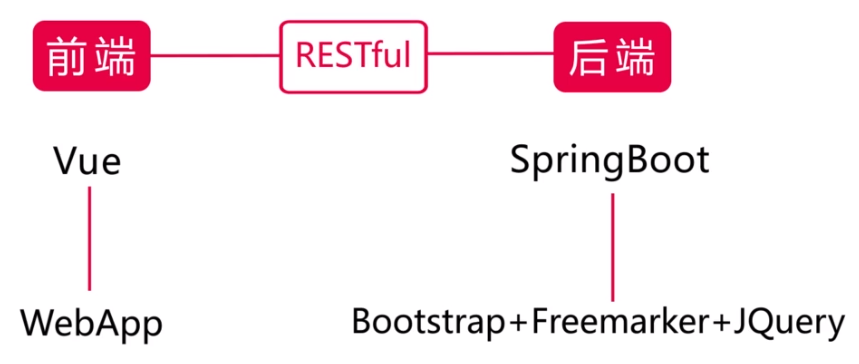
**Spring Boot 微信点餐系统**

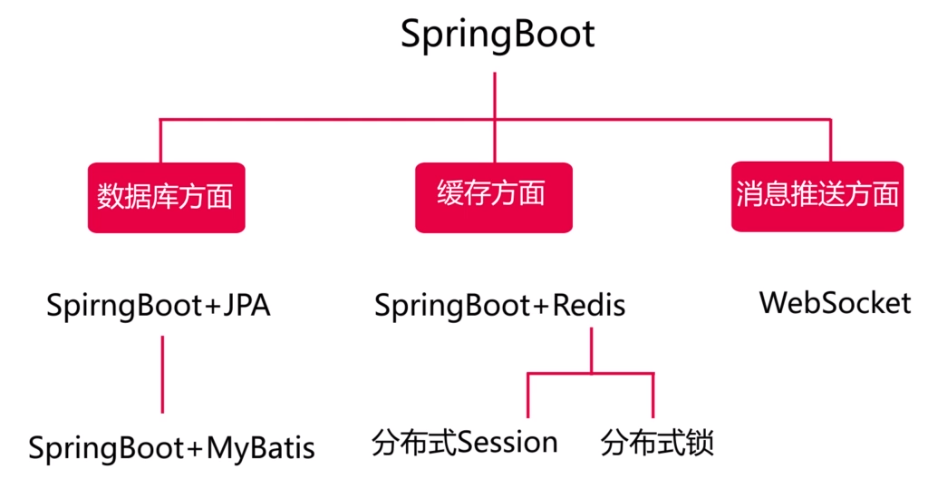
1. 项目技术

源码地址：https://gitlab-demo.com/

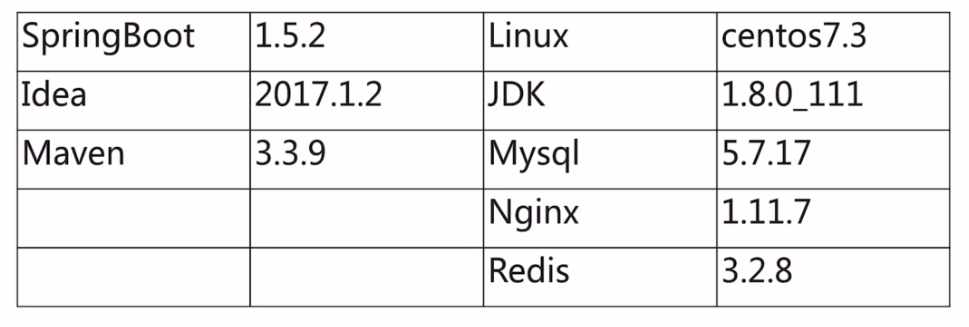
1.1.前后端分离



1.2.数据库方面



1.2.软件版本



1. 项目设计

2.1.角色划分

2.1.1.买家（手机端）

微信公众账号提供的服务

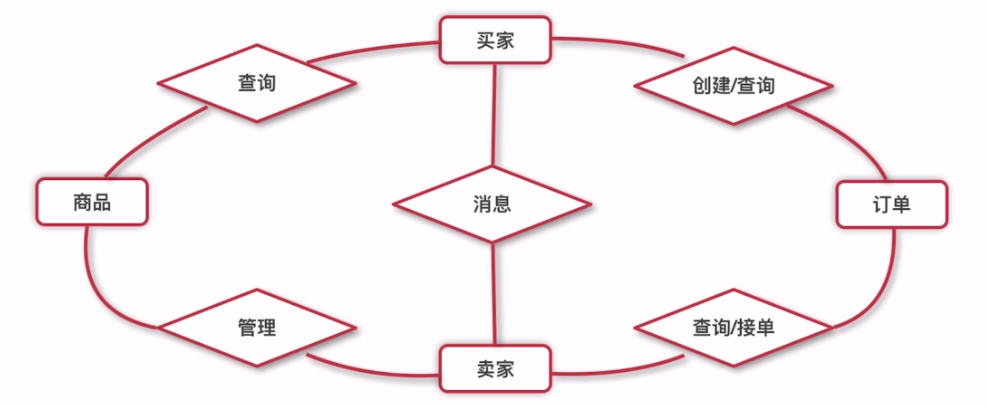
2.1.2.卖家（PC端）

通过PC端去管理他的业务

2.2.功能模块划分



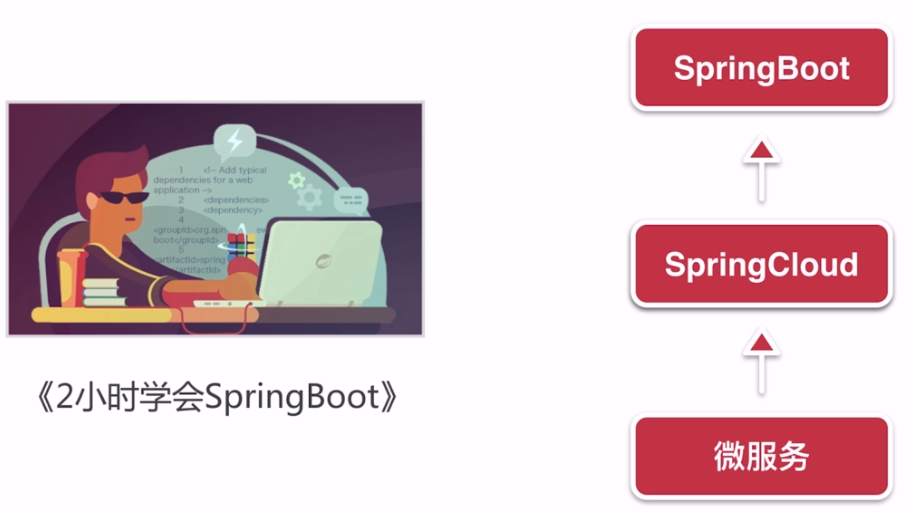
2.2.1.关系图

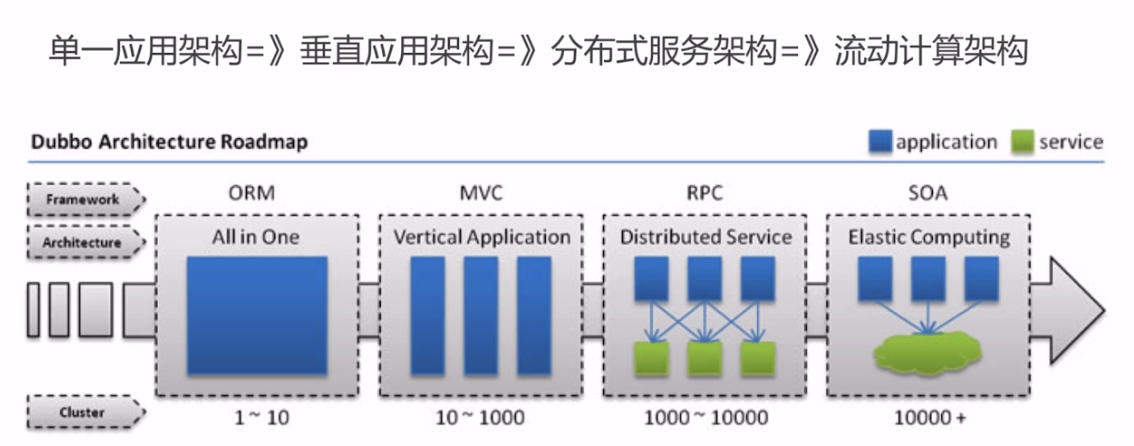


2.3.部署架构



2.3.1.架构和基础框架

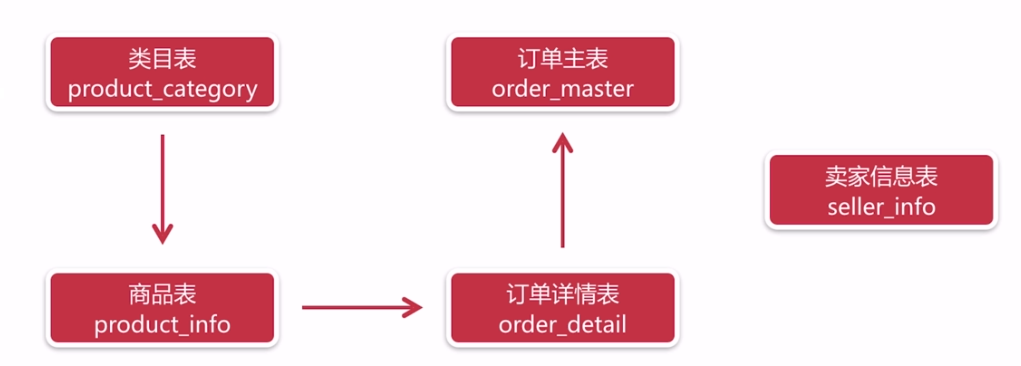


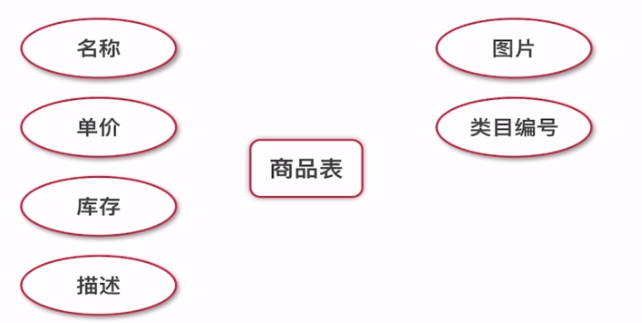




2.4.数据库设计

2.4.1.表与表之间的关系











2.4.2.建表sql

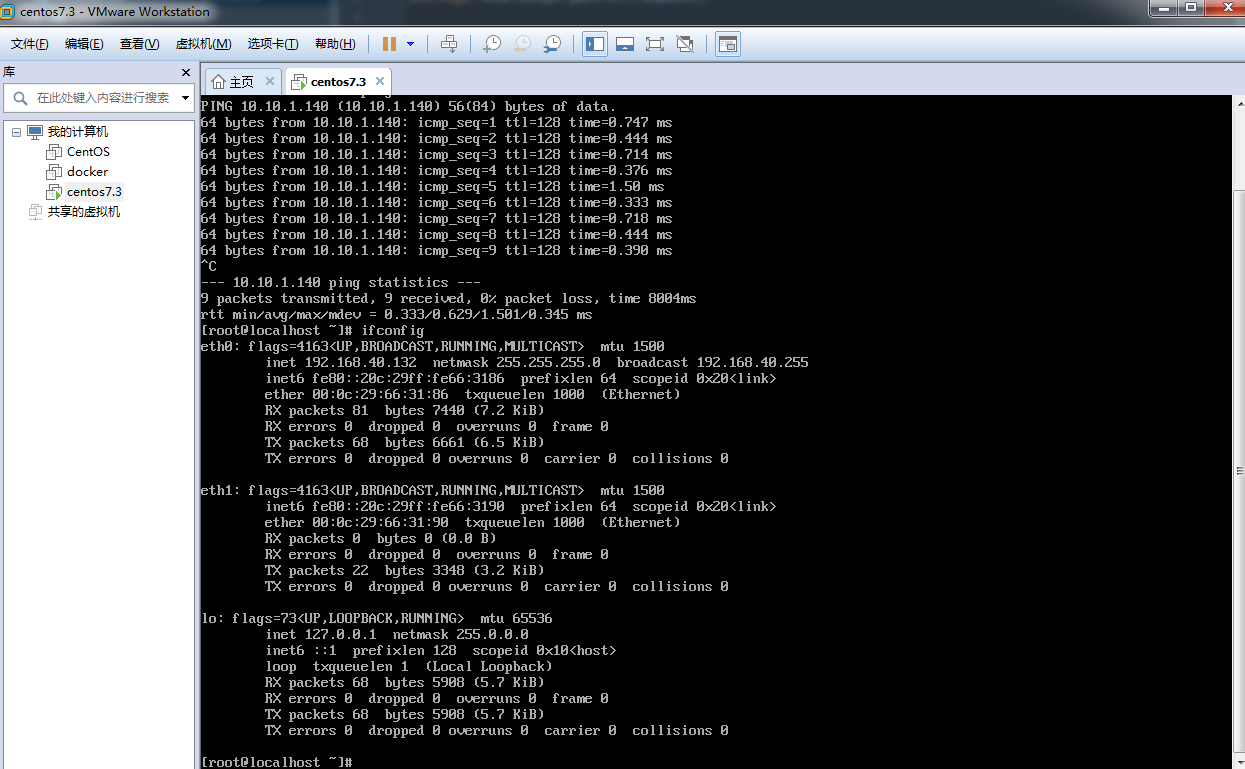
|  |  |
| --- | --- |
| 商品表 | create table 'product\_info' {  'product\_id' varchar(32) not null,  'product\_name' varchar(64) not null comment '商品名称',  'product\_price' decimal(8,2) not null comment '商品单价',  'product\_stock' int not null comment '商品库存',  'product\_description' varchar(64) comment '商品描述',  'product\_icon' varchar(512) comment '商品小图',  'category\_type' int not null comment '类目编号',  'create\_time' timestamp not null default current\_timestamp comment '创建时间',  'update\_time' timestamp not null default current\_timestamp on update current\_timestamp comment '更新时间',  primary key('product\_id')  } comment '商品表'; |
| 类目表 | create table 'product\_category' {  'category\_id' int not null auto\_increment,  'category\_name' varchar(64) not null comment '类目名称',  'category\_type' int not null comment '类目编号',  'create\_time' timestamp not null default current\_timestamp comment '创建时间',  'update\_time' timestamp not null default current\_timestamp on update current\_timestamp comment '更新时间',  primary key('category\_id'),  unique key 'uqe\_category\_type' ('category\_type')  } comment '类目表'; |
