**品优购电商系统开发**

**第1章**

传智播客.黑马程序员

# 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Dubbox+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

# 课程计划

电商行业的背景

电商行业的技术

电商模式(知道)

品优购系统介绍

品优购系统架构介绍

dubbox/dubbo的框架的介绍

后台系统的搭建

# 电商行业

## 电商行业分析

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%。







电商行业的钱多。推荐找互联网的相关工作。

## 电商行业技术特点

 技术新 （LAMP）ebay ---->oralce--->redis(no sql数据库)

 技术范围广。java领域。

 分布式

 高并发、集群、负载均衡高可用（HA）。（99.999999999999999999%）是可用的。

海量数据

 业务复杂 (优惠)

 系统安全

分布式：同一个事情 拆开来做。

集群：同一个事情，大家一起做。

1024B=1KB 1024kb=1mb 1024mb=1GB 1024GB=1Tb 1024TB=1PB ....

## 主要电商模式

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。(以前的京东)。360buy

C2C：用户到用户。(淘宝,闲鱼)。

B2B2C：商家到商家到用户。(天猫、现在的京东、品优购)。

O2O：线上到线下。美团、饿了么。（online to offline）。

其他的概念：互联网金融

P2P:Person 2 Person,个人对个人的信贷，互联网的个人对个人的信贷平台。

# 2.品优购

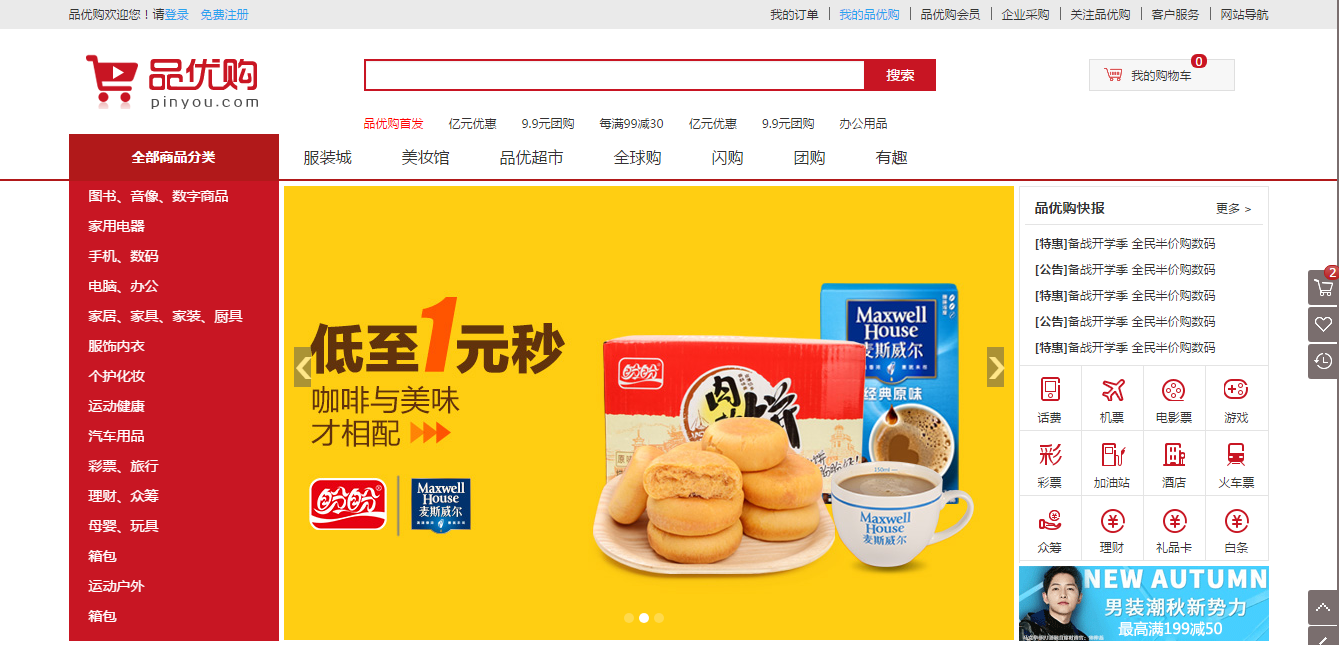
## 2.1品优购简介

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、商家管理后台三个子系统

### 2.1.1 网站前台

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



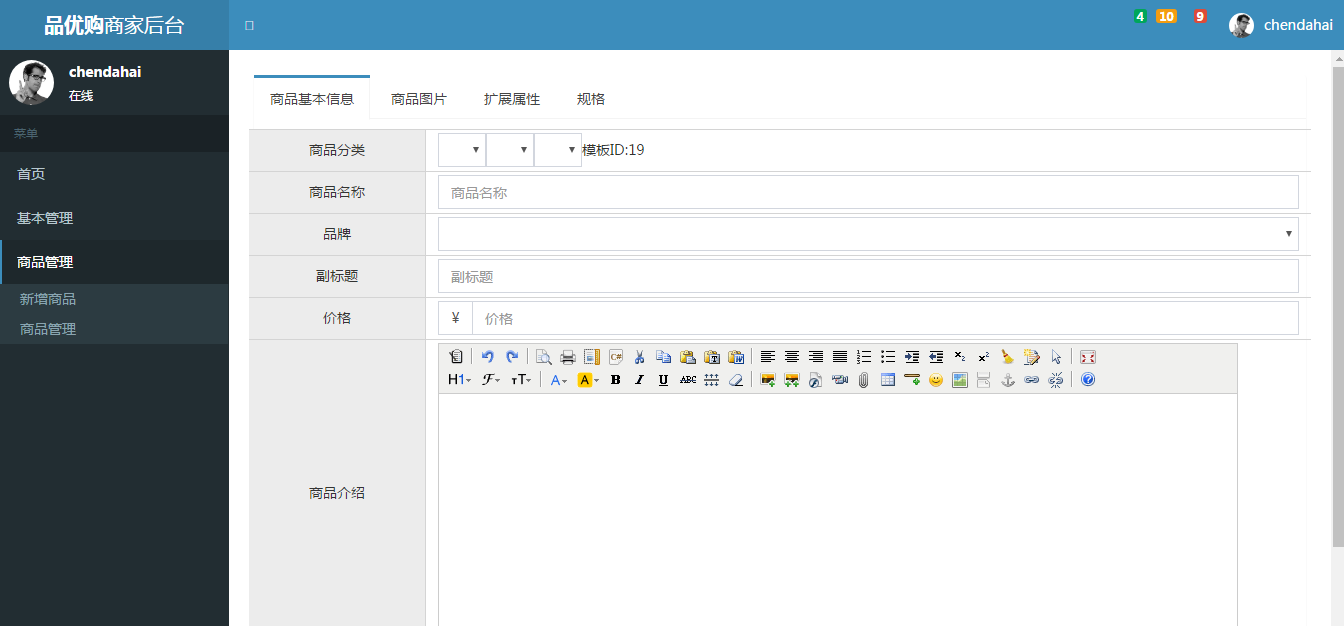
### 2.1.2 运营商后台

是运营商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。



### 2.1.3 商家管理后台

入驻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。



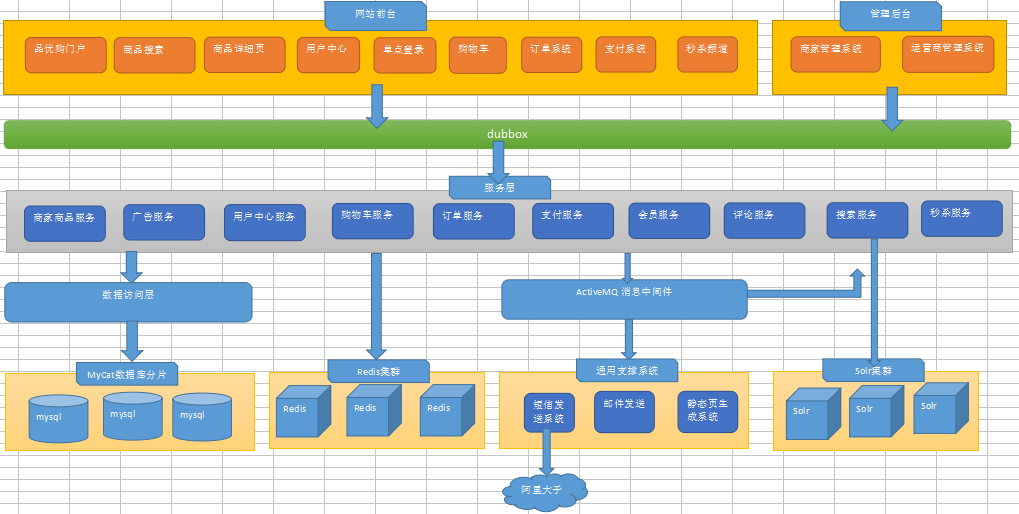
## 2.2系统架构

### 2.2.1 什么是SOA架构

并发量。

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。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。

### 2.2.2 品优购架构分析（SOA）



（清晰的架构图请看**品优购架构图.xlsx**）

## 2.3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 |

## 2.4框架组合

品优购采用当前流行的前后端编程架构。

后端框架采用Spring +SpringMVC+mybatis +Dubbox 。前端采用angularJS + Bootstrap。

spring springmvc mybatis

angularjs bootstrap

myql

dubbox（系统间通信）

activemq（消息的中间件）

mycat （数据库要读写分离 分库分表----数据库集群 mycat）

redis spring data redis

solr spring data solr

kindEditor （富文本编辑器）

fastdfs 分布式文件系统（OSS）

阿里大于 (短信服务 )

freemarker（模板类似JSP）

spring boot (微服务)

