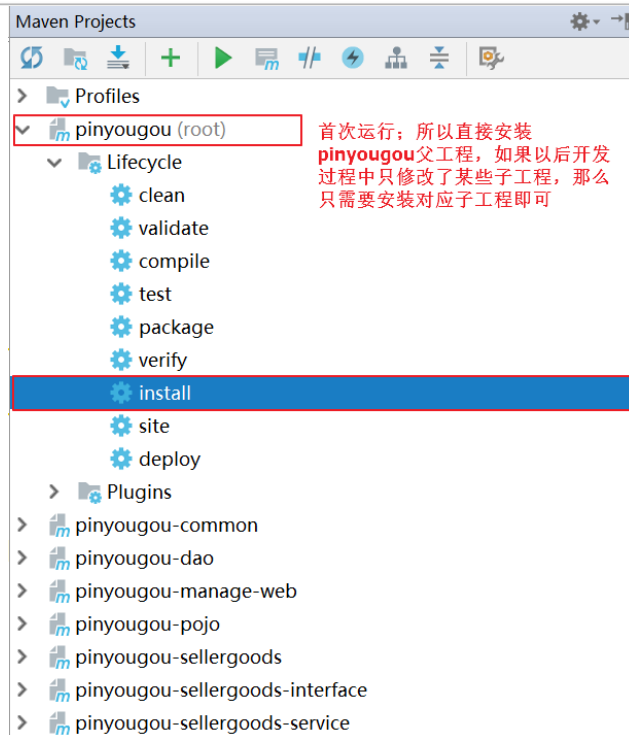
该Reference注解来自阿里巴巴；不是其它的，请注意不要引入错误注解。

还有要注意的是这个注入使用@Reference 不是@Autowired

## 7.4. 安装工程

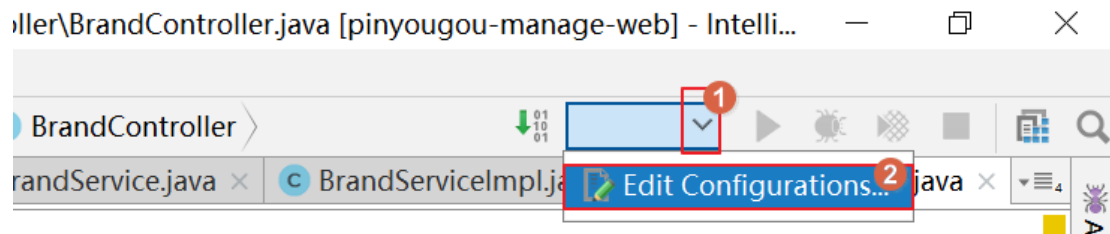
点击 IDEA 最右边的“Maven Projects”

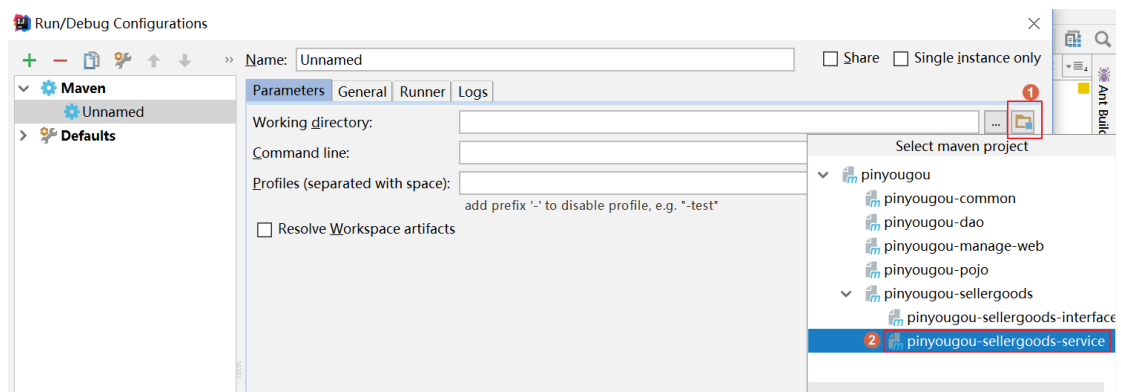
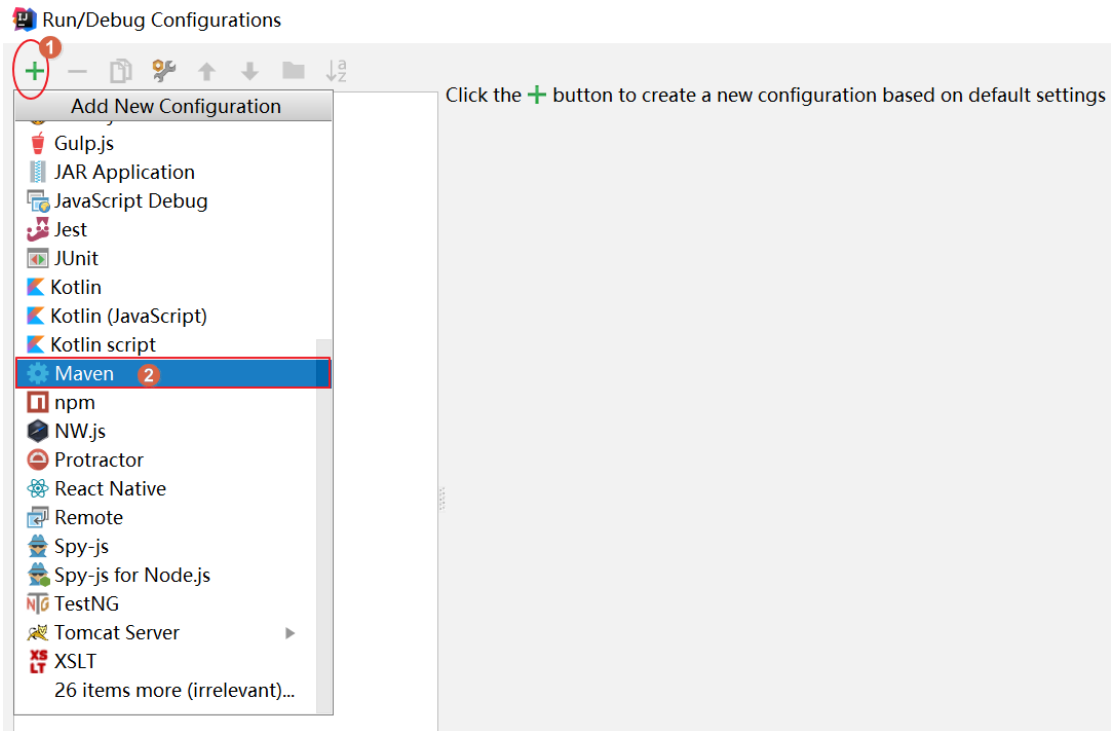