| 订单主表 | create table 'order\_master' {  'order\_id' varchar(32) not null,  'buyer\_name' varchar(32) not null comment '买家名字',  'buyer\_phone' varchar(32) not null comment '买家电话',  'buyer\_address' varchar(128) not null comment '买家地址',  'buyer\_openid' varchar(64) not null comment '买家微信openid',  'order\_amount' decimal(8,2) not null comment '订单总金额',  'order\_status' tinyint(3) not null default '0' comment '订单状态,默认0新下单',  'pay\_status' tinyint(3) not null default '0' comment '支付状态,默认0未支付',  'create\_time' timestamp not null default current\_timestamp comment '创建时间',  'update\_time' timestamp not null default current\_timestamp on update current\_timestamp comment '更新时间',  primary key('order\_id'),  key 'idx\_buyer\_openid' ('buyer\_openid')  } comment '订单主表'; |
| 订单详情表 | create table 'order\_detail' {  'detail\_id' varchar(32) not null,  'order\_id' varchar(32) not null,  'product\_id' varchar(32) not null,  'product\_name' varchar(64) not null comment '商品名称',  'product\_price' decimal(8,2) not null comment '商品单价',  'product\_quantity' int not null comment '商品数量',  'product\_icon' varchar(512) comment '商品小图',  'create\_time' timestamp not null default current\_timestamp comment '创建时间',  'update\_time' timestamp not null default current\_timestamp on update current\_timestamp comment '更新时间',  primary key('detail\_id'),  key 'idx\_order\_id' ('order\_id')  } comment '订单详情表'; |