微信支付（支付流程）

spring security（类似shiro）

CAS (单点登录)

httpclient(solrj--->封装了httpclient)

nginx(负载均衡 反向代理 )

docker (服务器部署)

redis集群

solr集群

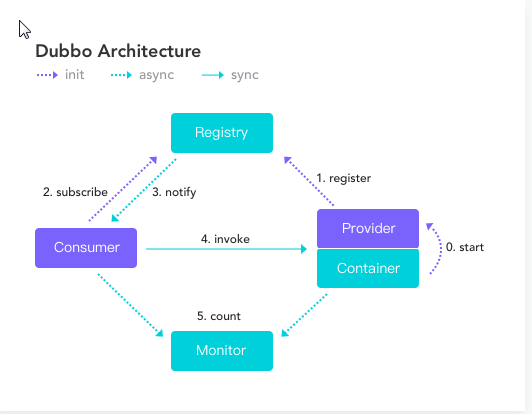
mysql集群

zookeeper集群。

# 3.Dubbo框架

## 3.1 Dubbo简介

Apache Dubbo (incubating) |ˈdʌbəʊ| 是一款高性能、轻量级的开源Java RPC框架，它提供了三大核心能力：面向接口的远程方法调用，智能容错和负载均衡，以及服务自动注册和发现。



**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: 统计服务的调用次调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。

## 注册中心

注册中心负责服务地址（ip和端口）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。使用dubbo-2.3.3以上版本，**官方建议使用Zookeeper作为注册中心。**

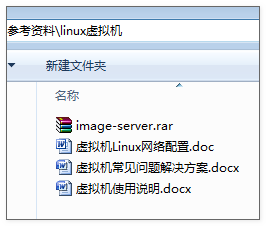
### zookeeper介绍

Zookeeper是Apache hadoop pig hive 。。。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(稳定性好)，可用于生产环境，并推荐使用。

注意：zookeeper是java开发的，所以需要使用JDK的环境。

### 环境准备

可以使用如下图所示的虚拟机：

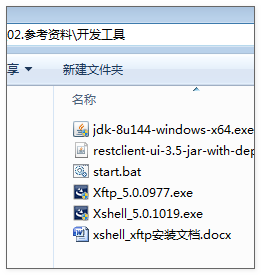


使用手册可以参考虚拟机使用说明。

开机。

### 使用xshell连接服务器

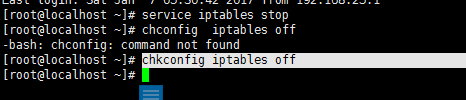
远程工具的安装请参考文档：xshell\_xftp安装文档.docx



连接linux时可能会连接不上，首先手动关闭防火墙：

service iptables stop 作用于当前系统状态，一旦重启，设置失效。

**chkconfig iptables off 永久关闭防火墙，如下图：**



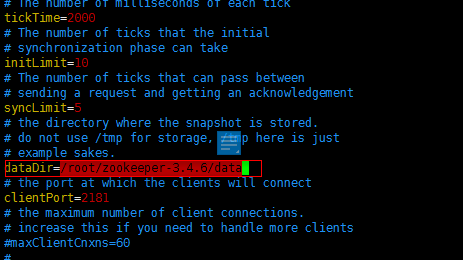
### zookeeper的安装

第一步：安装jdk

第二步：解压缩zookeeper压缩包

第三步：将conf文件夹下zoo\_sample.cfg复制一份，改名为zoo.cfg

第四步：修改配置dataDir属性，指定一个真实目录（进入zookeeper解压目录，创建data目录：mkdir data)



第五步：

启动zookeeper：bin/zkServer.sh start

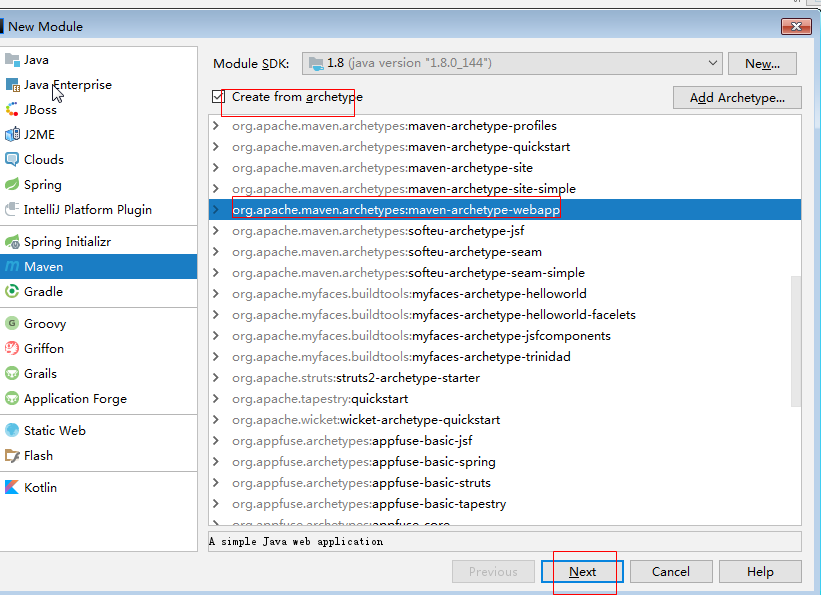
关闭zookeeper：bin/zkServer.sh stop

查看zookeeper状态：bin/zkServer.sh status

# dubbo入门使用

### 服务提供者方开发

#### 第一步：创建maven工程 (war)



#### 第二步：添加依赖

|  |
| --- |
| *<?***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"**>  <**modelVersion**>4.0.0</**modelVersion**>   <**groupId**>com.itheima</**groupId**>  <**artifactId**>itheima-dubbo-provider</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  <**packaging**>war</**packaging**>   <**name**>itheima-dubbo-provider Maven Webapp</**name**>  *<!--* ***FIXME change it to the project's website*** *-->* <**url**>http://www.example.com</**url**>   <**properties**>  *<!--设置编译的版本使用1.8版本-->* <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  <**spring.version**>4.3.10.RELEASE</**spring.version**>  <**dubbo.version**>2.5.9</**dubbo.version**>  <**zookeeper.version**>3.4.6</**zookeeper.version**>  <**zkclient.version**>0.1</**zkclient.version**>   </**properties**>    <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**version**>4.12</**version**>  <**scope**>test</**scope**>  </**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**>${spring.version}</**version**>  </**dependency**>  *<!--添加dubbo的依赖-->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>dubbo</**artifactId**>  <**version**>${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**>   </**dependencies**>   <**build**>  <**finalName**>itheima-dubbo-provider</**finalName**>   <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  <**path**>/</**path**>  <**port**>8081</**port**>  </**configuration**>  </**plugin**>  </**plugins**>   </**build**> </**project**> |

#### 第三步：创建接口和实现类

|  |
| --- |
| **package** com.itheima.dubbo.service.impl;  **import** com.alibaba.dubbo.config.annotation.Service; **import** com.itheima.dubbo.service.TestService;  /\*\*  \* 描述  \*  \* **@author** 三国的包子  \* **@version** 1.0  \* **@package** com.itheima.dubbo.service.impl  \* **@since** 1.0  \*/ @Service*//使用dubbo的Service注解* **public class** TestServiceImpl **implements** TestService {  @Override  **public** String sayhello() {  **return "hello world"**;  } } |

#### 第四步：配置spring配置文件

在/src/main/resources下创建配置文件

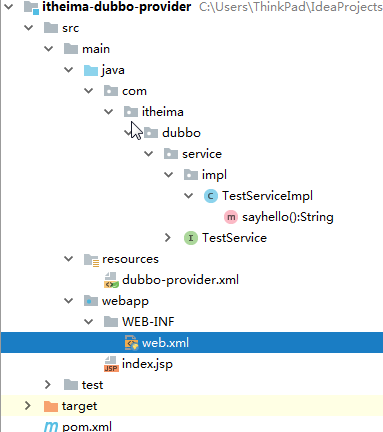
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd"**>   *<!--应用名称唯一-->* <**dubbo:application name="dubbo-provider"**></**dubbo:application**>  *<!--设置注册中心的配置-->* <**dubbo:registry protocol="zookeeper" address="192.168.25.128" port="2181"**></**dubbo:registry**>   *<!--暴露服务的端口需要配置，默认的情况下可以不配置就是20880-->* <**dubbo:protocol name="dubbo" port="20880"**></**dubbo:protocol**>  *<!--扫描组件-->* <**dubbo:annotation package="com.itheima.dubbo.service"**></**dubbo:annotation**>   </**beans**> |

#### 第五步：发布服务

|  |
| --- |
| *<?***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"  id="WebApp\_ID" version="2.5"**>  <**welcome-file-list**>  <**welcome-file**>index.jsp</**welcome-file**>  </**welcome-file-list**>   <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:dubbo-provider.xml</**param-value**>  </**context-param**>  <**listener**>  <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>   </**web-app**> |