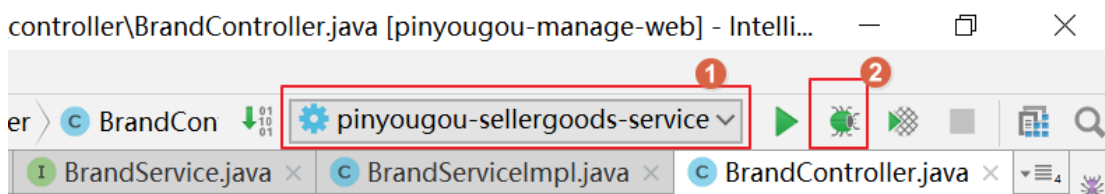
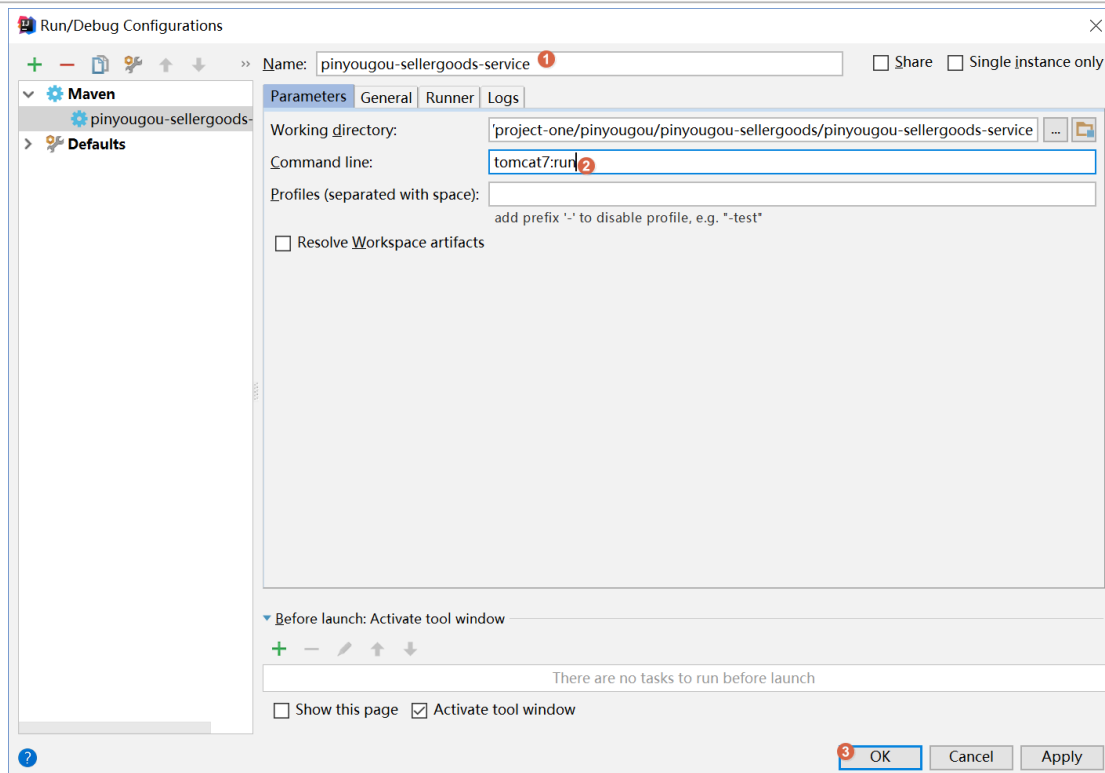


## 7.5. 启动

### 7.5.1. 启动 pinyougou-sellergoods-service

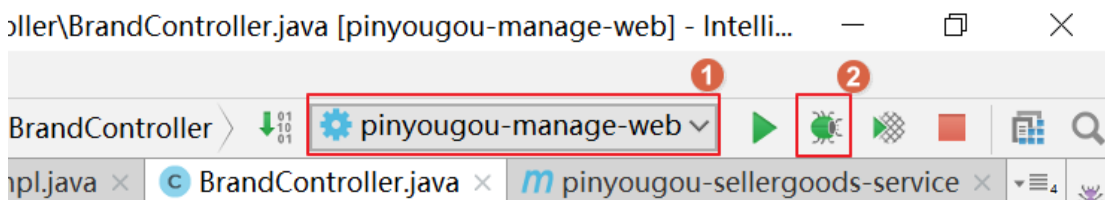






## 7.5.2. 启动 pinyougou-manage-web

这个启动与 pinyougou-sellergoods-service 的启动配置是一样的；不在赘述图。



## 7.6. 测试

打开 chrome 浏览器；输入：<http://localhost:9100/brand/findAll.do>

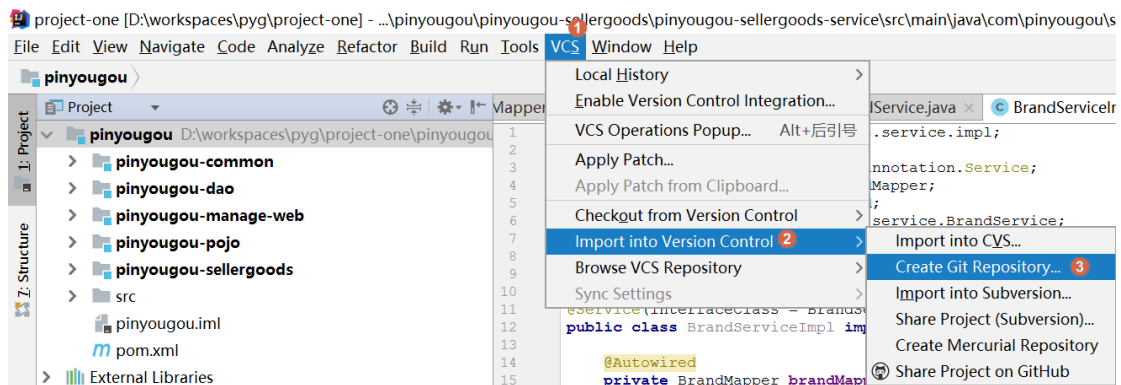


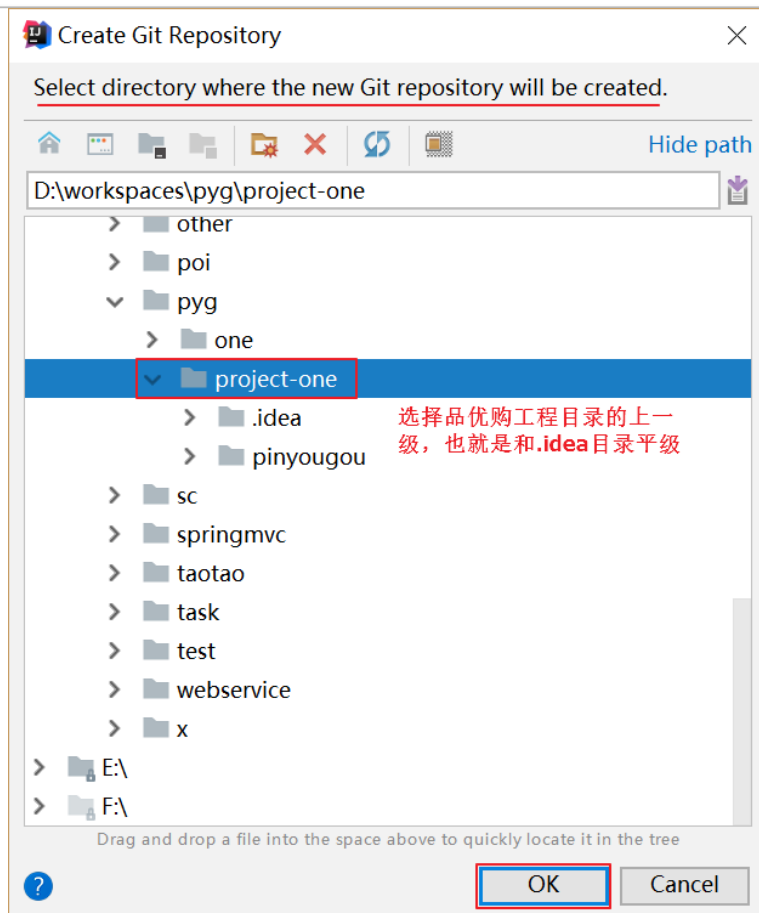
```
localhost:9100/brand/ x
localhost:9100/brand/findAll.do
应用 点击这里导入书签。 开始
[[{"firstChar": "L", "id": 1, "name": "联想"}, {"firstChar": "H", "id": 2, "name": "华为"}, {"firstChar": "S", "id": 3, "name": "三星"}, {"firstChar": "X", "id": 4, "name": "小米"}, {"firstChar": "O", "id": 5, "name": "OPPO"}, {"firstChar": "S", "id": 6, "name": "360"}, {"firstChar": "Z", "id": 7, "name": "中兴"}, {"firstChar": "M", "id": 8, "name": "魅族"}, {"firstChar": "V", "id": 10, "name": "VIVO"}, {"firstChar": "N", "id": 11, "name": "诺基亚"}, {"firstChar": "C", "id": 12, "name": "锤子"}, {"firstChar": "C", "id": 13, "name": "长虹"}, {"firstChar": "H", "id": 14, "name": "海尔"}, {"firstChar": "F", "id": 15, "name": "飞利浦"}, {"firstChar": "T", "id": 16, "name": "TCL"}, {"firstChar": "H", "id": 17, "name": "海信"}, {"firstChar": "X", "id": 18, "name": "夏普"}, {"firstChar": "C", "id": 19, "name": "创维"}, {"firstChar": "D", "id": 20, "name": "东芝"}, {"firstChar": "K", "id": 21, "name": "康佳"}, {"firstChar": "L", "id": 22, "name": "LG"}]]
```