2.4.3.注意事项

1. 开发环境搭建

3.1.虚拟机的使用

Virtualbox 虚拟机账号： root 密码：123456



Jdk 1.8

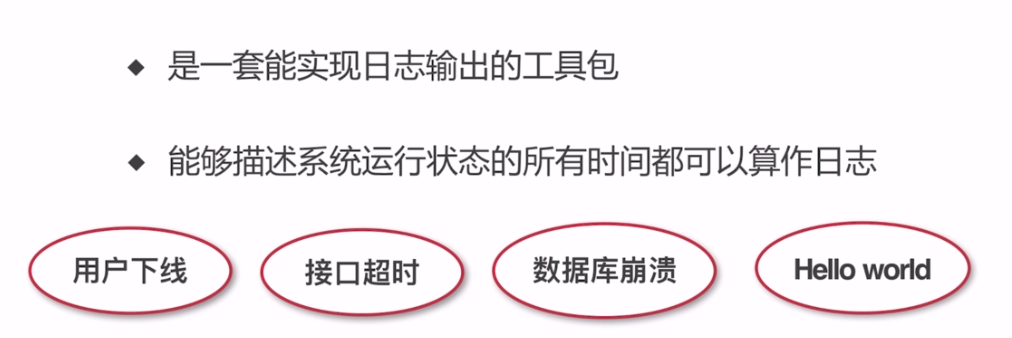
Maven 3.3.9

Idea 编辑器 下载有所有功能的那个

创建Spring boot项目

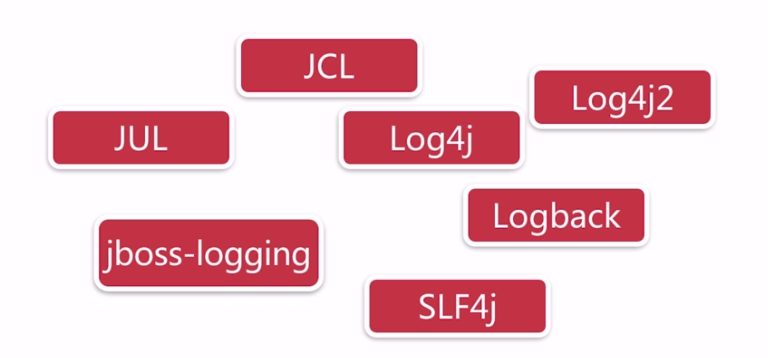
1. 日志

4.1.什么是日志框架



**4.1.1日志框架的能力：**

常见的日志框架：





4.2.日志框架的选择

Ceki写了3个日志框架： SLF4J Log4j Logback

JUL: 实现太过简陋，受到大家的吐槽。

Jboss-logging：实现之初就不是为了大众化服务的。

Log4j：Logback是这个的升级版

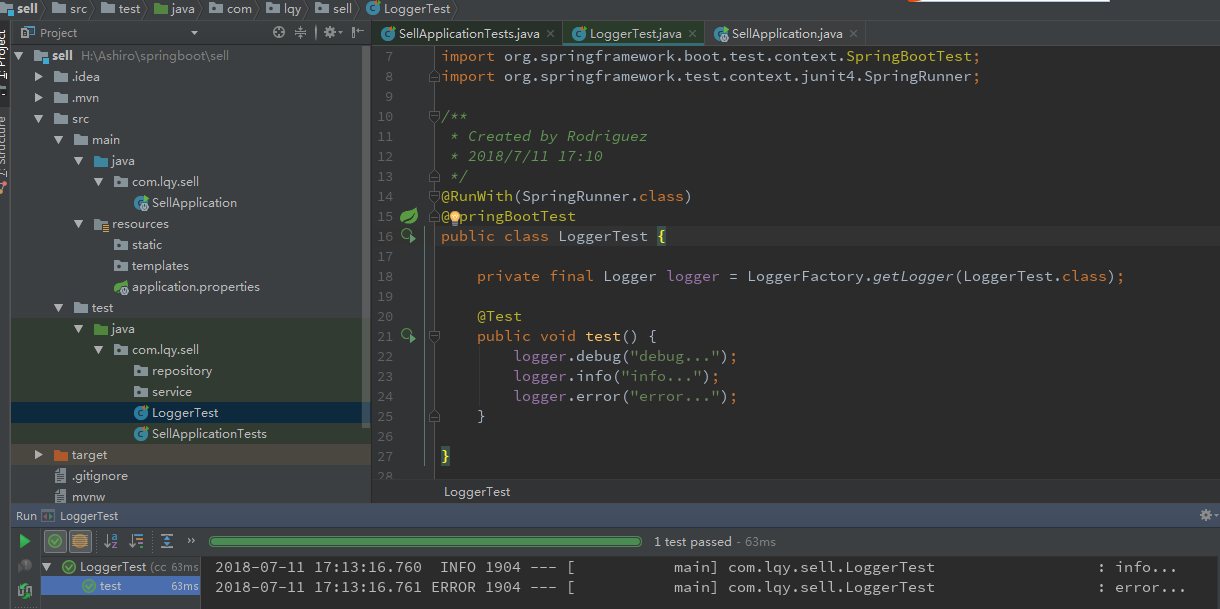
Log4j2：设计上很优秀，太先进了，很多开源框架对其支持有限。优于Logback的性能

JCL: SLF4J和Logback是同一个作者写的，所以划掉这个。

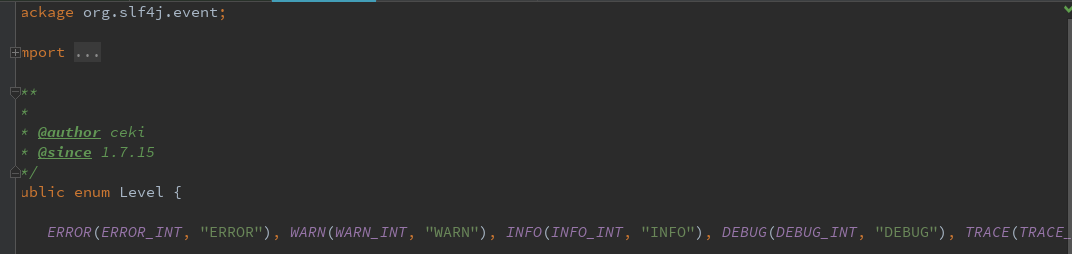
选择使用： SLF4J和Logback

4.3.Logback的使用与配置

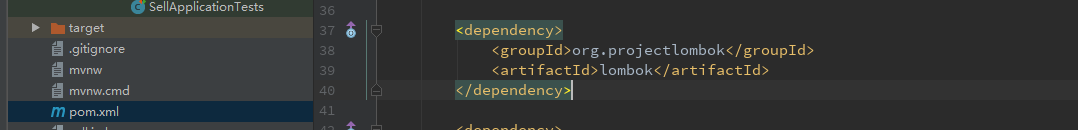
日志的使用：



系统默认的日志级别就是INFO级别

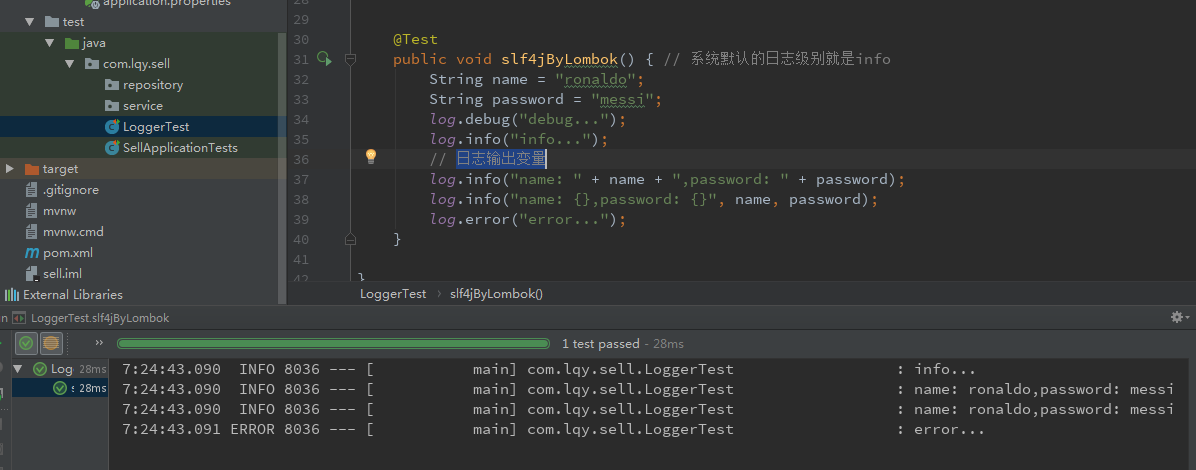


Lombok 简化日志使用打印使用





日志输出变量



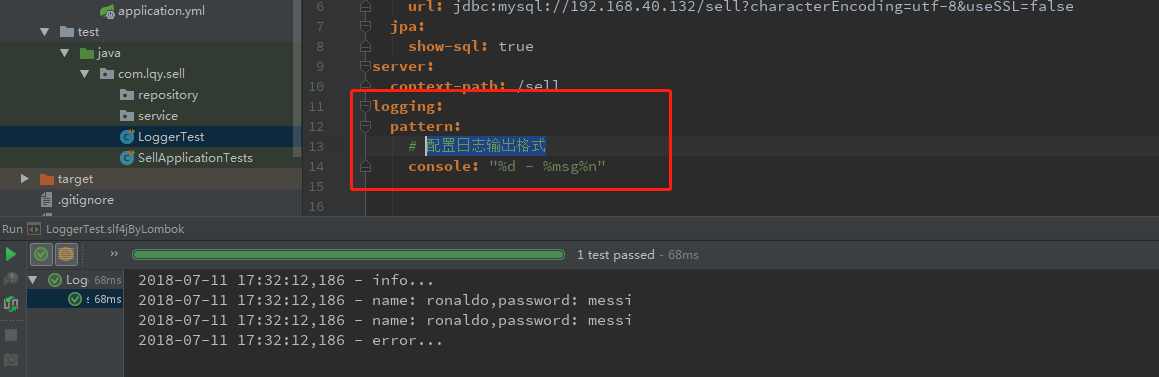
Logback日志的配置:

application.yml: 日志文件路径、日志文件格式等简单的配置

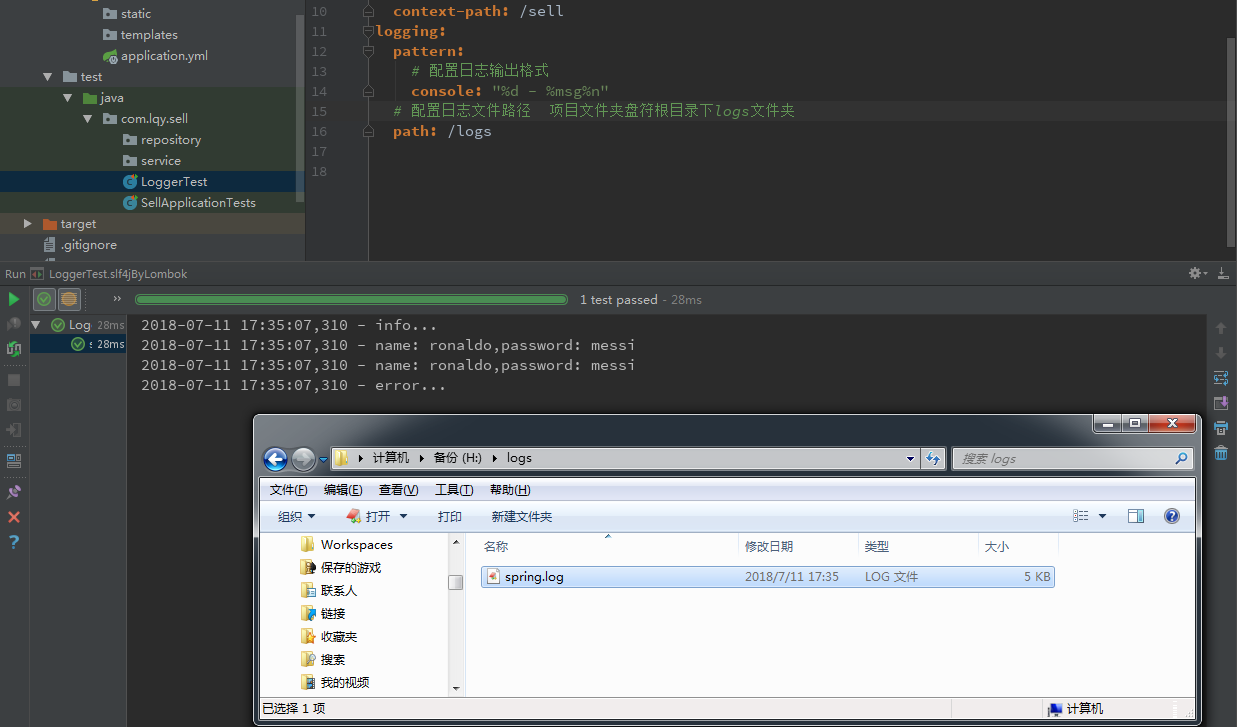
logback-spring.xml: 可以进行复杂的配置

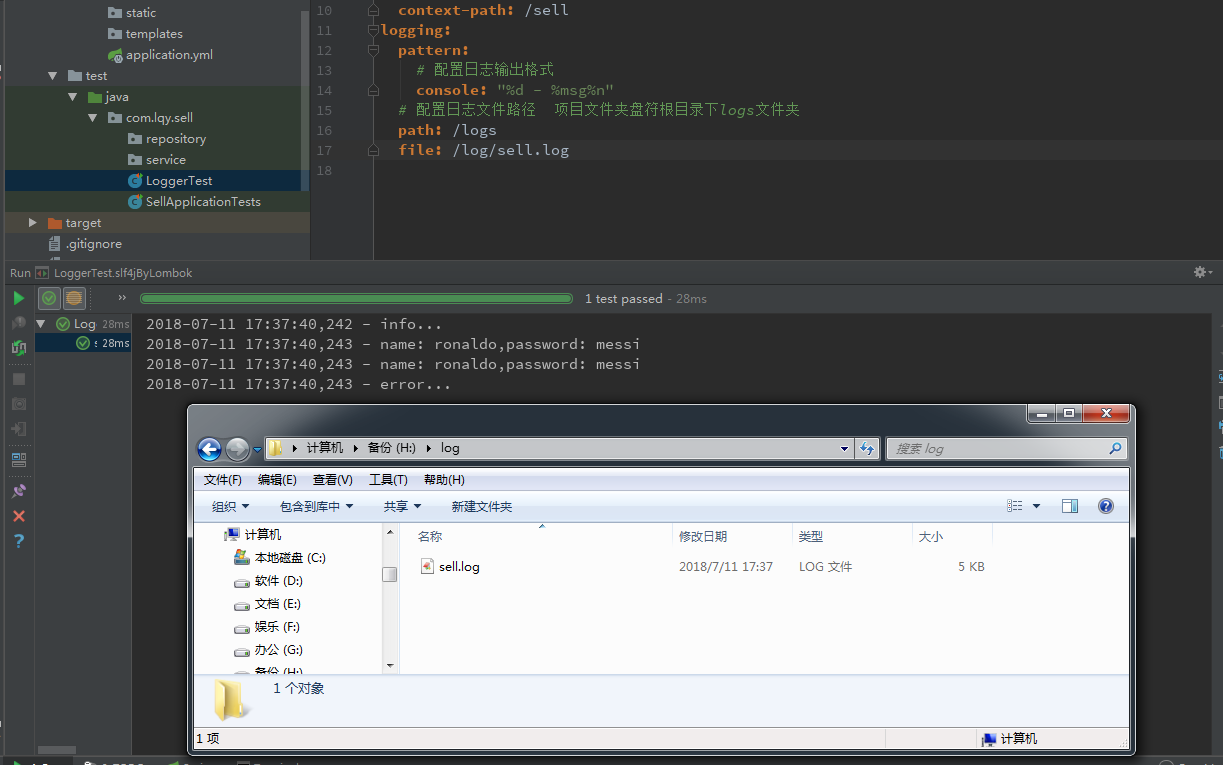
比如区分error和info日志、每天产生一个日志文件

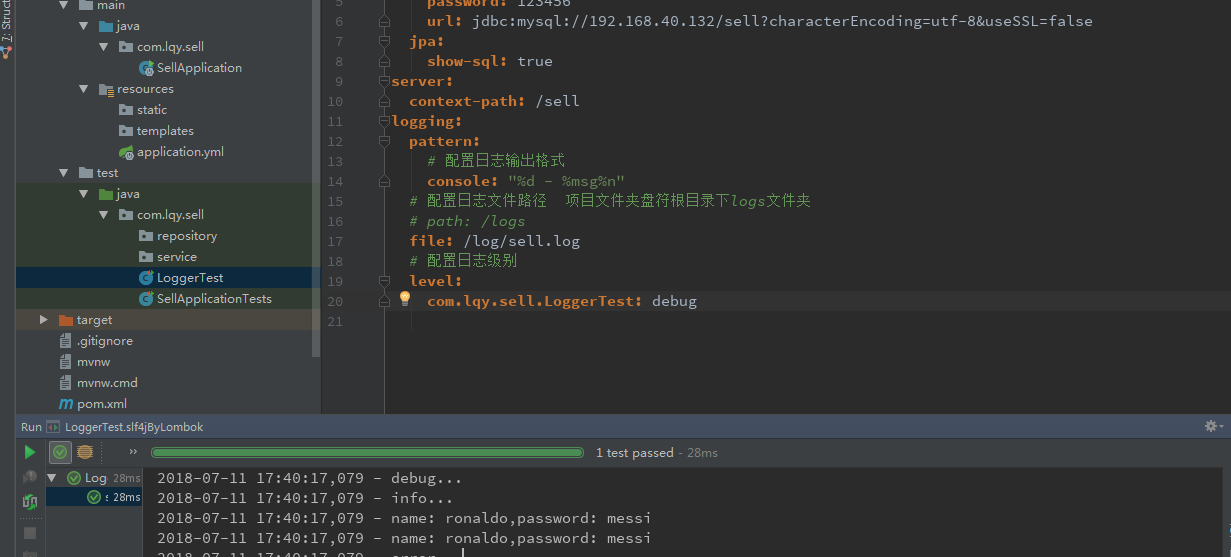
配置日志输出格式