#### 第六步:启动项目

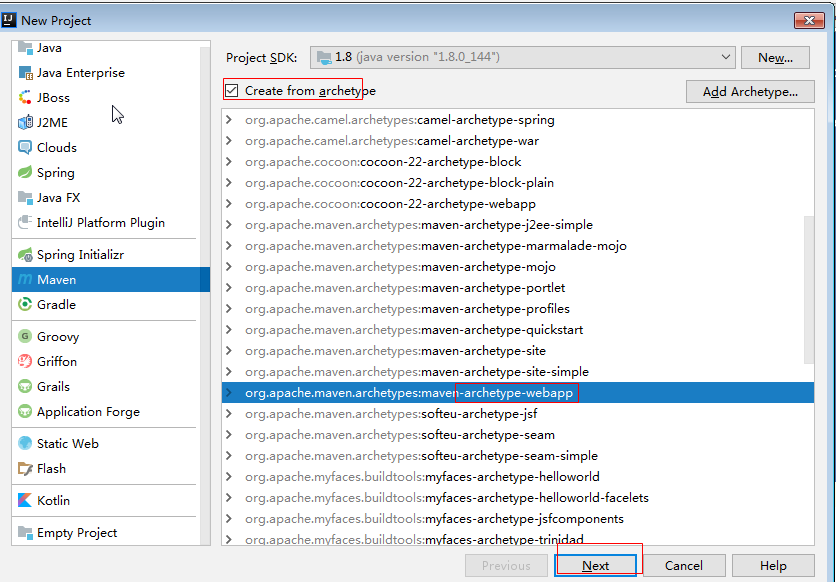
整体结构：



先启动zookeeper,再启动tomcat

### 服务消费者方开发

#### 第一步：创建maven工程(war)



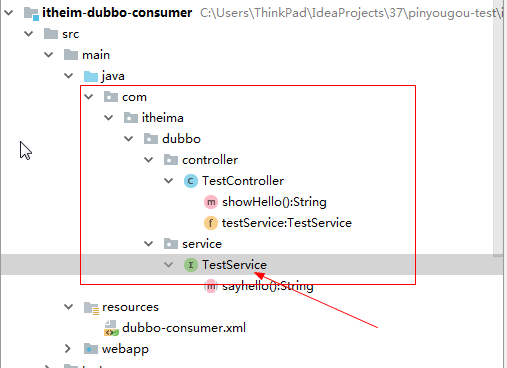
#### 第二步：添加依赖

|  |
| --- |
| *<?***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"**>  <**modelVersion**>4.0.0</**modelVersion**>   <**groupId**>com.itheima</**groupId**>  <**artifactId**>itheim-dubbo-consumer</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  <**packaging**>war</**packaging**>   <**name**>itheim-dubbo-consumer Maven Webapp</**name**>  *<!--* ***FIXME change it to the project's website*** *-->* <**url**>http://www.example.com</**url**>   <**properties**>  *<!--设置编译的版本使用1.8版本-->* <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  <**spring.version**>4.3.10.RELEASE</**spring.version**>  <**dubbo.version**>2.5.9</**dubbo.version**>  <**zookeeper.version**>3.4.6</**zookeeper.version**>  <**zkclient.version**>0.1</**zkclient.version**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**version**>4.12</**version**>  <**scope**>test</**scope**>  </**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**>${spring.version}</**version**>  </**dependency**>  *<!--添加dubbo的依赖-->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>dubbo</**artifactId**>  <**version**>${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**>  </**dependencies**>   <**build**>  <**finalName**>itheim-dubbo-consumer</**finalName**>   <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  <**path**>/</**path**>  <**port**>8082</**port**>  </**configuration**>  </**plugin**>  </**plugins**>   </**build**> </**project**> |

#### 第三步：创建接口和controller

注意：创建的接口要和服务提供者方对应的接口完全一致。包名和方法名和接口名都要一致。

|  |
| --- |
| **package** com.itheima.dubbo.service;  /\*\*  \* 描述  \*  \* **@author** 三国的包子  \* **@version** 1.0  \* **@package** com.itheima.dubbo.service  \* **@since** 1.0  \*/ **public interface** TestService {  **public** String sayhello(); } |



创建Controller:

|  |
| --- |
| **package** com.itheima.dubbo.controller;  **import** com.alibaba.dubbo.config.annotation.Reference; **import** com.itheima.dubbo.service.TestService; **import** org.springframework.beans.factory.annotation.Autowired; **import** org.springframework.stereotype.Controller; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.ResponseBody; **import** org.springframework.web.bind.annotation.RestController;  /\*\*  \* 描述  \*  \* **@author** 三国的包子  \* **@version** 1.0  \* **@package** com.itheima.dubbo.controller  \* **@since** 1.0  \*/ @RestController **public class** TestController {    @Reference  **private** TestService **testService**;    @RequestMapping(**"/sayhello"**)  **public** String showHello() {  **return testService**.sayhello();  } } |

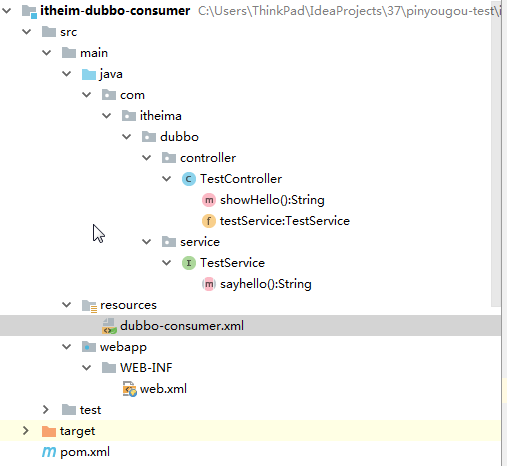
#### 第四步：配置springmvc.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  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://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd"**>    *<!--配置应用名称唯一-->* <**dubbo:application name="dubbo-consumer"**></**dubbo:application**>  *<!--注册中心地址-->* <**dubbo:registry protocol="zookeeper" address="192.168.25.128" port="2181"**></**dubbo:registry**>   *<!--扫描注解@reference所在包-->* <**dubbo:annotation package="com.itheima.dubbo.controller"**></**dubbo:annotation**>   *<!--spring组件扫描-->* <**context:component-scan base-package="com.itheima.dubbo.controller"**></**context:component-scan**>  *<!--注解驱动-->* <**mvc:annotation-driven**></**mvc:annotation-driven**>  </**beans**> |

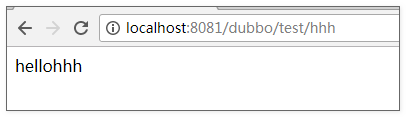
#### 第五步：配置web.xml

|  |
| --- |
| *<?***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"  id="WebApp\_ID" version="2.5"**>  <**welcome-file-list**>  <**welcome-file**>index.jsp</**welcome-file**>  </**welcome-file-list**>    *<!--前端控制器-->* <**servlet**>  <**servlet-name**>springmvc</**servlet-name**>  <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:dubbo-consumer.xml</**param-value**>  </**init-param**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>springmvc</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>  </**web-app**> |

#### 第六步：启动测试



先启动zookeeper 再启动tomcat



## 监控中心安装

### 准备好war包

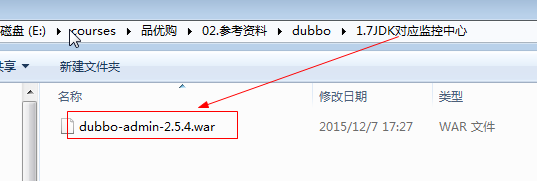
监控中心就是一个war包，将其部署在tomcat中即可。

注意下

如果你的linux虚拟机是jdk1.7 就使用jdk1.7对应监控中心war。

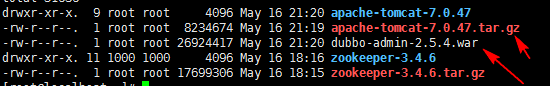
如果你的linux虚拟机是jdk1.8 就使用jdk1.8对应监控中心war。

我这里选择：1.7



以及准备好：tomcat

### 上传到linux系统中



### 部署

需要安装先tomcat，然后部署监控中心即可。

1、部署监控中心：

|  |
| --- |
| [root@localhost ~]# cp dubbo-admin-2.5.4.war apache-tomcat-7.0.47/webapps/dubbo-admin.war |

1. 启动tomcat

cd bin 目录，输入：./startup.sh

查询tomcat是否已经启动：ps -ef|grep tomcat

1. 访问http://192.168.25.128:8080/dubbo-admin/

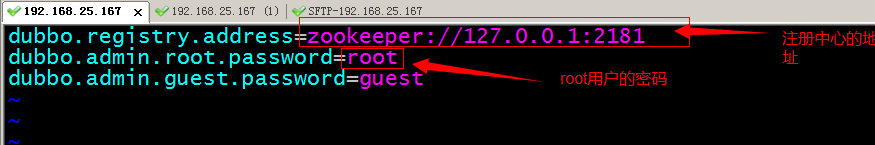
用户名：root

密码：root

如果监控中心和注册中心在同一台服务器上，可以不需要任何配置。

如果不在同一台服务器，需要修改配置文件：

/root/apache-tomcat-7.0.47/webapps/dubbo-admin/WEB-INF/dubbo.properties



如果要运行监控中心，必须先启动zookeeper. 监控中心对于dubbo的服务的调用来说不是必须的，不安装也可以运行。安装的目的是为了更好的统计其调用的次数，方便管理。