## 8. Git 管理代码

### 8.1. 提交到本地仓库

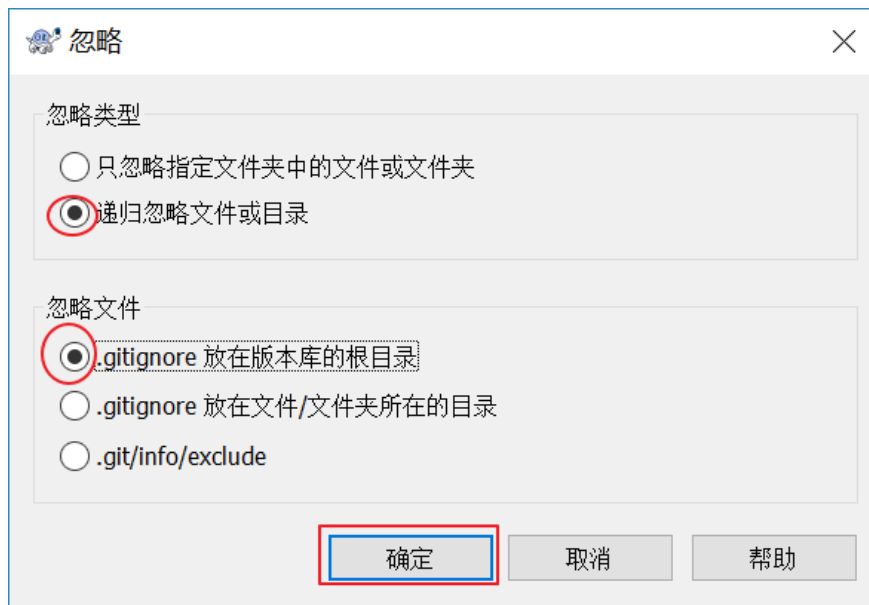
#### 8.1.1. 创建本地仓库



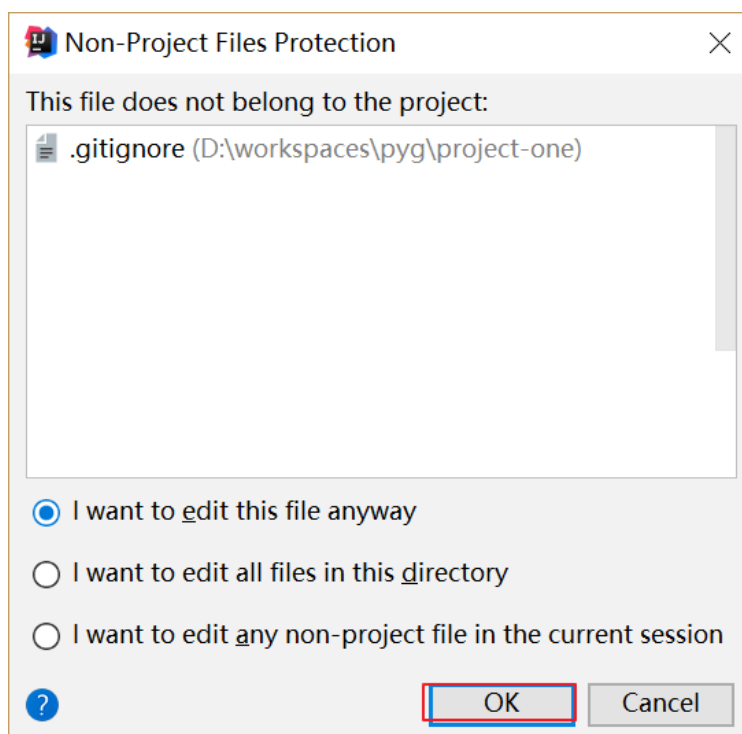


## 8.1.2. 添加忽略

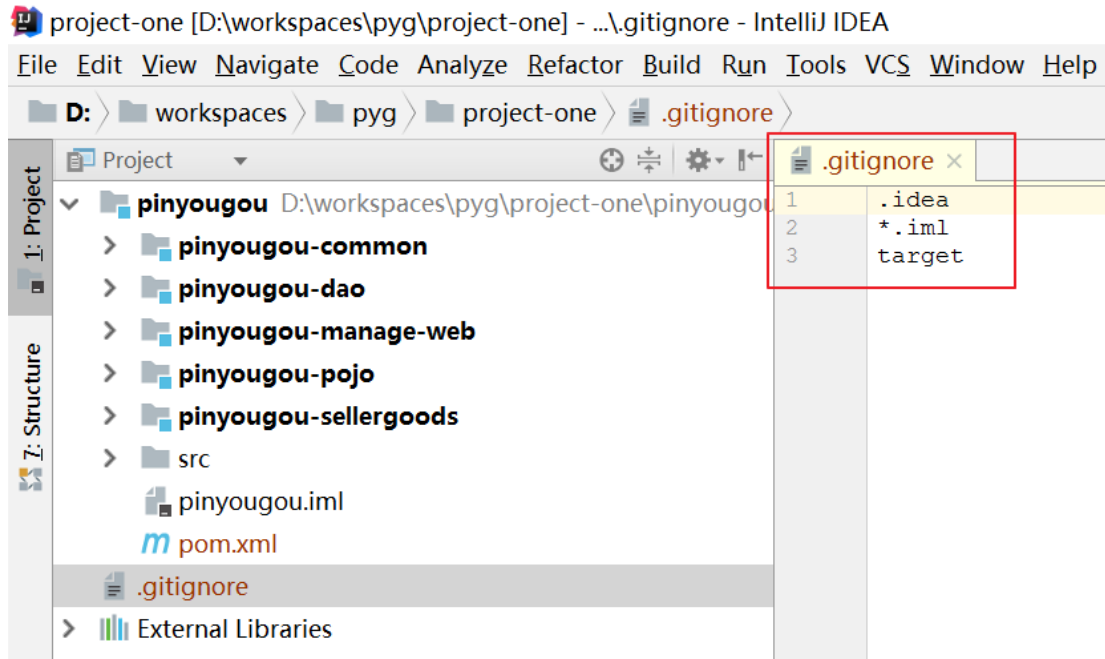




回到 IDEA 工具；双击打开“.gitignore”文件

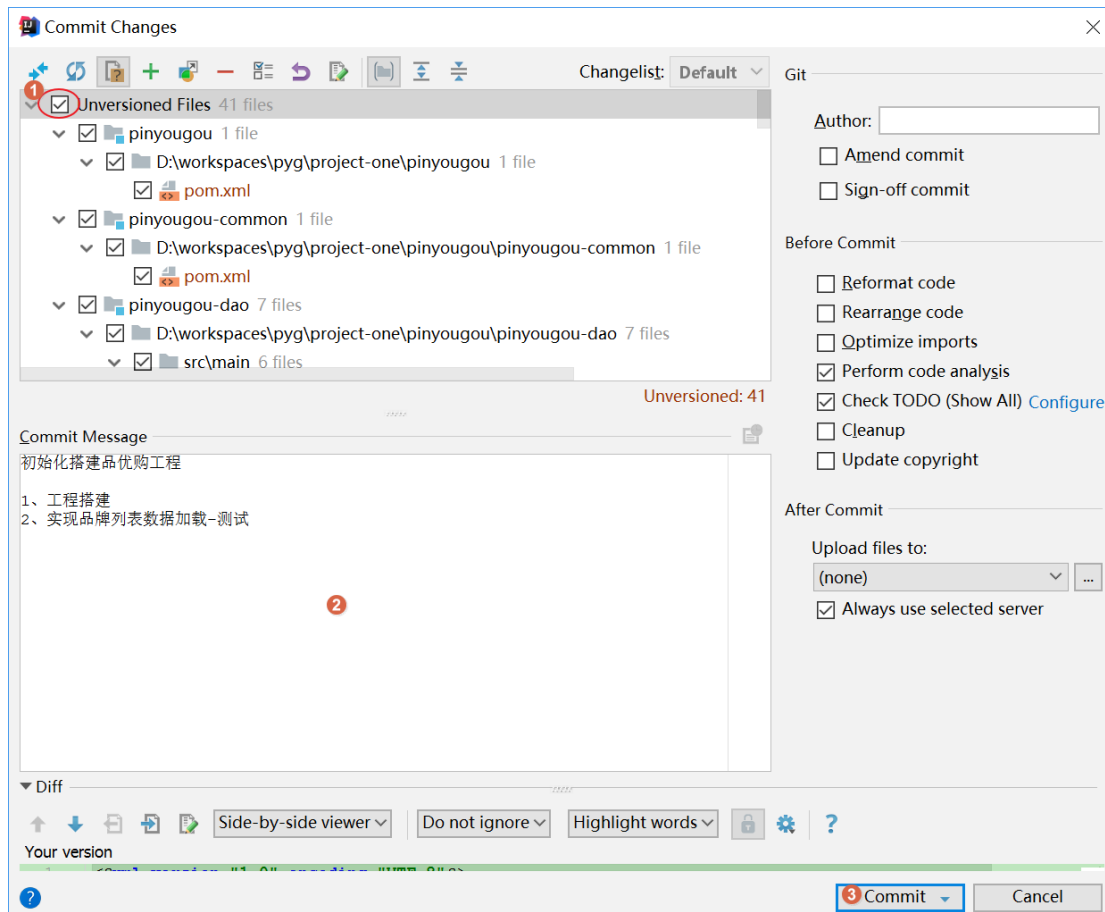