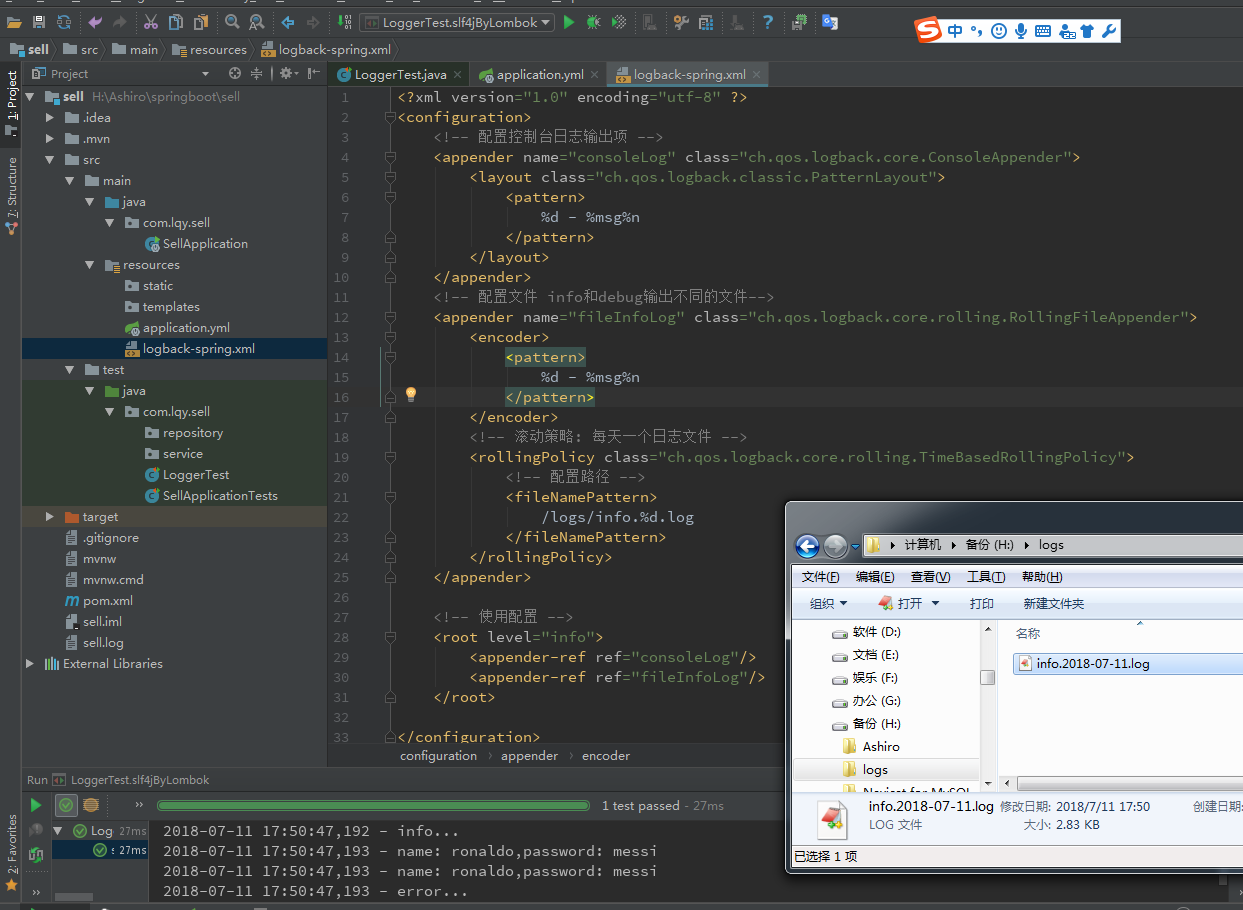
配置日志文件路径

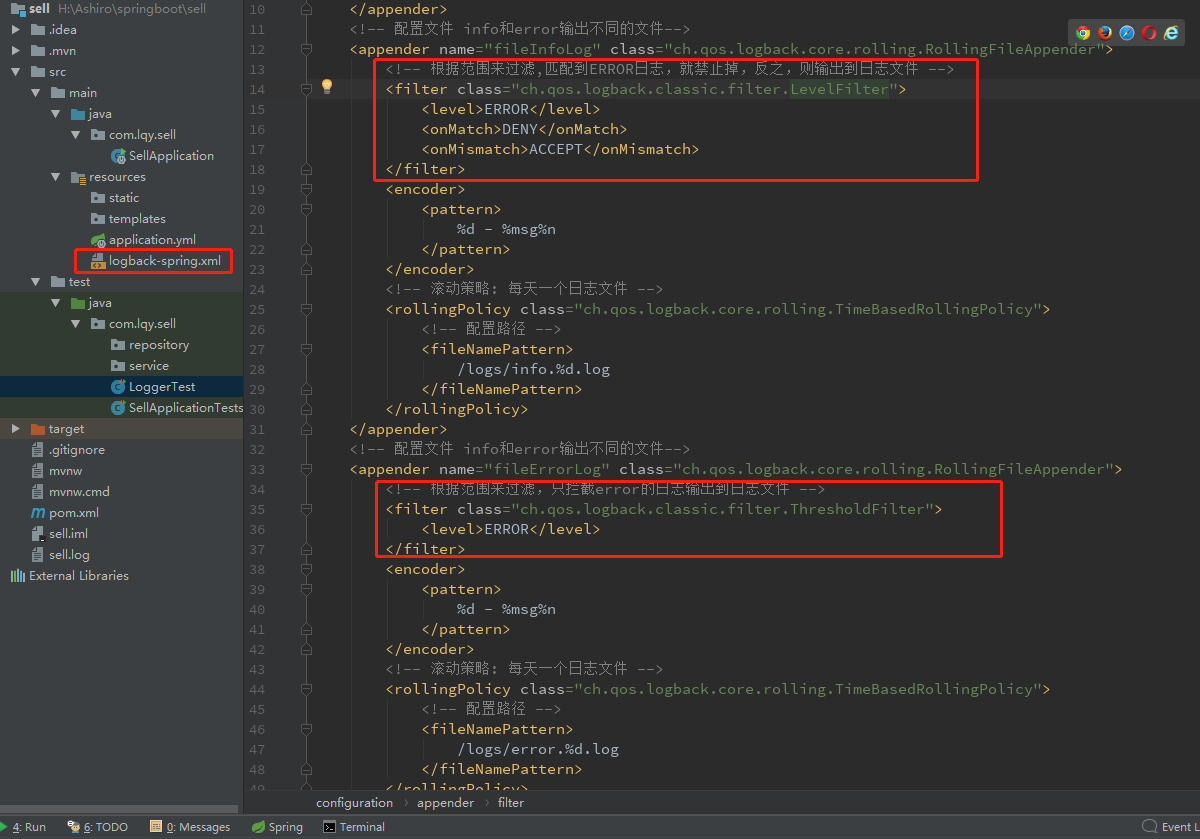


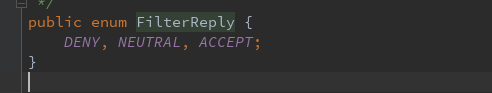




使用logback-spring.xml配置日志：

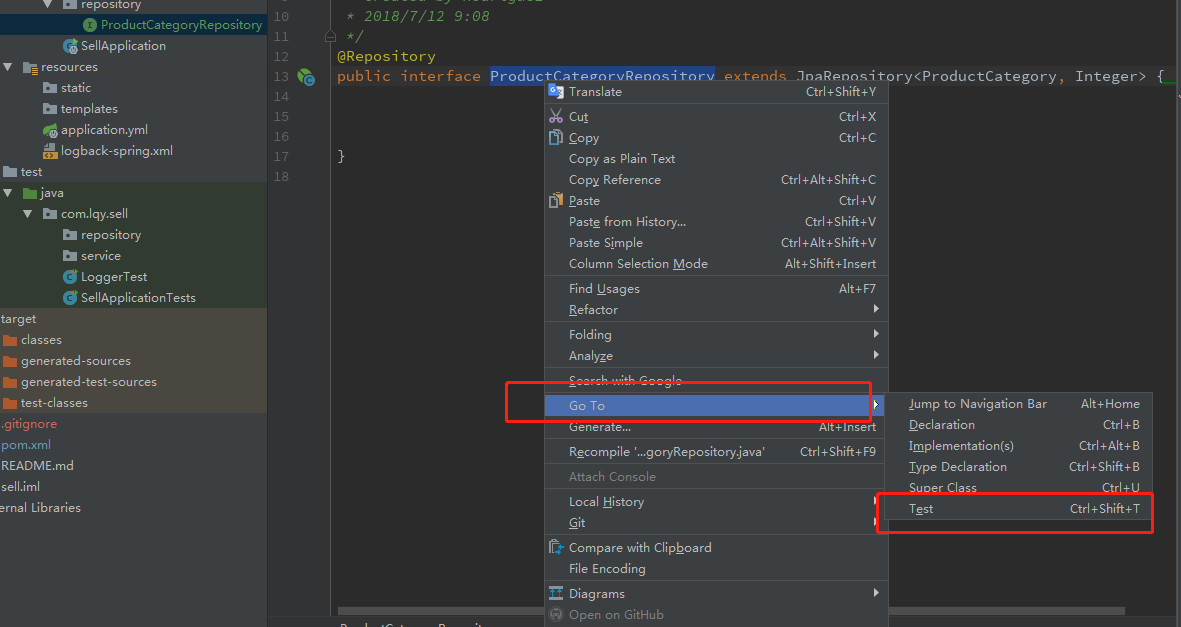


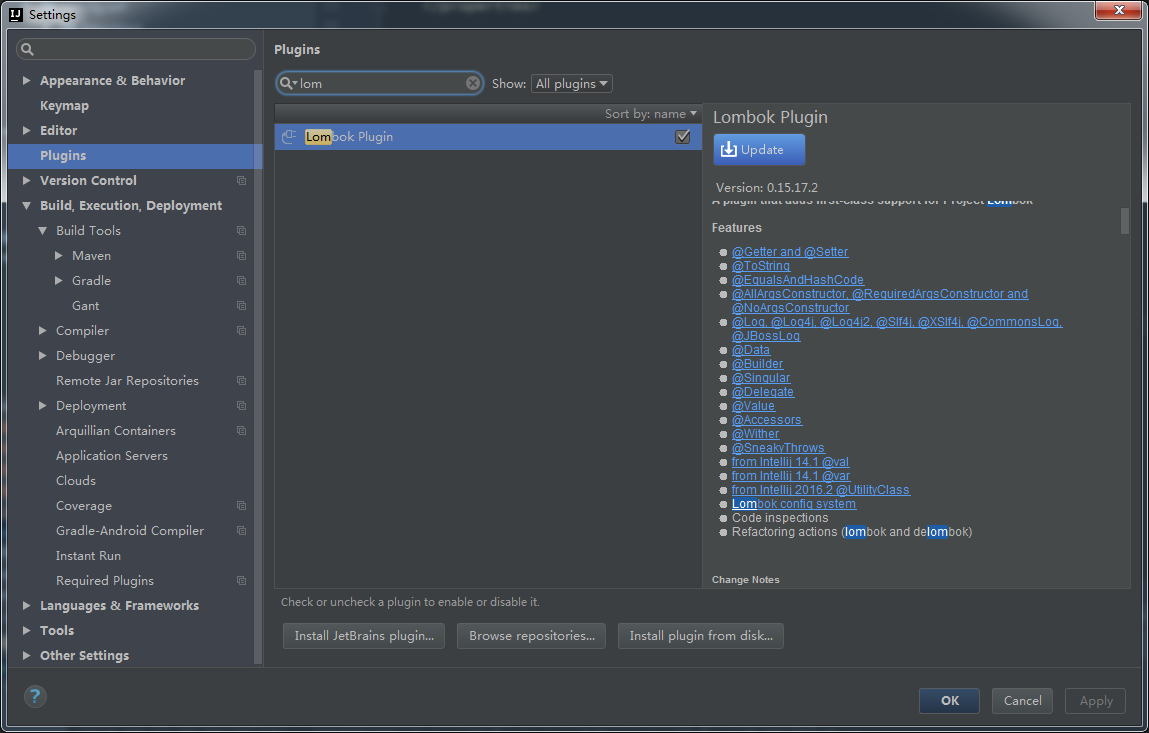