## 工程搭建

### 工程结构分析

pinyougou-parent：父工程，打包方式pom，仅仅管理jar包的统一版本版本号。

| 项目中所有工程都应该继承父工程。

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

|-- pinyougou-dao：通用的数据访问层,Mapper等。打包方式jar

|--pinyougou-pojo：通用的数据库表pojo。打包方式Jar

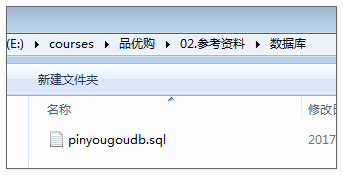
|--pinyougou-xxx-interface：某服务的接口工程。打包方式jar(dubbo官方推荐单独打包)

|--pinyougou-xxx-service:某服务的接口实现工程。打包方式war（用于发布服务）

|--pinyougou-xxx-web:某web工程。调用服务，并处理页面转发等。打包方式war。

### 创建数据库表

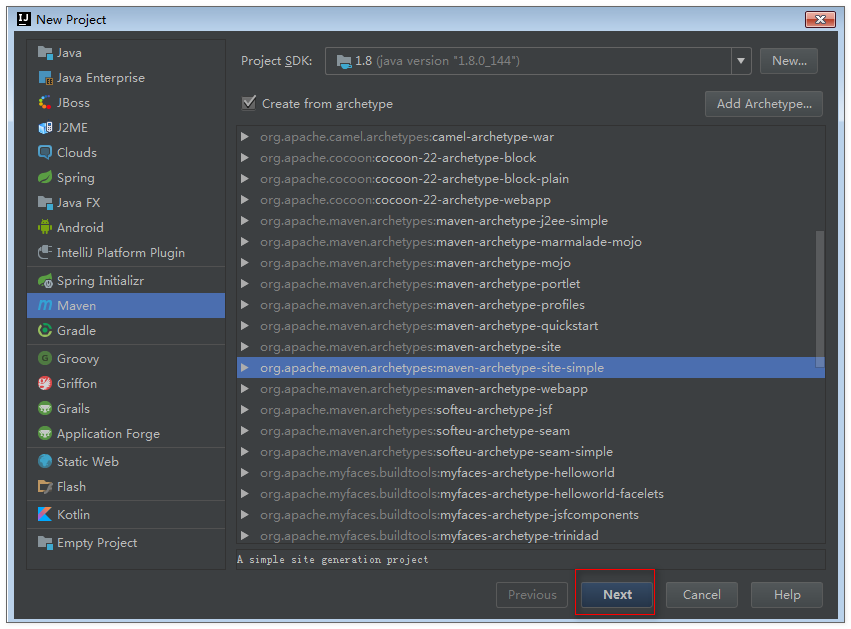
执行以下的sql脚本创建数据库。

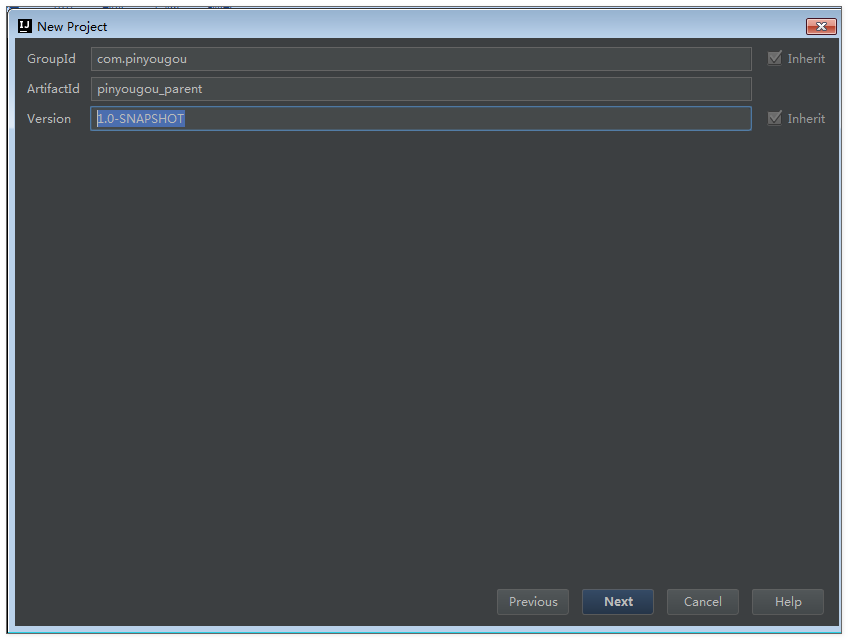


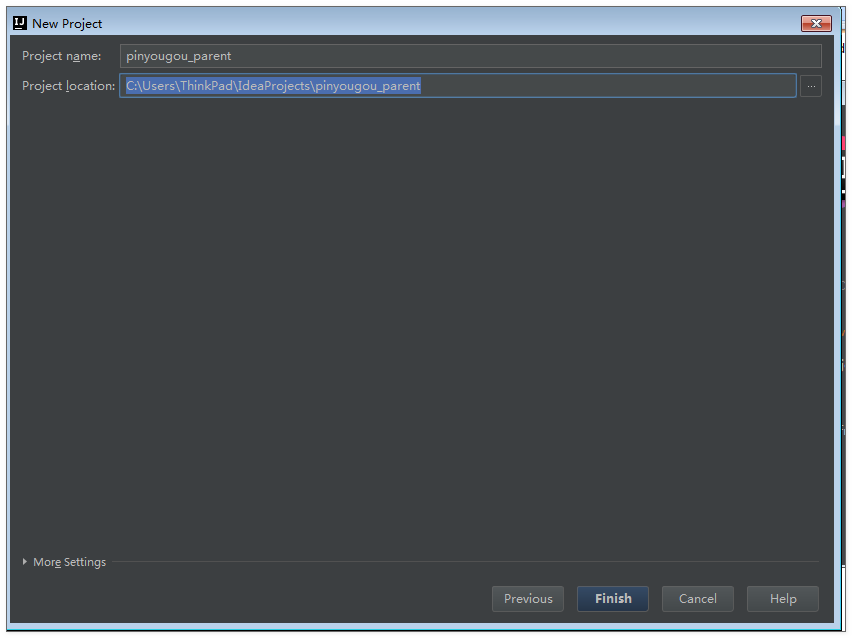
### 工程搭建

#### pinyougou-parent

##### 搭建截图





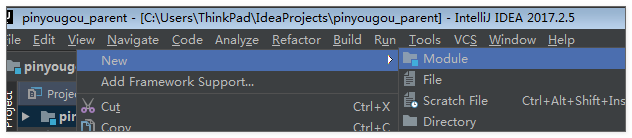


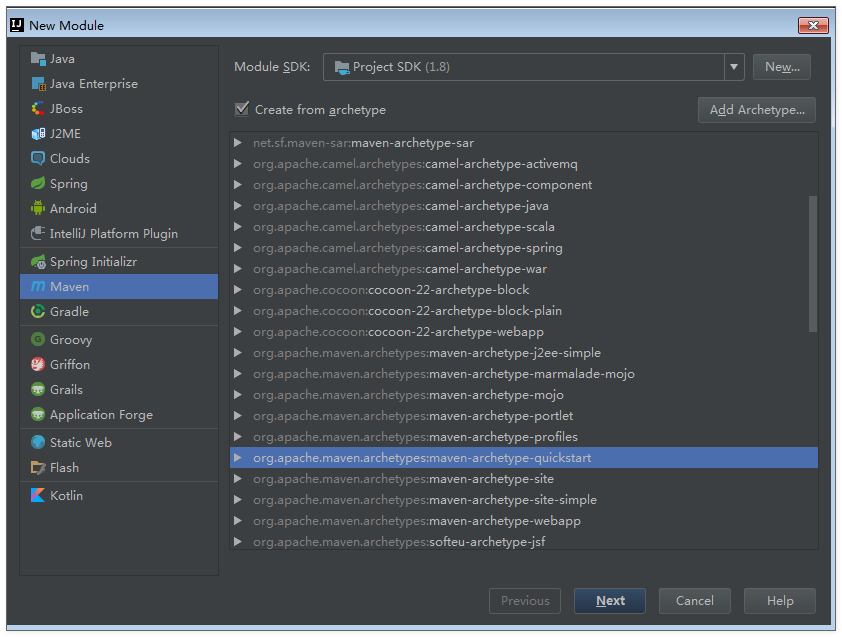
##### pom.xml

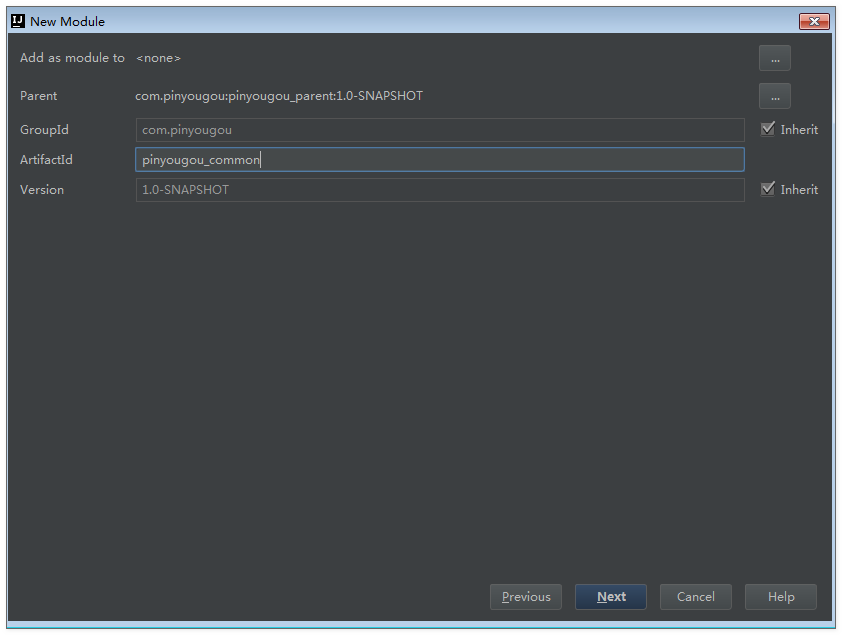
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**modelVersion**>4.0.0</**modelVersion**>   <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-parent</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  <**packaging**>pom</**packaging**>   <**properties**>  *<!--设置编译的版本使用1.8版本-->* <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  *<!--//和设置编译插件是一样的效果-->* <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  <**junit.version**>4.12</**junit.version**>  <**spring.version**>4.3.10.RELEASE</**spring.version**>  <**pagehelper.version**>4.0.0</**pagehelper.version**>  <**servlet-api.version**>2.5</**servlet-api.version**>  <**dubbo.version**>2.5.9</**dubbo.version**>  <**zookeeper.version**>3.4.7</**zookeeper.version**>  <**zkclient.version**>0.1</**zkclient.version**>  <**mybatis.version**>3.2.8</**mybatis.version**>  <**mybatis.spring.version**>1.2.2</**mybatis.spring.version**>  <**javassist.version**>3.20.0-GA</**javassist.version**>  <**mysql.version**>5.1.32</**mysql.version**>  <**druid.version**>1.0.9</**druid.version**>  <**commons-fileupload.version**>1.3.1</**commons-fileupload.version**>  <**freemarker.version**>2.3.23</**freemarker.version**>  <**solrj.version**>4.10.3</**solrj.version**>  <**ik.version**>2012\_u6</**ik.version**>  <**fastdfs-client.versioin**>1.25</**fastdfs-client.versioin**>  <**jackson.version**>2.8.9</**jackson.version**>  <**druid.version**>1.0.9</**druid.version**>  <**httpclient.version**>4.3.5</**httpclient.version**>  <**httpcore.version**>4.4.4</**httpcore.version**>  <**jstl.version**>1.2</**jstl.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**>  <**jsqlparser.version**>0.9.1</**jsqlparser.version**>  <**jedis.version**>2.8.2</**jedis.version**>  <**fastjson.version**>1.2.28</**fastjson.version**>  <**commons-codec.version**>1.10</**commons-codec.version**>  <**spring.data.redis**>1.7.2.RELEASE</**spring.data.redis**>  <**spring.security.version**>4.1.1.RELEASE</**spring.security.version**>  <**cas-client.version**>3.3.3</**cas-client.version**>  <**slf4j.version**>1.6.4</**slf4j.version**>  </**properties**>    <**dependencyManagement**>  <**dependencies**>  *<!-- 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**>  *<!-- Jackson Json处理工具包 -->* <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**>  <**version**>${jackson.version}</**version**>  </**dependency**>   *<!--alibaba json处理工具包-->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>fastjson</**artifactId**>  <**version**>${fastjson.version}</**version**>  </**dependency**>  *<!-- httpclient -->* <**dependency**>  <**groupId**>org.apache.httpcomponents</**groupId**>  <**artifactId**>httpclient</**artifactId**>  <**version**>${httpclient.version}</**version**>  </**dependency**>  *<!-- 日志处理 -->* <**dependency**>  <**groupId**>org.slf4j</**groupId**>  <**artifactId**>slf4j-log4j12</**artifactId**>  <**version**>${slf4j.version}</**version**>  </**dependency**>  *<!--单元测试-->* <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**version**>${junit.version}</**version**>  </**dependency**>   *<!-- Spring -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-context</**artifactId**>  <**version**>${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**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-test</**artifactId**>  <**version**>${spring.version}</**version**>  </**dependency**>  *<!-- dubbo相关 -->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>dubbo</**artifactId**>  <**version**>${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**>  <**groupId**>javassist</**groupId**>  <**artifactId**>javassist</**artifactId**>  <**version**>${javassist.version}</**version**>  </**dependency**>  <**dependency**>  <**groupId**>commons-codec</**groupId**>  <**artifactId**>commons-codec</**artifactId**>  <**version**>${commons-codec.version}</**version**>  </**dependency**>  *<!--jsp相关-->* <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>servlet-api</**artifactId**>  <**version**>${servlet-api.version}</**version**>  <**scope**>provided</**scope**>  </**dependency**>  *<!-- 文件上传组件 -->* <**dependency**>  <**groupId**>commons-fileupload</**groupId**>  <**artifactId**>commons-fileupload</**artifactId**>  <**version**>${commons-fileupload.version}</**version**>  </**dependency**>  *<!