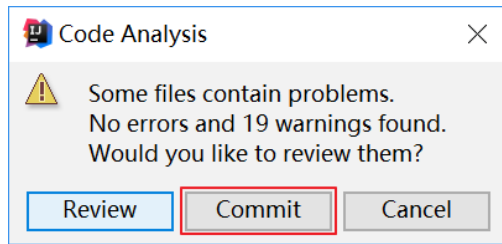




### 8.1.3. 提交

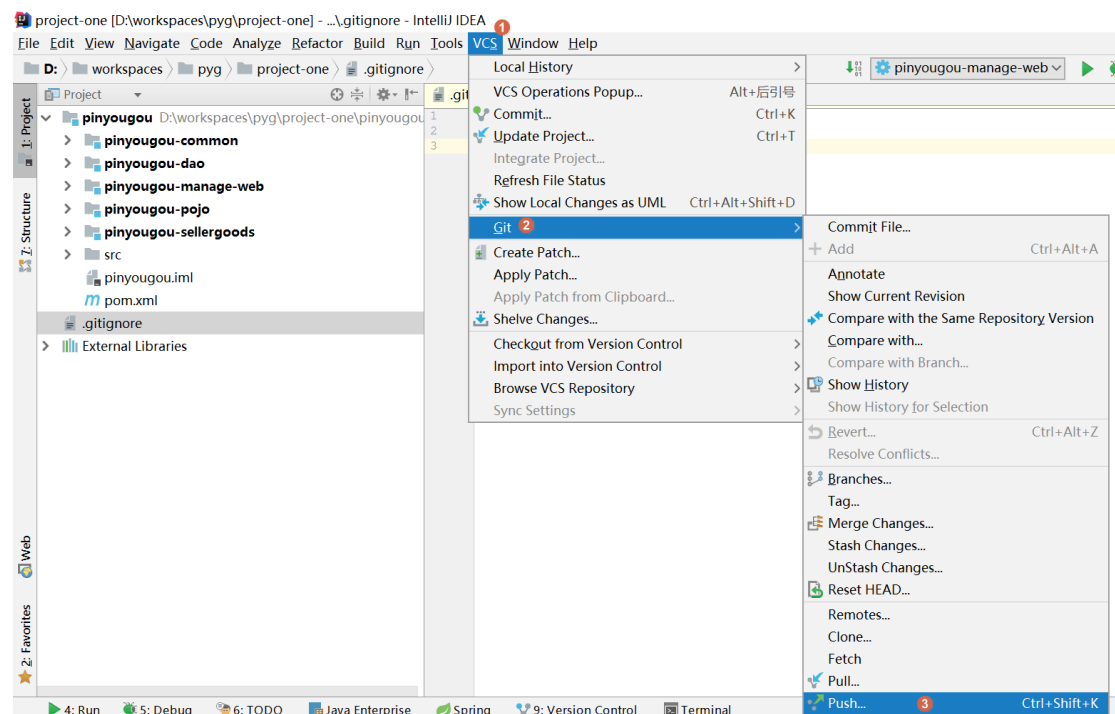
按 ctrl + K

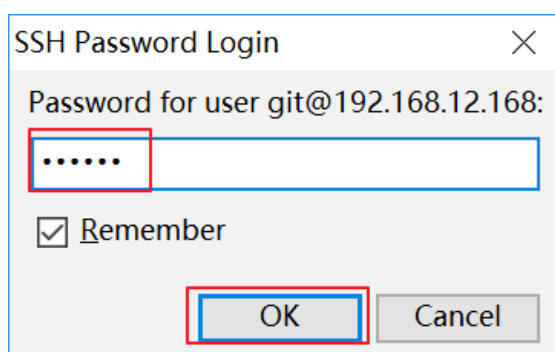
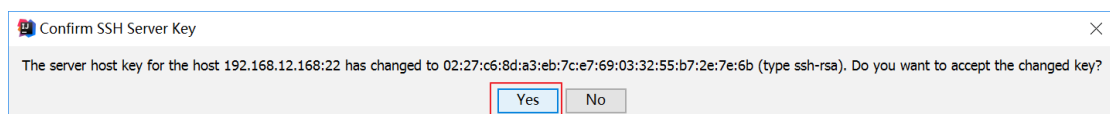
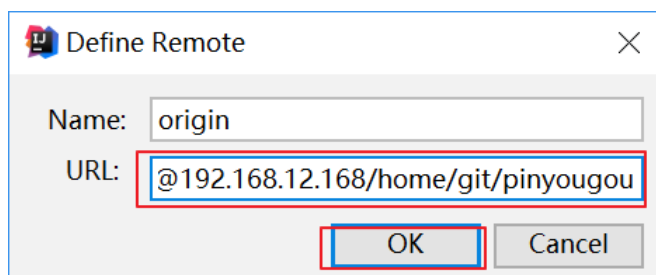
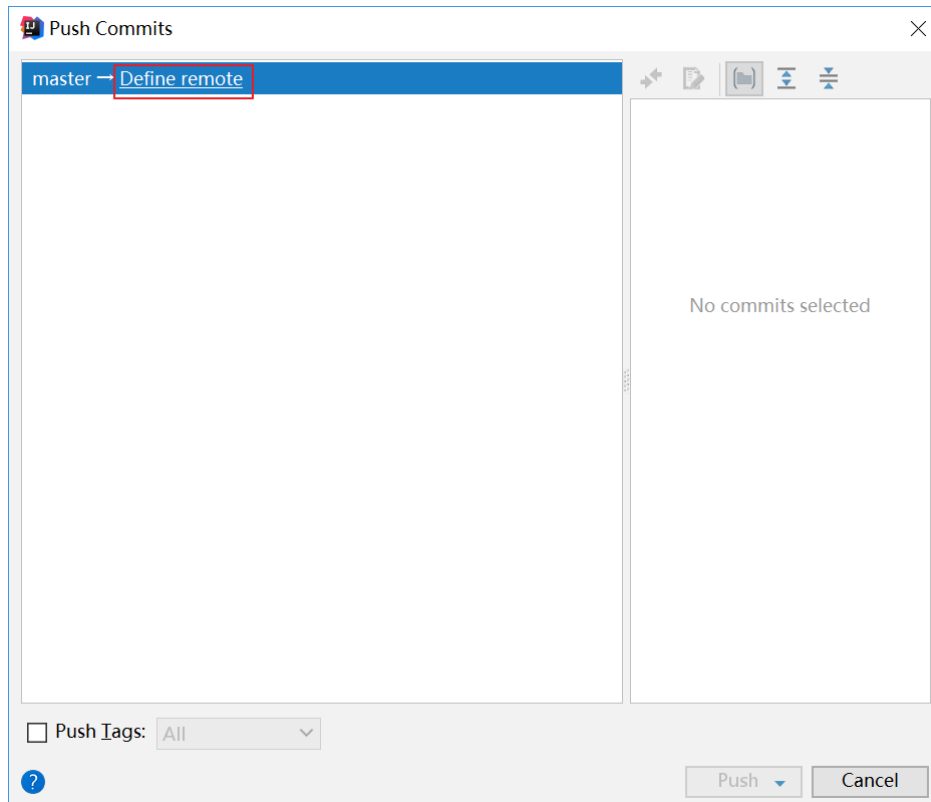