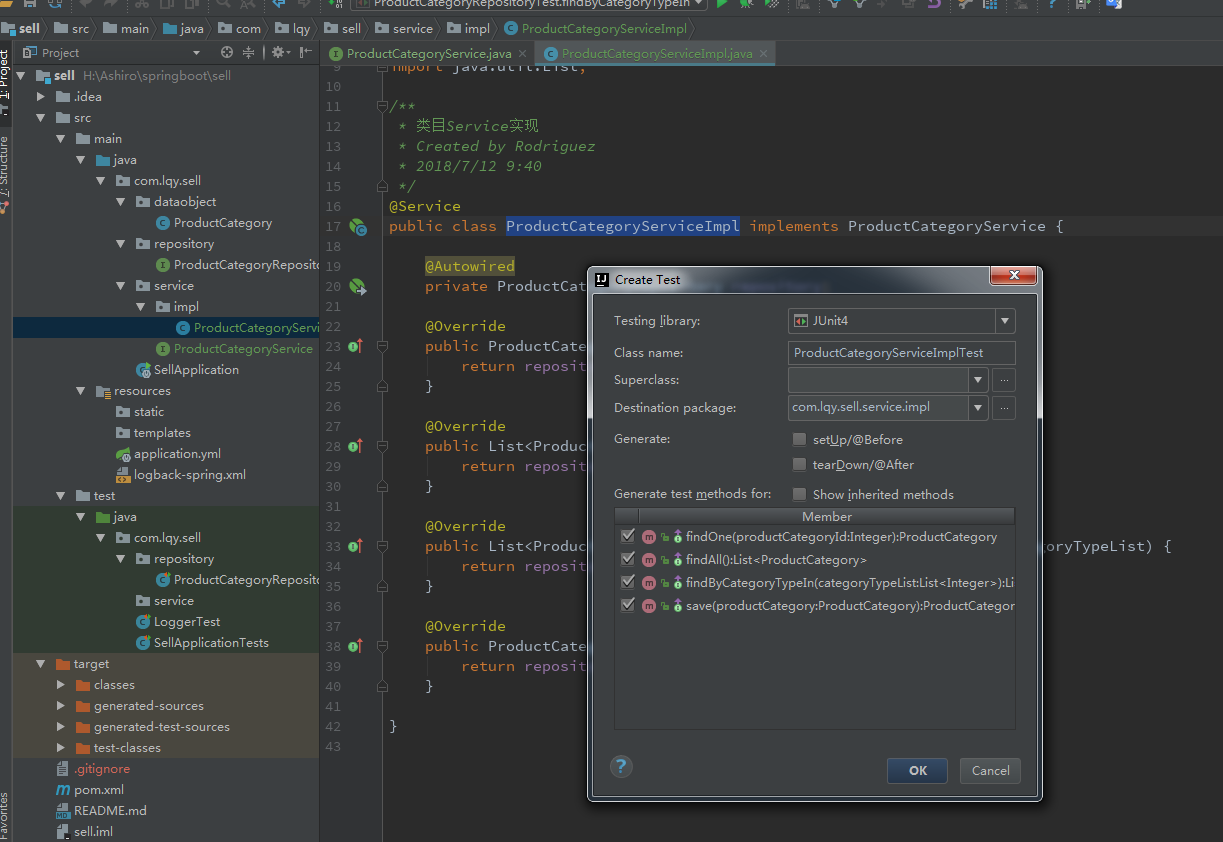
1. 功能实现

5.1.DAO层设计与开发



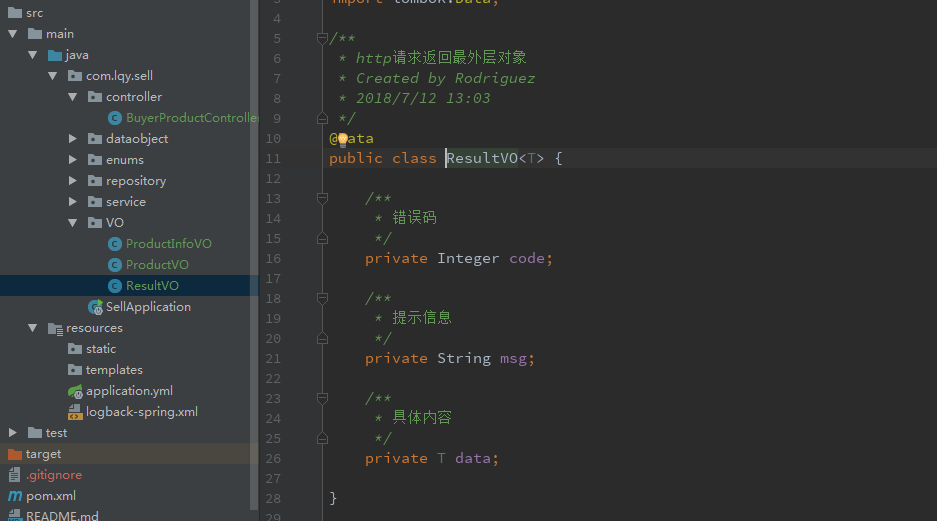


5.2.Service层设计与开发

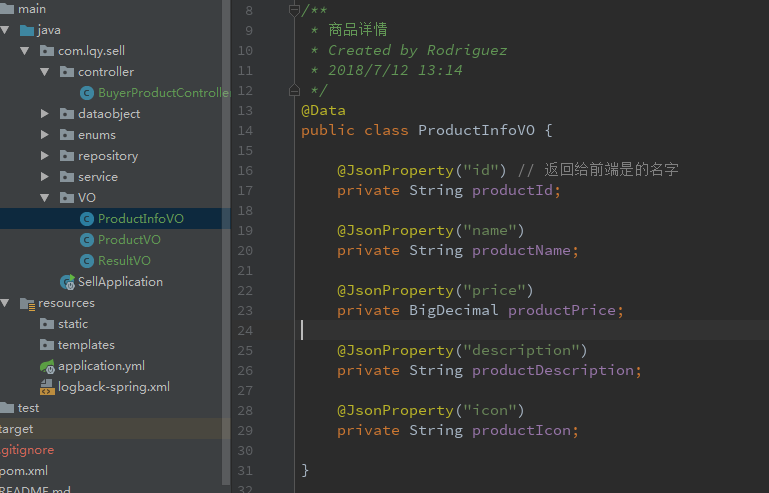


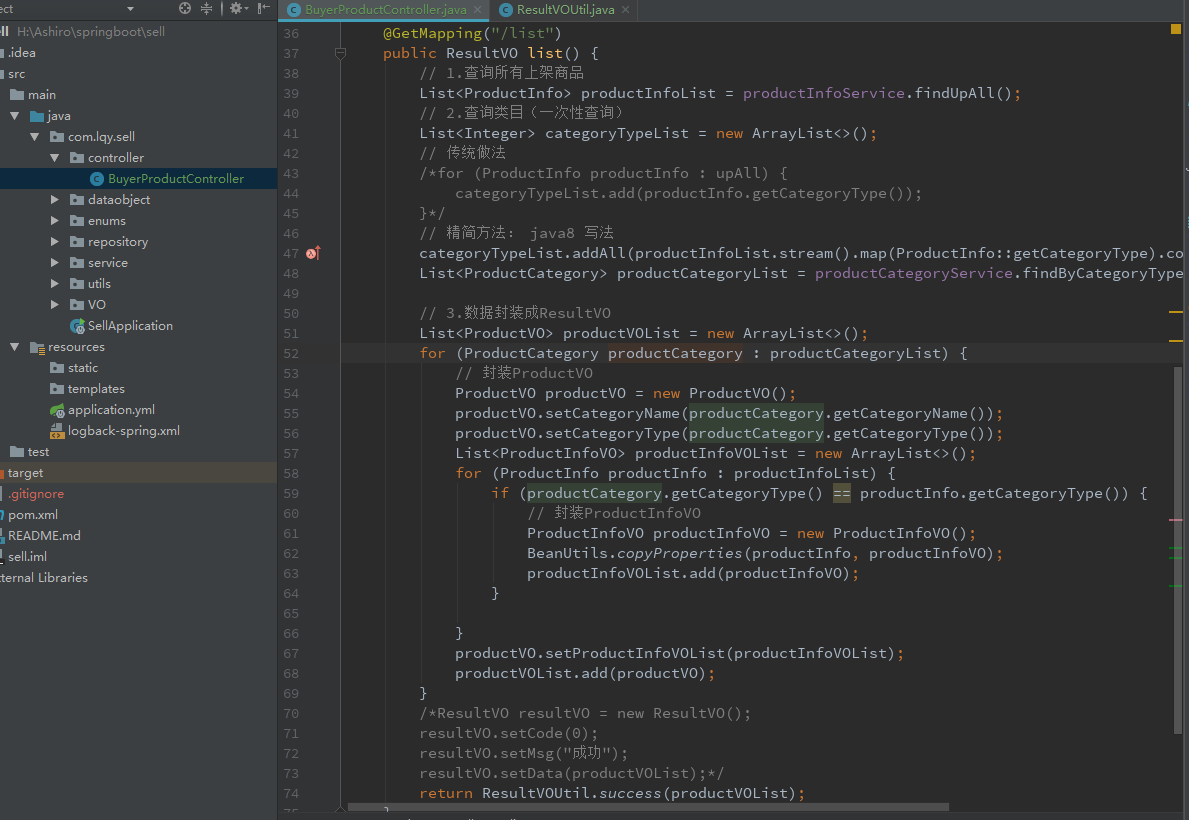