-- 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.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**>${druid.version}</**version**>  </**dependency**>  *<!-- 身份验证 -->* <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>spring-security-web</**artifactId**>  <**version**>${spring.security.version}</**version**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>spring-security-config</**artifactId**>  <**version**>${spring.security.version}</**version**>  </**dependency**>   <**dependency**>  <**groupId**>org.springframework.security</**groupId**>  <**artifactId**>spring-security-cas</**artifactId**>  <**version**>${spring.security.version}</**version**>  </**dependency**>  <**dependency**>  <**groupId**>org.jasig.cas.client</**groupId**>  <**artifactId**>cas-client-core</**artifactId**>  <**version**>${cas-client.version}</**version**>  *<!-- 排除log4j包冲突 -->* <**exclusions**>  <**exclusion**>  <**groupId**>org.slf4j</**groupId**>  <**artifactId**>log4j-over-slf4j</**artifactId**>  </**exclusion**>  </**exclusions**>  </**dependency**>  *<!--spring data redis-->* <**dependency**>  <**groupId**>org.springframework.data</**groupId**>  <**artifactId**>spring-data-redis</**artifactId**>  <**version**>${spring.data.redis}</**version**>  </**dependency**>  *<!-- redis客户端 -->* <**dependency**>  <**groupId**>redis.clients</**groupId**>  <**artifactId**>jedis</**artifactId**>  <**version**>${jedis.version}</**version**>  </**dependency**>  *<!--freemarker-->* <**dependency**>  <**groupId**>org.freemarker</**groupId**>  <**artifactId**>freemarker</**artifactId**>  <**version**>${freemarker.version}</**version**>  </**dependency**>  *<!--fastdfs客户端-->* <**dependency**>  <**groupId**>org.csource.fastdfs</**groupId**>  <**artifactId**>fastdfs</**artifactId**>  <**version**>${fastdfs-client.versioin}</**version**>  </**dependency**>  *<!-- solr客户端 -->* <**dependency**>  <**groupId**>org.apache.solr</**groupId**>  <**artifactId**>solr-solrj</**artifactId**>  <**version**>${solrj.version}</**version**>  </**dependency**>  <**dependency**>  <**groupId**>com.janeluo</**groupId**>  <**artifactId**>ikanalyzer</**artifactId**>  <**version**>2012\_u6</**version**>  </**dependency**>  </**dependencies**>  </**dependencyManagement**>   <**build**>  <**plugins**>  *<!--打包跳过测试-->* <**plugin**>  <**groupId**>org.apache.maven.plugins</**groupId**>  <**artifactId**>maven-surefire-plugin</**artifactId**>  <**version**>2.12.4</**version**>  <**configuration**>  <**skipTests**>true</**skipTests**>  </**configuration**>  </**plugin**>  *<!--maven清理target插件-->* <**plugin**>  <**groupId**>org.apache.maven.plugins</**groupId**>  <**artifactId**>maven-clean-plugin</**artifactId**>  <**version**>2.6.1</**version**>  <**configuration**>  <**filesets**>  <**fileset**>  <**directory**>${basedir}/pinyougou-common/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-pojo/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-dao/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-sellergoods-interface/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-sellergoods-service/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-manager-web/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  </**filesets**>  </**configuration**>  </**plugin**>  </**plugins**>   <**pluginManagement**>  <**plugins**>  *<!-- 配置Tomcat7插件 -->* <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**version**>2.2</**version**>  </**plugin**>  *<!-- 配置打包时跳过测试 -->* <**plugin**>  <**groupId**>org.apache.maven.plugins</**groupId**>  <**artifactId**>maven-surefire-plugin</**artifactId**>  <**version**>2.12.4</**version**>  </**plugin**>   </**plugins**>  </**pluginManagement**>  </**build**>      </**project**> |

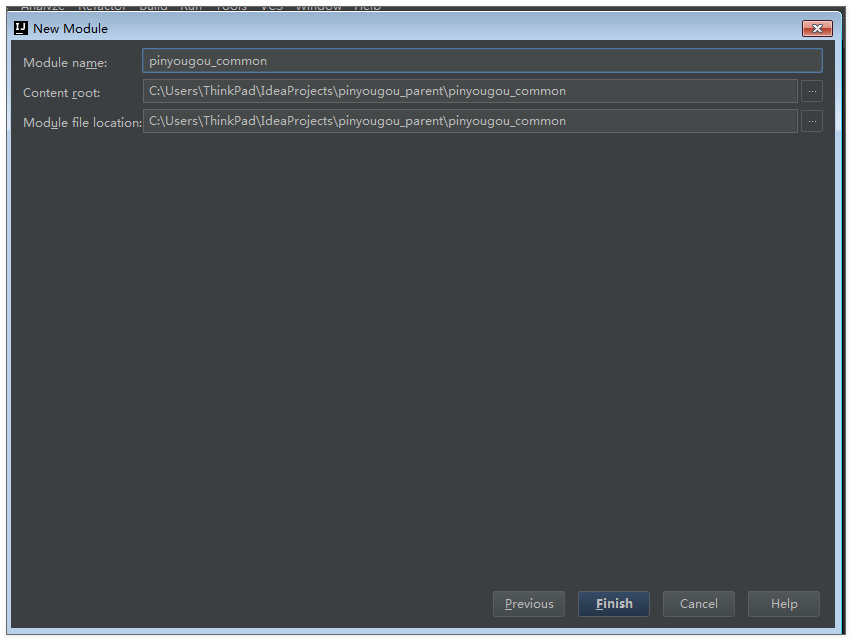
#### pinyougou-common

##### 搭建截图









##### pom.xml

|  |
| --- |
| <**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-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-common</**artifactId**>  <**packaging**>jar</**packaging**>   <**name**>pinyougou-common</**name**>  <**url**>http://maven.apache.org</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**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**>  *<!-- Jackson Json处理工具包 -->* <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**>  <**version**>${jackson.version}</**version**>  </**dependency**>  *<!--alibaba json处理工具包-->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>fastjson</**artifactId**>  <**version**>${fastjson.version}</**version**>  </**dependency**>  *<!-- httpclient -->* <**dependency**>  <**groupId**>org.apache.httpcomponents</**groupId**>  <**artifactId**>httpclient</**artifactId**>  <**version**>${httpclient.version}</**version**>  </**dependency**>  *<!-- 日志处理 -->* <**dependency**>  <**groupId**>org.slf4j</**groupId**>  <**artifactId**>slf4j-log4j12</**artifactId**>  <**version**>${slf4j.version}</**version**>  </**dependency**>  </**dependencies**> </**project**> |