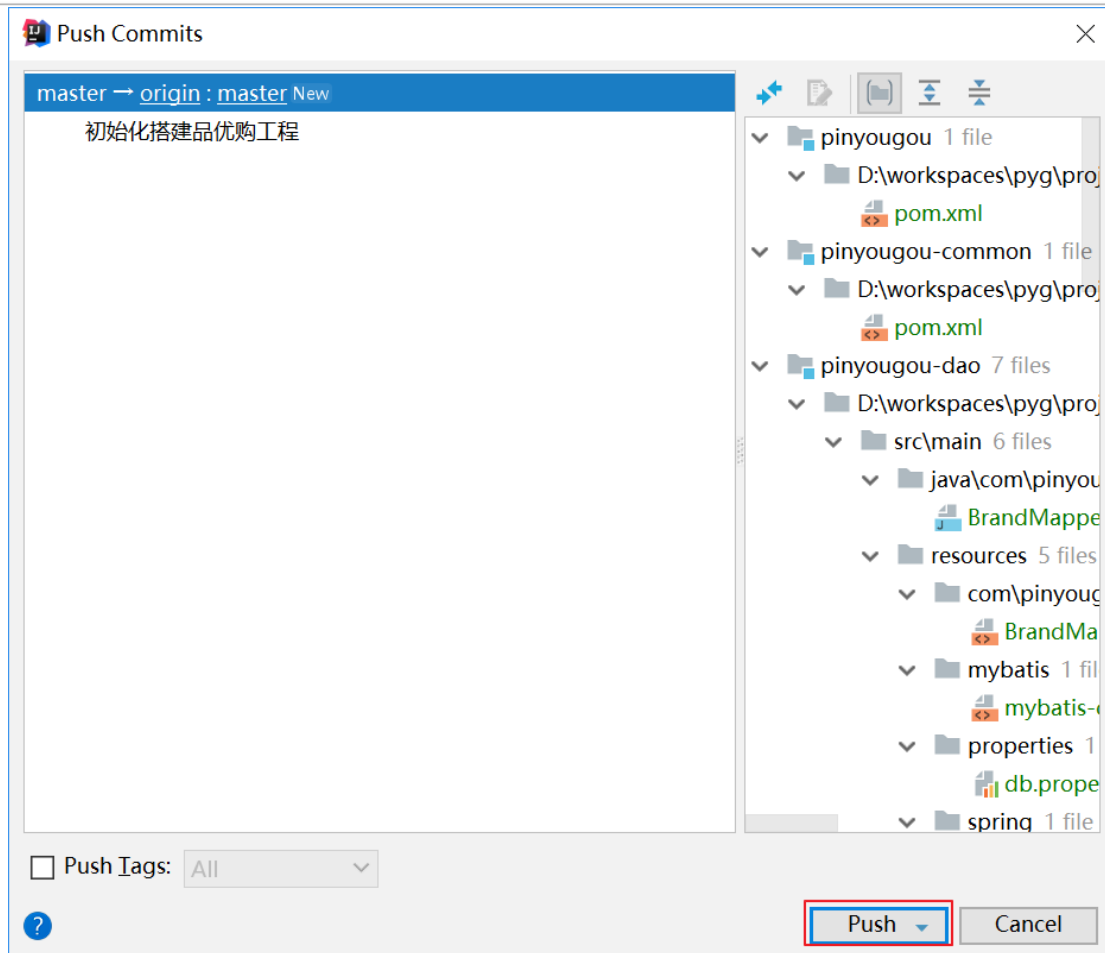


## 8.2. 推送到远程仓库

推送到的远程仓库为：



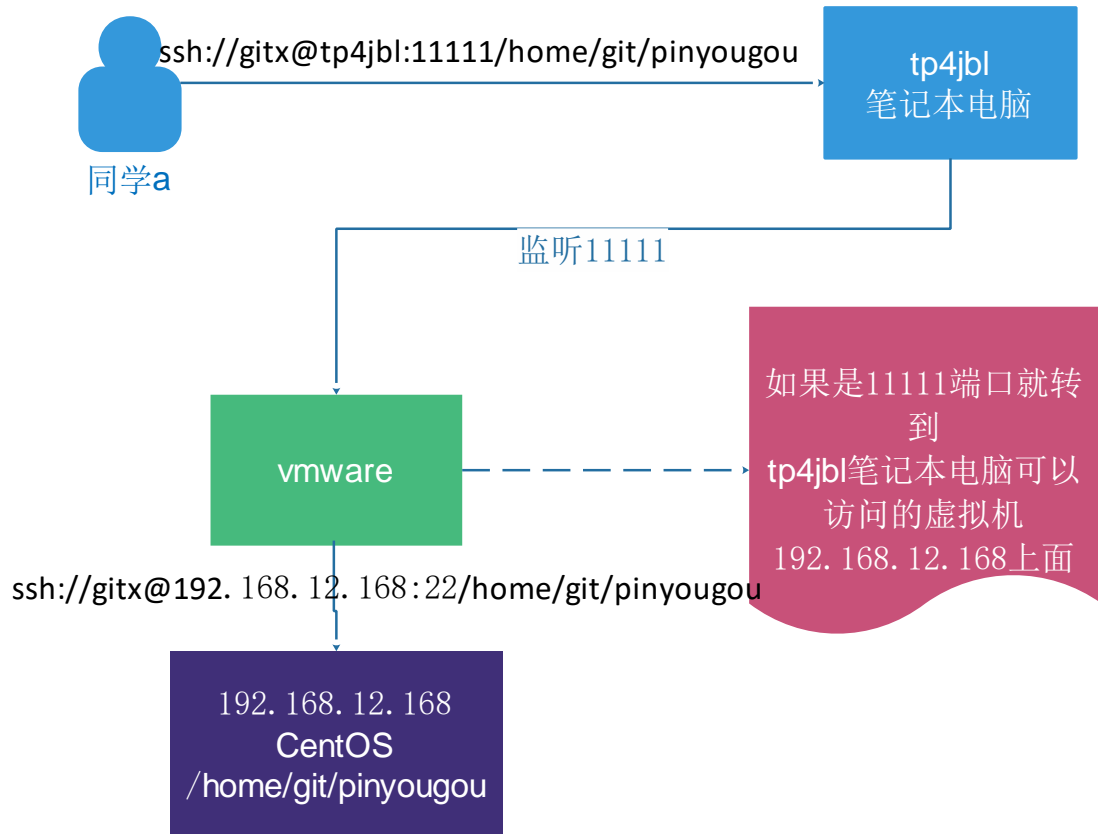






## 8.3. 克隆代码

### 8.3.1. 克隆地址访问流程



### 8.3.2. 克隆远程仓库

#### 8.3.2.1. 克隆地址说明

这个远程仓库就是品优购上课时提交的代码仓库；以后大家都到这个地方拉取代码到自己本地仓库。

克隆地址：ssh://gitx@tp4jbl:11111/home/git/pinyougou

用户名：gitx 密码：123456

使用 gitx 用户连接的时候，只能拉取、克隆代码；但是不能推送本地仓库的代码到远程仓库。

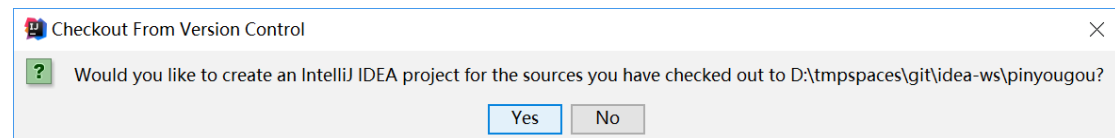
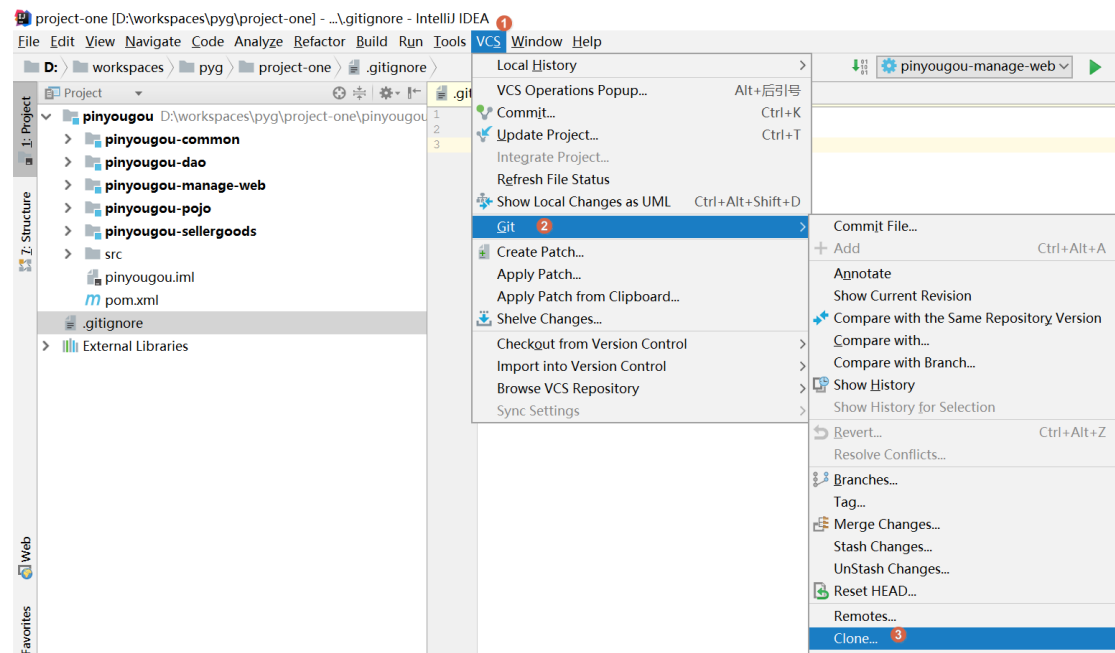


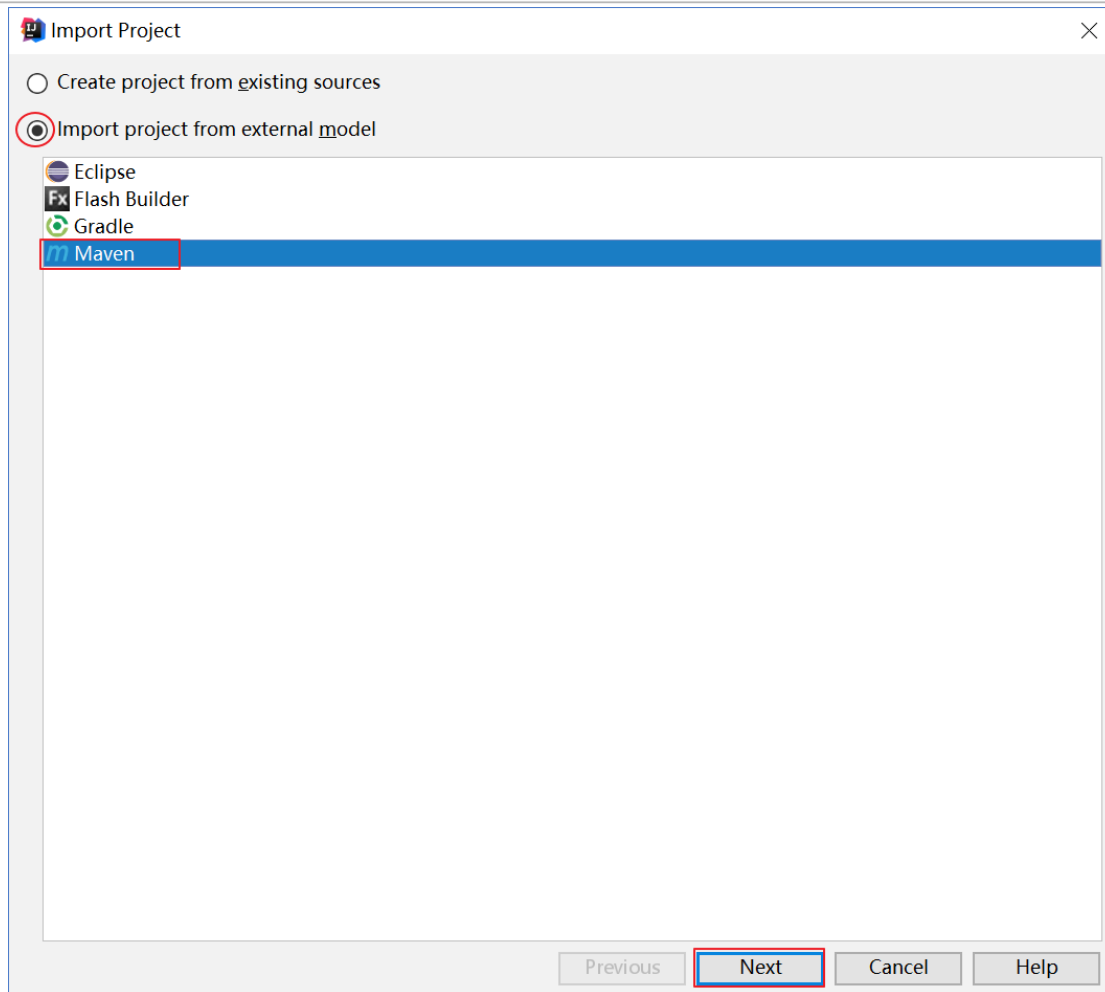
### 8.3.2.2. 克隆远程仓库

这个操作也就是大家从我提供的远程仓库：

ssh://gitx@tp4jbl:11111/home/git/pinyougou

中获取代码，以后每天的代码更新也是从这个地方拉取。具体的克隆步骤如下：







Import Project ×

Root directory:

☒ Search for projects recursively

Project format:

☐ Keep project files in:

☒ Import Maven projects automatically

☒ Create IntelliJ IDEA modules for aggregator projects (with 'pom' packaging)

☐ Create module groups for multi-module Maven projects

☒ Keep source and test folders on reimport

☒ Exclude build directory (%PROJECT\_ROOT%/target)

☒ Use Maven output directories

Generated sources folders:

Phase to be used for folders update:

IDEA needs to execute one of the listed phases in order to discover all source folders that are configured via Maven plugins.

**Note** that all test-\* phases firstly generate and compile production sources.

Automatically download: ☐ Sources ☐ Documentation

Dependency types:

Comma separated list of dependency types that should be imported

Environment settings...