5.3.Controller层设计与开发

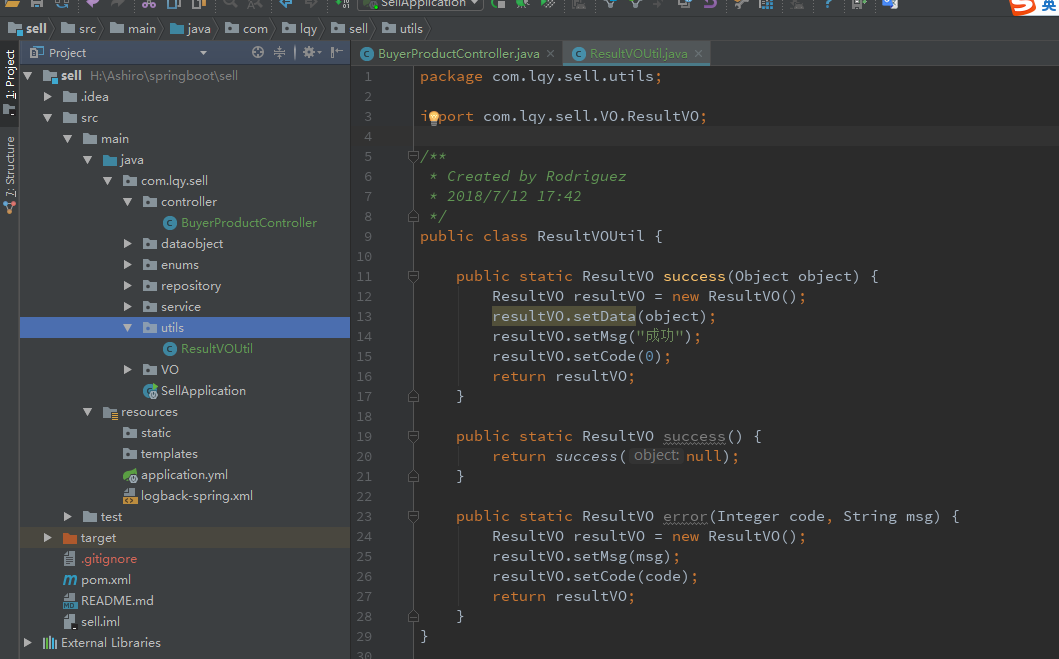








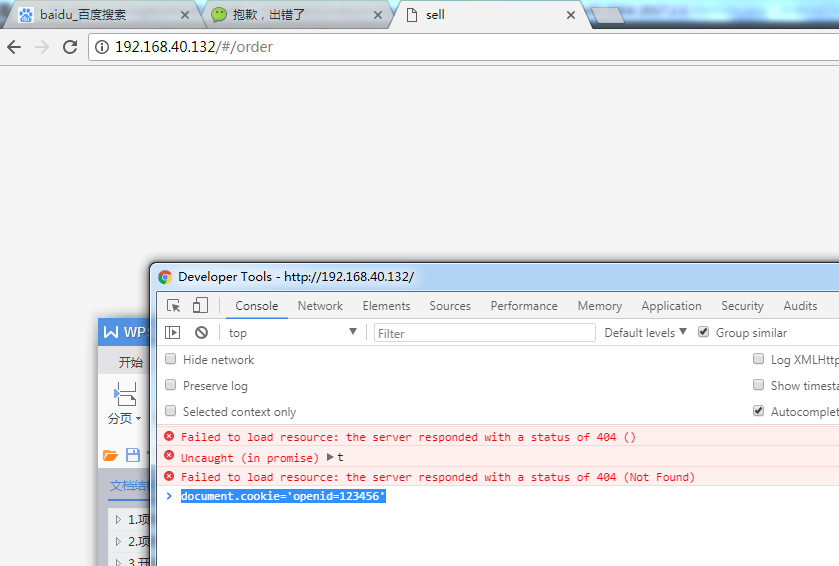


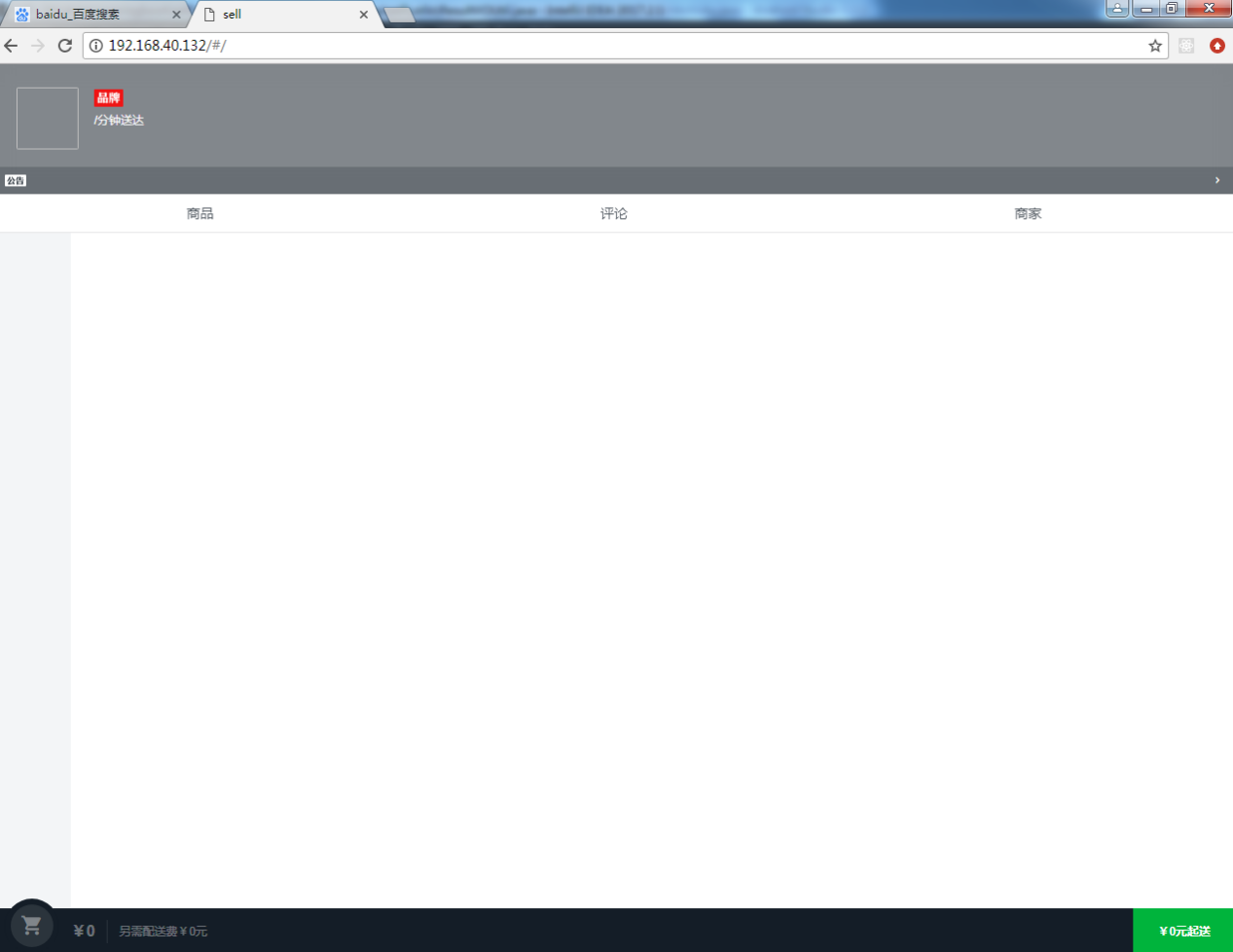


5.4.前端获取后端数据展示

浏览器访问：<http://虚拟机id地址/#/order>

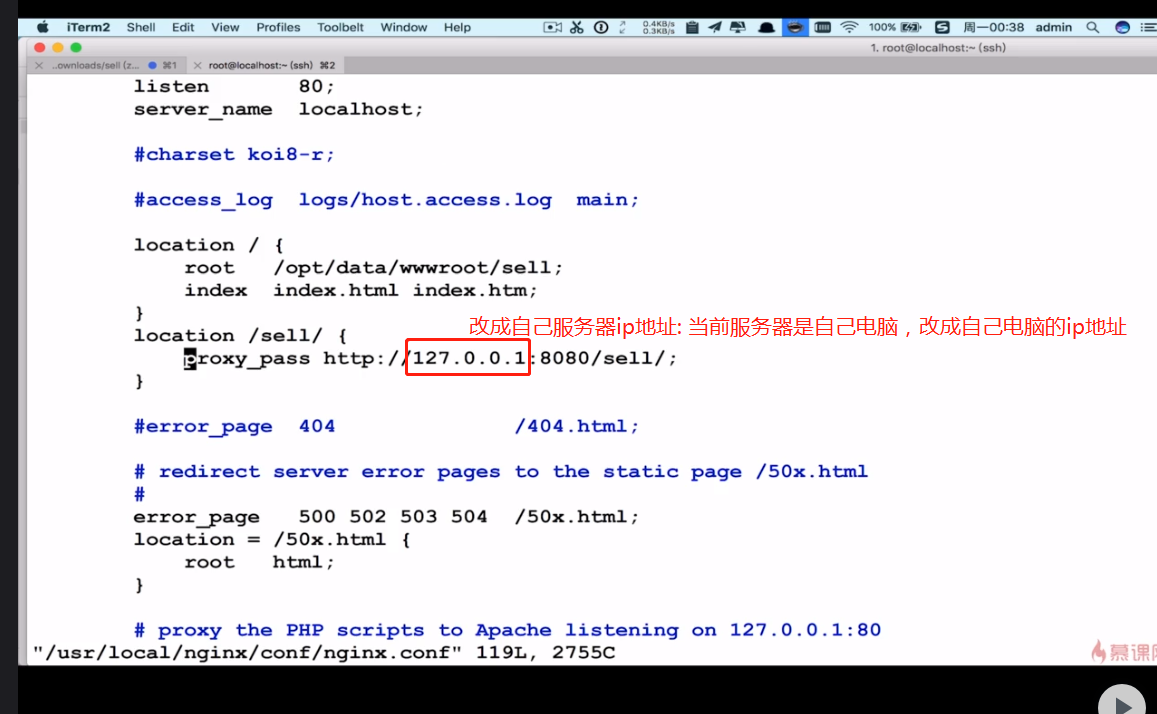