#### pinyougou-pojo

##### 搭建截图

参考common搭建

##### pom.xml

|  |
| --- |
| <**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-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-pojo</**artifactId**>  <**packaging**>jar</**packaging**>   <**name**>pinyougou-pojo</**name**>  <**url**>http://maven.apache.org</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  </**dependencies**> </**project**> |

#### pinyougou-dao

##### 搭建截图

参考common

##### pom.xml

|  |
| --- |
| <**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-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-dao</**artifactId**>  <**packaging**>jar</**packaging**>   <**name**>pinyougou-dao</**name**>  <**url**>http://maven.apache.org</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-pojo</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  *<!-- Mybatis -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis-spring</**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**>  <**dependency**>  <**groupId**>com.github.pagehelper</**groupId**>  <**artifactId**>pagehelper</**artifactId**>  <**version**>${pagehelper.version}</**version**>  </**dependency**>  </**dependencies**>  <**build**>  <**resources**>  *<!--拷贝xml文件到classpath下-->* <**resource**>  <**directory**>src/main/java</**directory**>  <**includes**>  <**include**>\*\*/\*.properties</**include**>  <**include**>\*\*/\*.xml</**include**>  </**includes**>  <**filtering**>false</**filtering**>  </**resource**>  </**resources**>  </**build**> </**project**> |

#### pinyougou-sellergoods-interface

商家商品服务接口工程。打包方式jar

##### 搭建截图

参考common

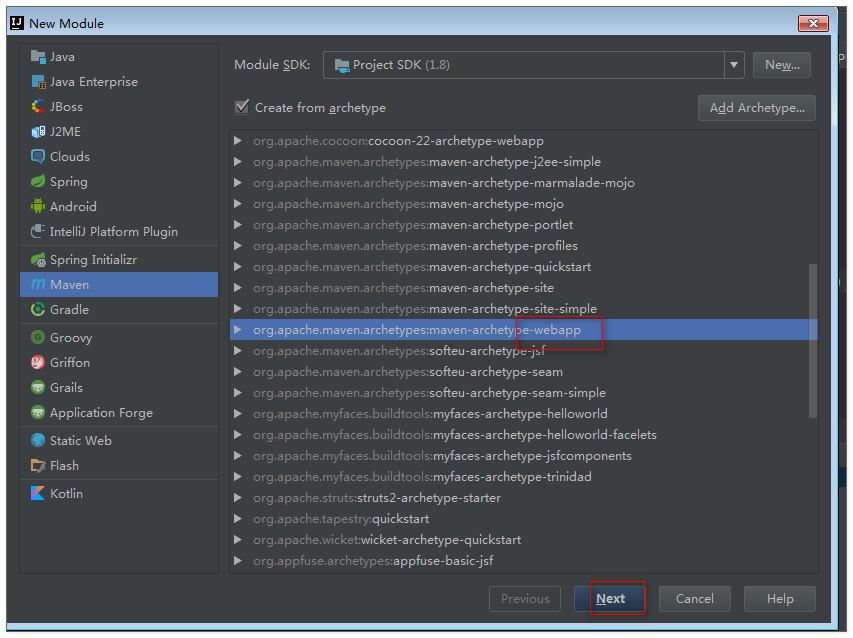
##### pom.xml

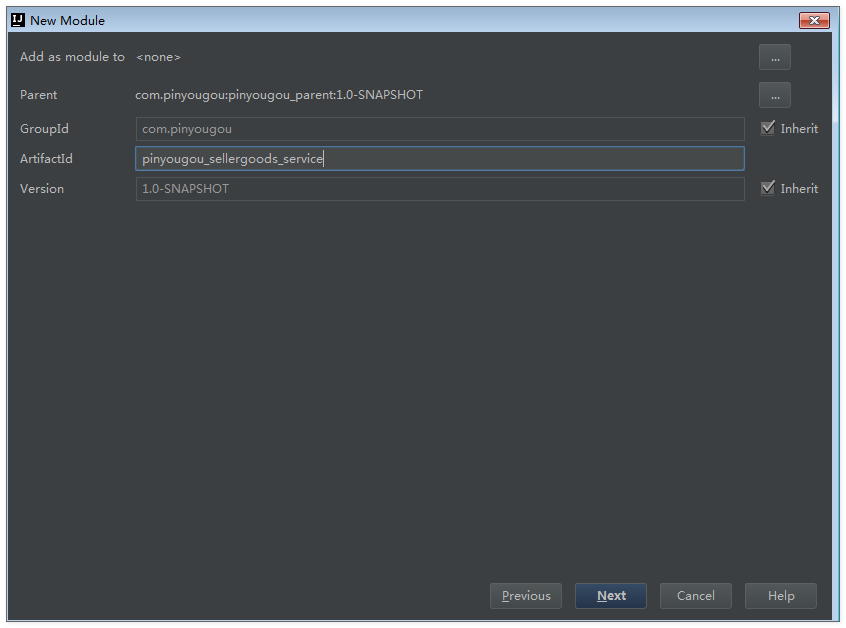
|  |
| --- |
| <**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**parent**>  <**artifactId**>pinyougou-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-sellergoods-interface</**artifactId**>  <**packaging**>jar</**packaging**>   <**name**>pinyougou-sellergoods-interface</**name**>  <**url**>http://maven.apache.org</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-common</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-pojo</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  </**dependencies**> </**project**> |

#### pinyougou-sellergoods-service

商家商品服务实现类工程

##### 搭建截图





下一步点击 finish

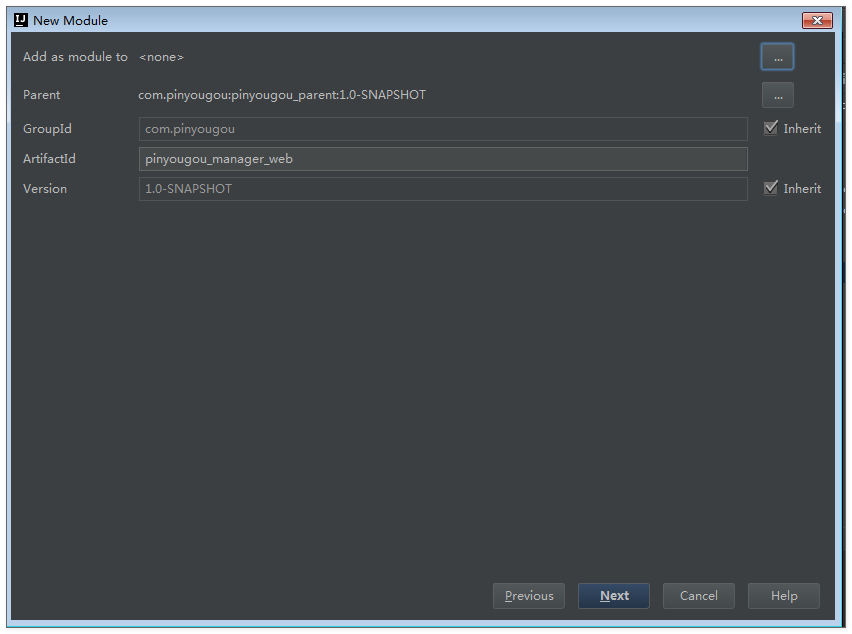
##### pom.xml

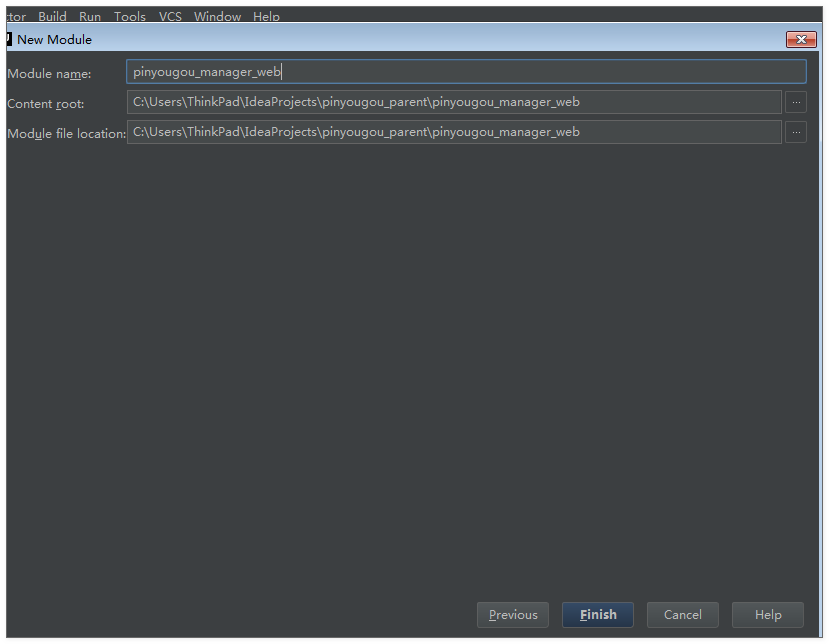
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**parent**>  <**artifactId**>pinyougou-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-sellergoods-service</**artifactId**>  <**packaging**>war</**packaging**>   <**name**>pinyougou-sellergoods-service Maven Webapp</**name**>  *<!--* ***FIXME change it to the project's website*** *-->* <**url**>http://www.example.com</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**version**>4.12</**version**>  <**scope**>test</**scope**>  </**dependency**>  *<!--通用类-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-common</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!--实现接口-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-sellergoods-interface</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!--调用dao-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-dao</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!-- 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**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-test</**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**>   </**dependencies**>   <**build**>  <**finalName**>pinyougou-sellergoods-service</**finalName**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  <**path**>/</**path**>  <**port**>9001</**port**>  </**configuration**>  </**plugin**>  </**plugins**>  </**build**> </**project**> |