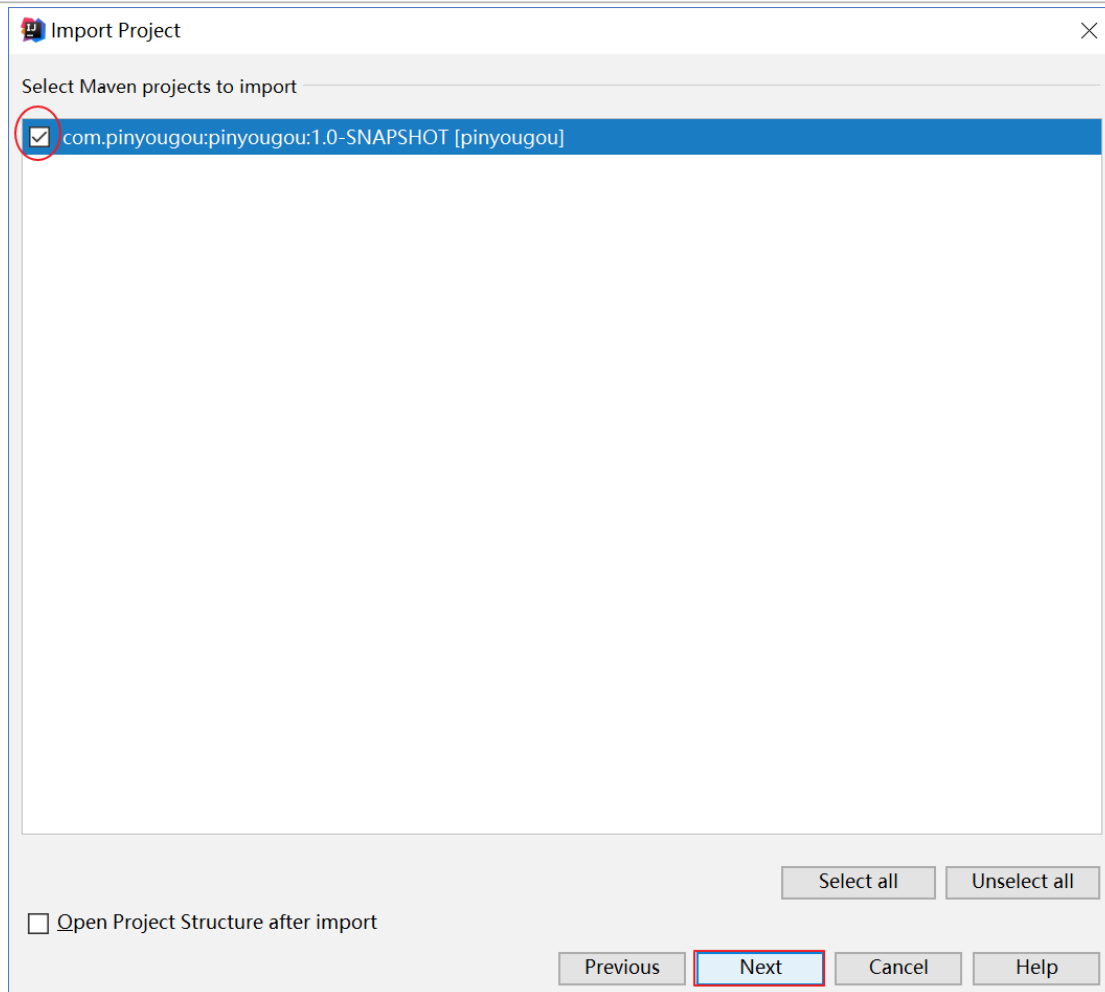
Previous

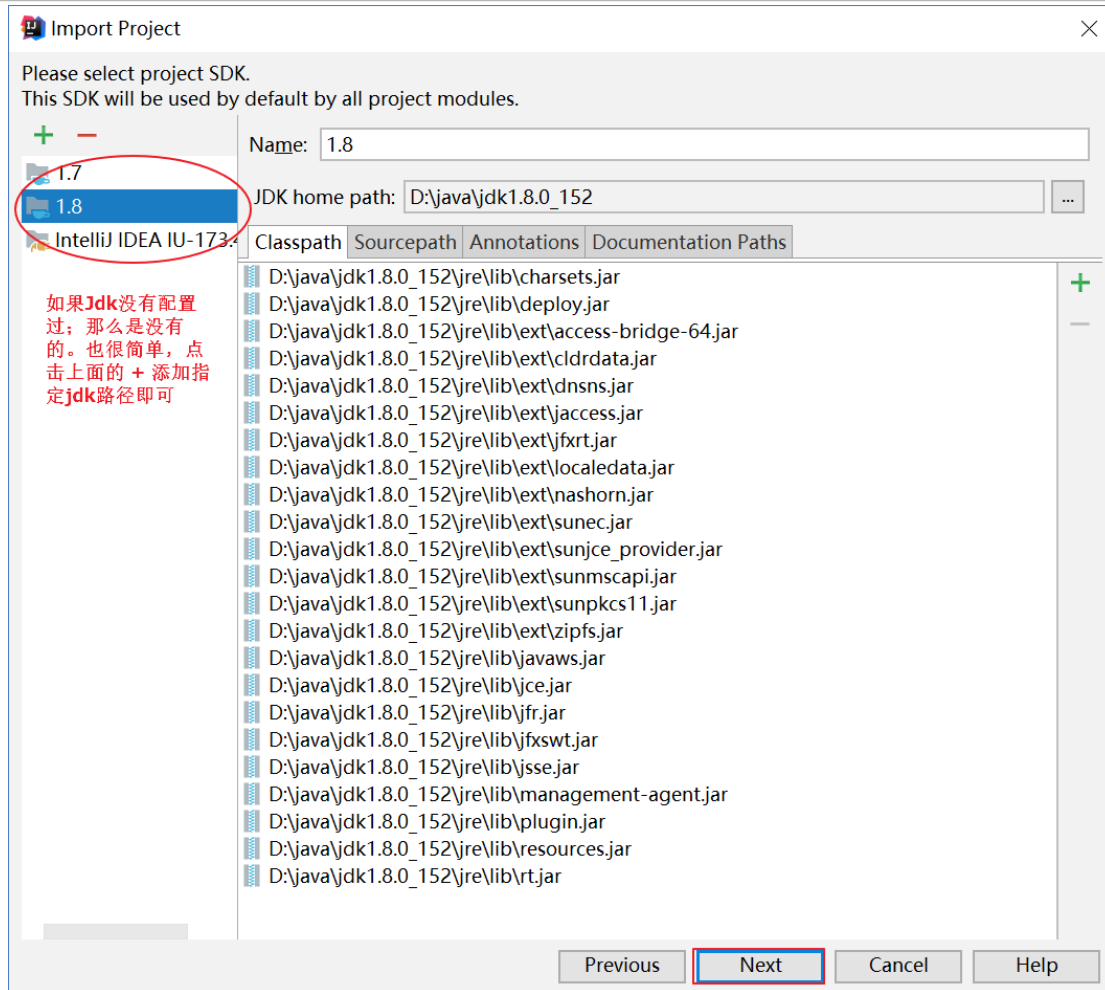
Next

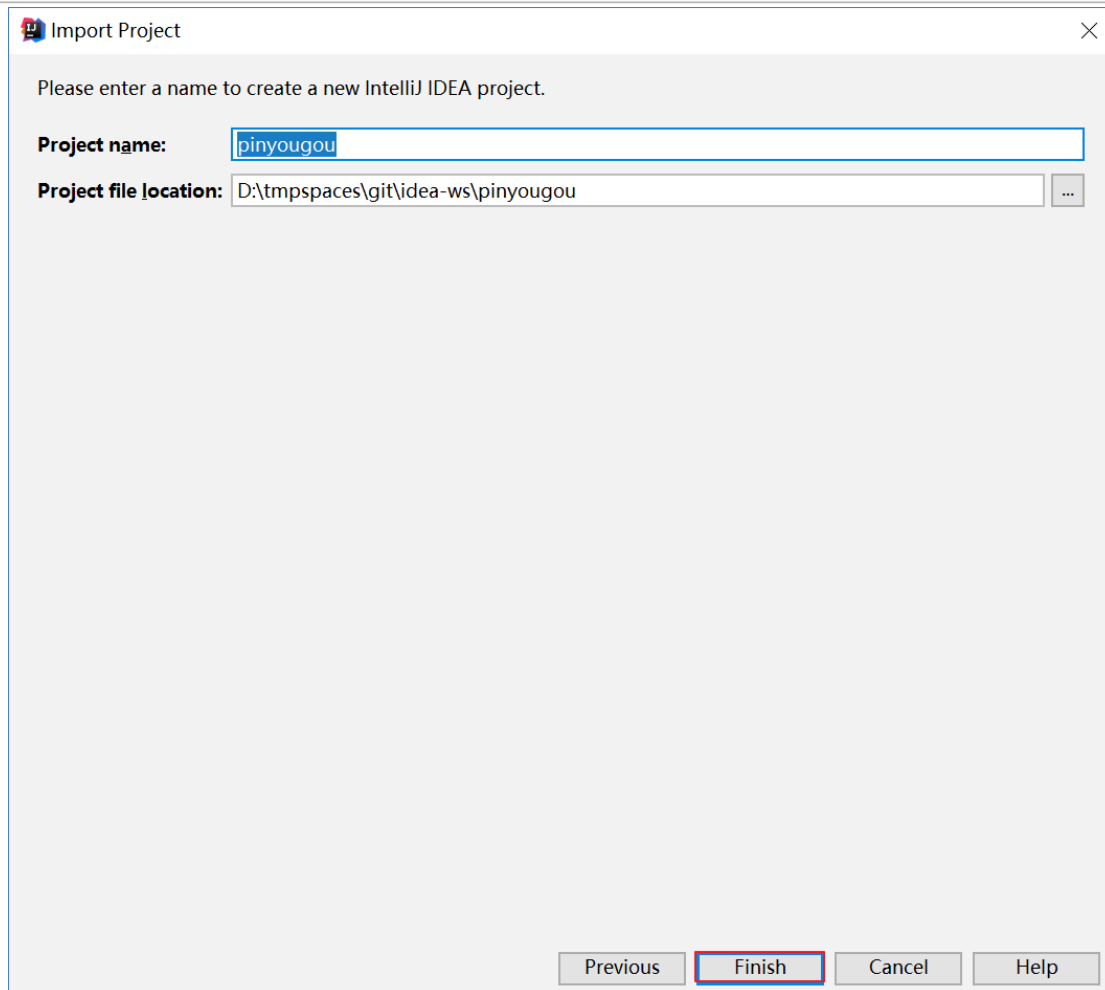
Cancel

Help



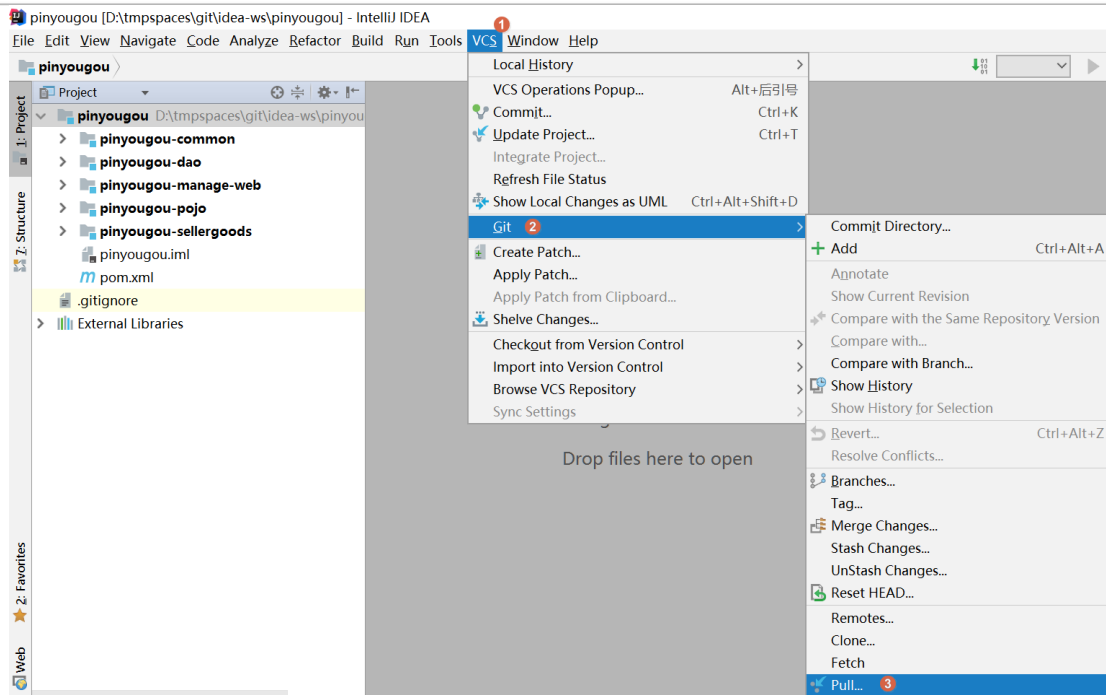




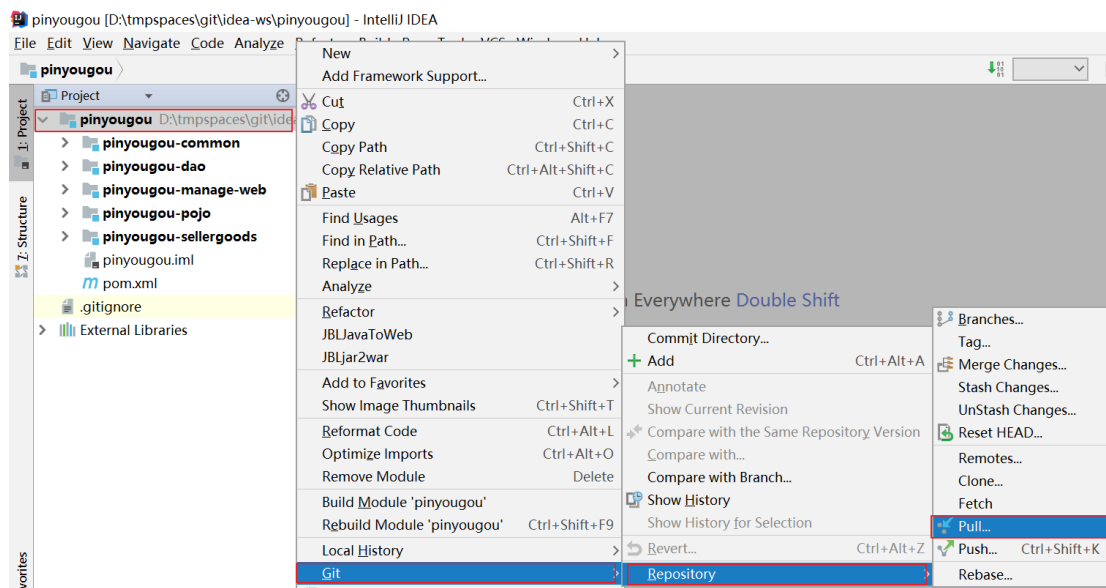


## 8.4. 拉取代码

以后每天从远程仓库拉取代码；则可以如下操作：



或者如下：



## 9. 常见问题

### 9.1. 找不到服务

```
java.lang.IllegalStateException: Failed to check the status of the service  
com.pinyougou.sellergoods.service.BrandService. No provider available for the  
service com.pinyougou.sellergoods.service.BrandService from the url
```

检查@Service 注解是否为 com.alibaba.dubbo.config.annotation.Service；同样的如果在 BrandController 中调用 service 出现空指针异常，检查@Reference 注解是否为 com.alibaba.dubbo.config.annotation.Reference

### 9.2. Mybatis 中查询语句找不到

```
org.springframework.web.util.NestedServletException: Request processing failed;  
nested exception is java.lang.RuntimeException:  
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found):  
com.pinyougou.mapper.BrandMapper.queryAll  
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found):  
com.pinyougou.mapper.BrandMapper.queryAll
```

在 IDEA 中检查 resources 文件夹中 com.pinyougou.mapper 是不是已经变成了一个目录；真正的是三级目录，只是在 IDEA 上看不太出区别。请创建的时候创建 package 方式创建则不会出现；不管如何请保证 resources/com/pinyougou/mapper 目录的存在

### 9.3. 启动 tomcat 插件异常

```
java.util.concurrent.ExecutionException: org.apache.catalina.LifecycleException:
```



```
Failed to start component
[StandardEngine[Tomcat].StandardHost[localhost].StandardContext[]]
    at java.util.concurrent.FutureTask.report(FutureTask.java:122)
    at java.util.concurrent.FutureTask.get(FutureTask.java:192)
    at
org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1123)
    at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:800)
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
    at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1559)
    at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1549)
    at java.util.concurrent.FutureTask.run$$$capture(FutureTask.java:266)
    at java.util.concurrent.FutureTask.run(FutureTask.java)
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused by: org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Tomcat].StandardHost[localhost].StandardContext[]]
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
    ... 7 more
Caused by: java.lang.ClassCastException:
org.springframework.web.SpringServletContainerInitializer cannot be cast to
javax.servlet.ServletContainerInitializer
```



```
at
org.apache.catalina.startup.ContextConfig.getServletContainerInitializer(ContextC
onfig.java:1670)
at
org.apache.catalina.startup.ContextConfig.getServletContainerInitializers(ContextC
onfig.java:1652)
at
org.apache.catalina.startup.ContextConfig.processServletContainerInitializers(Cont
extConfig.java:1562)
at
org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:1270)
at
org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:878)
at
org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:376)
at
org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java
:119)
at
org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:90)
at
org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5322)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 7 more
```

将 IDEA 的 maven 换一个新的；再将项目安装并启动