#### pinyougou-manager-web

运营商管理后台工程，表现层工程，打包方式为war包。

##### 搭建截图





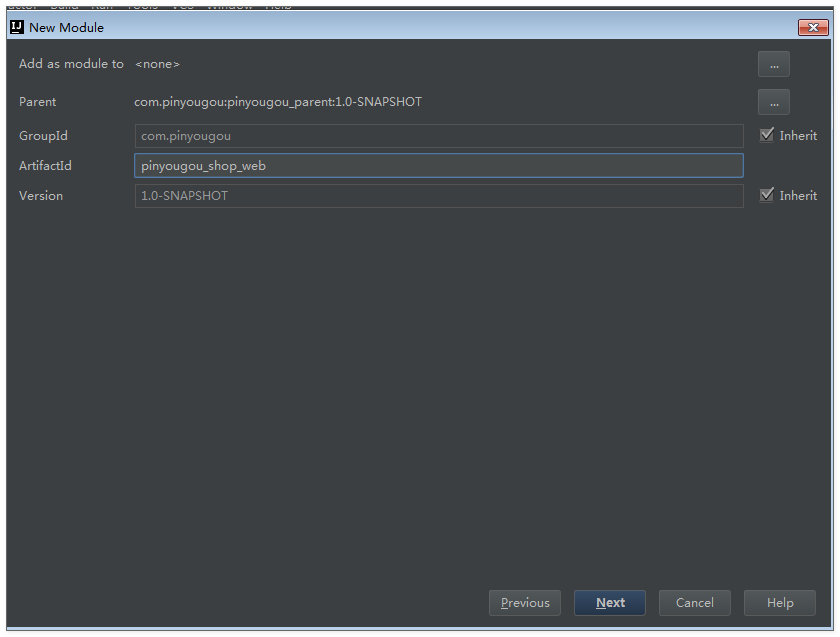
##### pom.xml

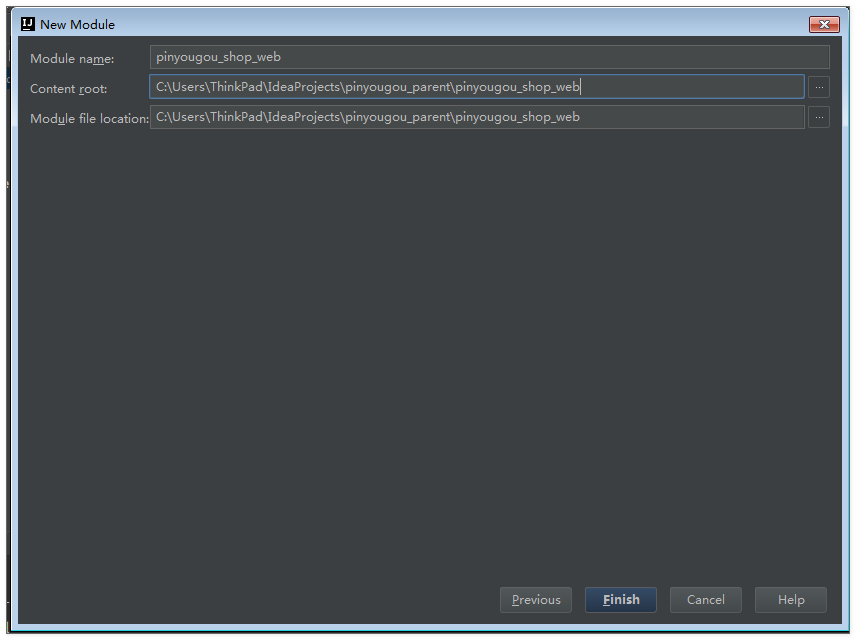
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**parent**>  <**artifactId**>pinyougou-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>   <**artifactId**>pinyougou-manager-web</**artifactId**>  <**packaging**>war</**packaging**>   <**name**>pinyougou-manager-web Maven Webapp</**name**>  *<!--* ***FIXME change it to the project's website*** *-->* <**url**>http://www.example.com</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**version**>4.12</**version**>  <**scope**>test</**scope**>  </**dependency**>   *<!--通用类-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-common</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!--商家商品服务-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-sellergoods-interface</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!-- 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**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-test</**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**>junit</**groupId**>  <**artifactId**>junit</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>servlet-api</**artifactId**>  <**scope**>provided</**scope**>  </**dependency**>   </**dependencies**>   <**build**>  <**finalName**>pinyougou-manager-web</**finalName**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  <**path**>/</**path**>  <**port**>9101</**port**>  </**configuration**>  </**plugin**>   </**plugins**>   </**build**> </**project**> |

#### pinyougou-shop-web

商家管理后台，打包方式war 。表现层工程

##### 搭建截图





##### pom.xml

|  |
| --- |
| <**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-parent</**artifactId**>  <**groupId**>com.pinyougou</**groupId**>  <**version**>1.0-SNAPSHOT</**version**>  </**parent**>  <**modelVersion**>4.0.0</**modelVersion**>  <**artifactId**>pinyougou-shop-web</**artifactId**>  <**packaging**>war</**packaging**>  <**name**>pinyougou-shop-web Maven Webapp</**name**>  <**url**>http://maven.apache.org</**url**>  <**dependencies**>  *<!--common通用的模块-->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-common</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**> </**dependency**>  *<!--需要依赖于接口 -->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-sellergoods-interface</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  </**dependency**>  *<!-- 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**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-test</**artifactId**>  </**dependency**>  *<!-- dubbo相关 -->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>dubbo</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>javassist</**groupId**>  <**artifactId**>javassist</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.apache.zookeeper</**groupId**>  <**artifactId**>zookeeper</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>com.github.sgroschupf</**groupId**>  <**artifactId**>zkclient</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>servlet-api</**artifactId**>  <**scope**>provided</**scope**>  </**dependency**>   </**dependencies**>  <**build**>  <**finalName**>pinyougou-shop-web</**finalName**>  <**plugins**>  <**plugin**>  <**groupId**>org.apache.tomcat.maven</**groupId**>  <**artifactId**>tomcat7-maven-plugin</**artifactId**>  <**configuration**>  <**path**>/</**path**>  <**port**>9102</**port**>  </**configuration**>  </**plugin**>  </**plugins**>  </**build**> </**project**> |

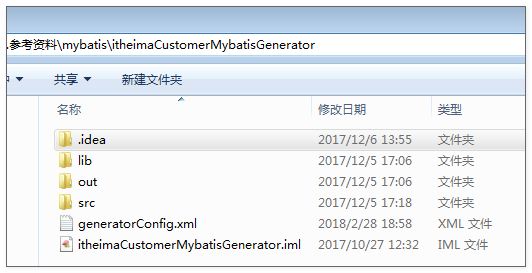
#### 配置清理插件(可选)

当配置完成之后，每次mvn install都会安装很多的jar和相关的配置文件target中，文件还很大。如果不想看到，可以在执行开发之后，清除掉。配置如下：

|  |
| --- |
| *<!--清理插件 用于清理target中的所有的内容，包括parent以及parent下的所有的子项目中的target-->* <**plugin**>  <**groupId**>org.apache.maven.plugins</**groupId**>  <**artifactId**>maven-clean-plugin</**artifactId**>  <**version**>2.6.1</**version**>  <**configuration**>  <**filesets**>  <**fileset**>  <**directory**>${basedir}/pinyougou-common/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-pojo/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-dao/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-sellergoods-interface/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-sellergoods-service/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-sellergoods-interface/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  <**fileset**>  <**directory**>${basedir}/pinyougou-manager-web/target</**directory**>  <**followSymlinks**>false</**followSymlinks**>  </**fileset**>  </**filesets**>  </**configuration**> </**plugin**> |

### 生成POJO 和Mapper

使用逆向工程生成通用POJO并实现序列化，生成Mapper接口

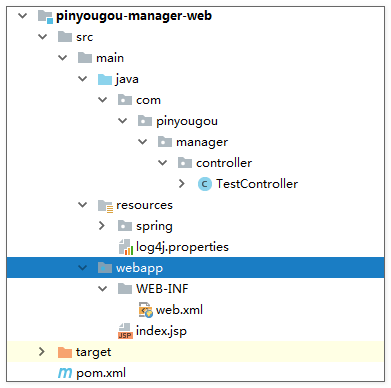


使用idea-->new project from exsits source-->create-->下一步。。。即可。

### 整合SSM

#### 整合manager-web

##### 整合结构如下：



##### springmvc.xml

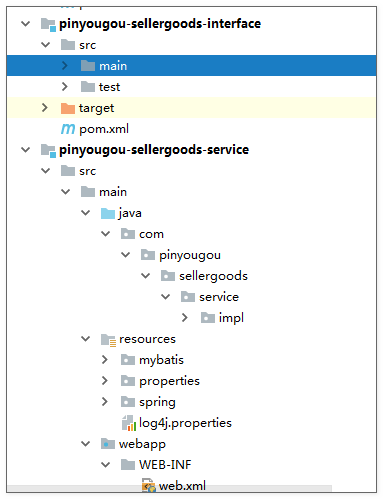
|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"**>   *<!--配置应用名 服务消费方-->* <**dubbo:application name="pinyougou-manager-web"**></**dubbo:application**>  *<!--配置注册中心-->* <**dubbo:registry protocol="zookeeper" address="192.168.25.133" port="2181"**></**dubbo:registry**>  *<!--引用服务 通过注解 @reference 扫描 即可引用 包含了组件扫描-->* <**dubbo:annotation package="com.pinyougou.manager.controller"**></**dubbo:annotation**>  *<!--设置消费者的调用超时时间-->* <**dubbo:consumer timeout="100000"**></**dubbo:consumer**>   *<!--组件扫描扫描带有@controller的注解-->* <**context:component-scan base-package="com.pinyougou.manager.controller"**></**context:component-scan**>   *<!--注解驱动并设置统一日期的输出格式-->* <**mvc:annotation-driven**>  <**mvc:message-converters register-defaults="true"**>  <**bean class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter"**>  <**property name="supportedMediaTypes" value="application/json"**/>  <**property name="features"**>  <**array**>  *<!--返回结果是包含null值 默认是过滤掉-->* <**value**>WriteMapNullValue</**value**>  *<!--修改全局日期格式 默认是false 表示不修改，自己定义 配置在此处表示要统一日期格式输出 ，默认格式为：yyyy-MM-dd -->  <!--JSON.DEFFAULT\_DATE\_FORMAT-->* <**value**>WriteDateUseDateFormat</**value**>  </**array**>  </**property**>  </**bean**>  </**mvc:message-converters**>  </**mvc:annotation-driven**> </**beans**> |

##### web.xml

|  |
| --- |
| *<?***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"  id="WebApp\_ID" version="2.5"**>  <**welcome-file-list**>  <**welcome-file**>index.jsp</**welcome-file**>  </**welcome-file-list**>   <**servlet**>  <**servlet-name**>springmvc</**servlet-name**>  <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**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**>  <**filter**>  <**filter-name**>characterEncoding</**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**>  </**filter**>  <**filter-mapping**>  <**filter-name**>characterEncoding</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   </**web-app**> |

#### 整合服务层（interface及service）

##### 整合结构图：



##### SqlMapConfig.xml:

|  |
| --- |
| *<?***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**> <**plugins**>  *<!-- com.github.pagehelper为PageHelper类所在包名 -->* <**plugin interceptor="com.github.pagehelper.PageHelper"**>  *<!-- 设置数据库类型 Oracle,Mysql,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库-->* <**property name="dialect" value="mysql"**/>  </**plugin**> </**plugins**>  </**configuration**> |

##### db.properties:

|  |
| --- |
| **jdbc.driver**=**com.mysql.jdbc.Driver jdbc.url**=**jdbc:mysql://localhost:3306/pinyougoudb?characterEncoding=utf-8 jdbc.username**=**root jdbc.password**=**itcast** |

##### applicationContext-dao.xml:

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  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"**>    *<!-- 配置数据源 配置sqlsessionfactory mapper扫描器 -->* <**context:property-placeholder location="classpath:properties/\*.properties"**/>    <**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**>    <**bean class="org.mybatis.spring.SqlSessionFactoryBean" id="sqlsessionFactory"**>  <**property name="dataSource" ref="dataSource"**></**property**>  <**property name="configLocation" value="classpath:mybatis/SqlMapConfig.xml"**></**property**>  </**bean**>    <**bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>  <**property name="basePackage" value="com.pinyougou.mapper"**></**property**>  </**bean**>  </**beans**> |

##### applicationContext-service.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: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"**>  *<!--配置服务提供者应用名称-->* <**dubbo:application name="pinyougou-sellergoods"**></**dubbo:application**>  *<!--配置注册中心-->* <**dubbo:registry protocol="zookeeper" address="192.168.25.133" port="2181"**></**dubbo:registry**>   *<!--配置服务调用的端口 默认端口为20880-->* <**dubbo:protocol name="dubbo" port="20880"**></**dubbo:protocol**>   *<!--发布服务:注解的方式 扫描对应的包 @service 即可发布服务-->* <**dubbo:annotation package="com.pinyougou.sellergoods.service"**></**dubbo:annotation**> </**beans**> |

##### applicationContext-trans.xml:

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:aop="http://www.springframework.org/schema/aop" xmlns:tx="http://www.springframework.org/schema/tx"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd   http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd"**>    *<!-- 配置事务管理器 -->* <**bean class="org.springframework.jdbc.datasource.DataSourceTransactionManager" id="transactionManager"**>  <**property name="dataSource" ref="dataSource"**></**property**>  </**bean**>  *<!-- 通知 -->* <**tx:advice transaction-manager="transactionManager" id="txAdvice"**>  <**tx:attributes**>  <**tx:method name="save\*" propagation="REQUIRED"**/>  <**tx:method name="add\*" propagation="REQUIRED"**/>  <**tx:method name="create\*" propagation="REQUIRED"**/>  <**tx:method name="update\*" propagation="REQUIRED"**/>  <**tx:method name="edit\*" propagation="REQUIRED"**/>  <**tx:method name="delete\*" propagation="REQUIRED"**/>  <**tx:method name="remove\*" propagation="REQUIRED"**/>  <**tx:method name="find\*" propagation="SUPPORTS" read-only="true"**/>  <**tx:method name="get\*" propagation="SUPPORTS" read-only="true"**/>  </**tx:attributes**>  </**tx:advice**>  *<!-- 切面 -->* <**aop:config**>  <**aop:advisor advice-ref="txAdvice" pointcut="execution(\* com.pinyougou.sellergoods.service.\*.\*(..))"**/>  </**aop:config**>   <**import resource="applicationContext-dao.xml"**></**import**>  <**import resource="applicationContext-service.xml"**></**import**>  </**beans**> |

##### web.xml:

|  |
| --- |
| *<?***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"  id="WebApp\_ID" version="2.5"**>  <**welcome-file-list**>  <**welcome-file**>index.jsp</**welcome-file**>  </**welcome-file-list**>   <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:spring/applicationContext-trans.xml</**param-value**>  </**context-param**>  <**listener**>  <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>   </**web-app**> |

## 品牌管理

### 需求分析

在运营商模块中，运营商可以添加品牌，查询，更新，删除等功能。

暂时实现一个功能就是：查询品牌列表 以JSON格式展示在页面中。

表现层接受请求，查询所有品牌列表数据 返回给页面JSON即可。

请求的URL:

/brand/getBrandList

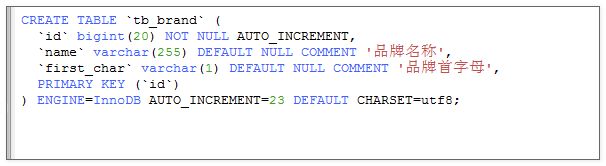
参数：

无

返回值：

json

### 数据库表分析



### 服务层开发（服务提供方）

dao可以使用逆向工程，在pinyougou-sellergoods-interface中实现接口开发，在pinyougou-sellergoods-service中实现实现类开发。

#### dao

使用逆向工程

#### 开发service接口

在pinyougou-sellergoods-interface工程下开发：

|  |
| --- |
| **public interface** BrandService {  **public** List<TbBrand> findAll(); } |

#### 开发service实现类

在pinyougou-sellergoods-ervice工程下开发：

|  |
| --- |
| **package** com.pinyougou.sellergoods.service.impl;  **import** com.alibaba.dubbo.config.annotation.Service; **import** com.pinyougou.mapper.TbBrandMapper; **import** com.pinyougou.pojo.TbBrand; **import** com.pinyougou.sellergoods.service.BrandService; **import** org.springframework.beans.factory.annotation.Autowired;  **import** java.util.List;  /\*\*  \* **@author** ljh  \* **@version** 1.0  \* **@description** 描述  \* **@title** 标题  \* **@package** com.pinyougou.sellergoods.service.impl  \* **@company** www.itheima.com  \*/ @Service **public class** BrandServiceImpl **implements** BrandService {  @Autowired  **private** TbBrandMapper **tbBrandMapper**;  @Override  **public** List<TbBrand> findAll() {  **return tbBrandMapper**.selectByExample(**null**);  } } |

### 表现层开发（服务消费方）

#### controller

|  |
| --- |
| **package** com.pinyougou.manager.controller;  **import** com.alibaba.dubbo.config.annotation.Reference; **import** com.pinyougou.pojo.TbBrand; **import** com.pinyougou.sellergoods.service.BrandService; **import** org.springframework.beans.factory.annotation.Autowired; **import** org.springframework.stereotype.Controller; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.ResponseBody;  **import** java.util.List;  /\*\*  \* **@author** ljh  \* **@version** 1.0  \* **@description** 描述  \* **@title** 标题  \* **@package** com.pinyougou.manager.controller  \* **@company** www.itheima.com  \*/ @Controller @RequestMapping(**"/brand"**) **public class** BrandController {   @Reference  **private** BrandService **brandService**;   @RequestMapping(**"/findAll"**)  @ResponseBody  **public** List<TbBrand> findAll(){  **return brandService**.findAll();  } } |

### 测试

启动zookeeper ,启动服务层，启动表现层 。

输入地址：http://localhost:9101/brand/findAll.do