<http://192.168.40.132/#/order>

F12设置cookie：



修改虚拟机Nginx配置：

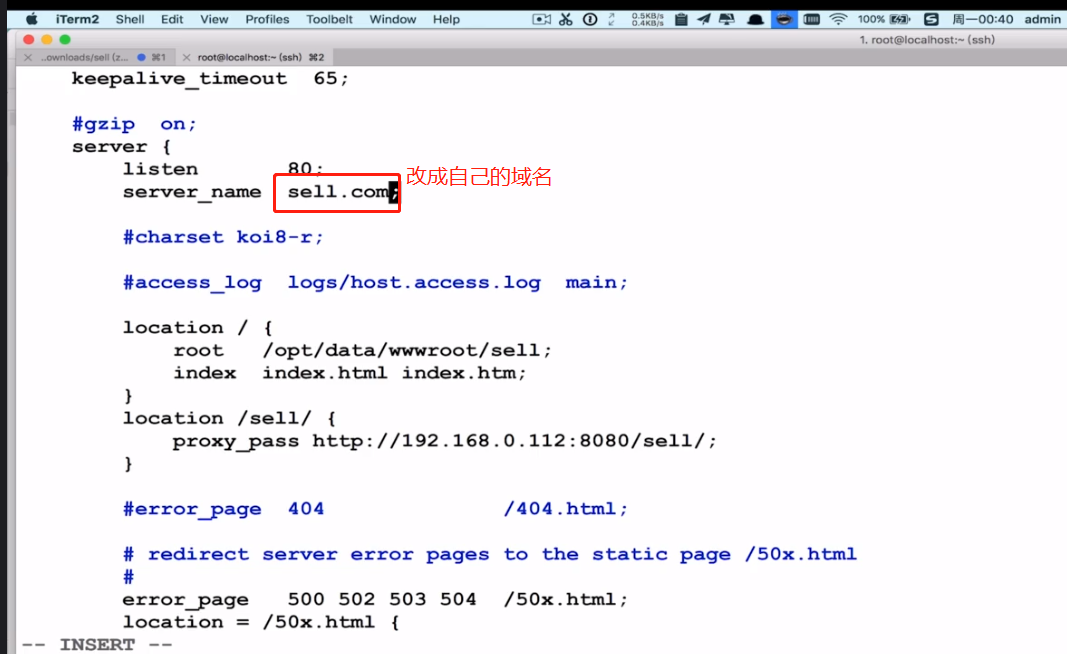
vim /usr/local/nginx/conf/nginx.conf ：



nginx -s reload: 重启nginx 服务器

修改域名：

vim /usr/local/nginx/conf/nginx.conf ：

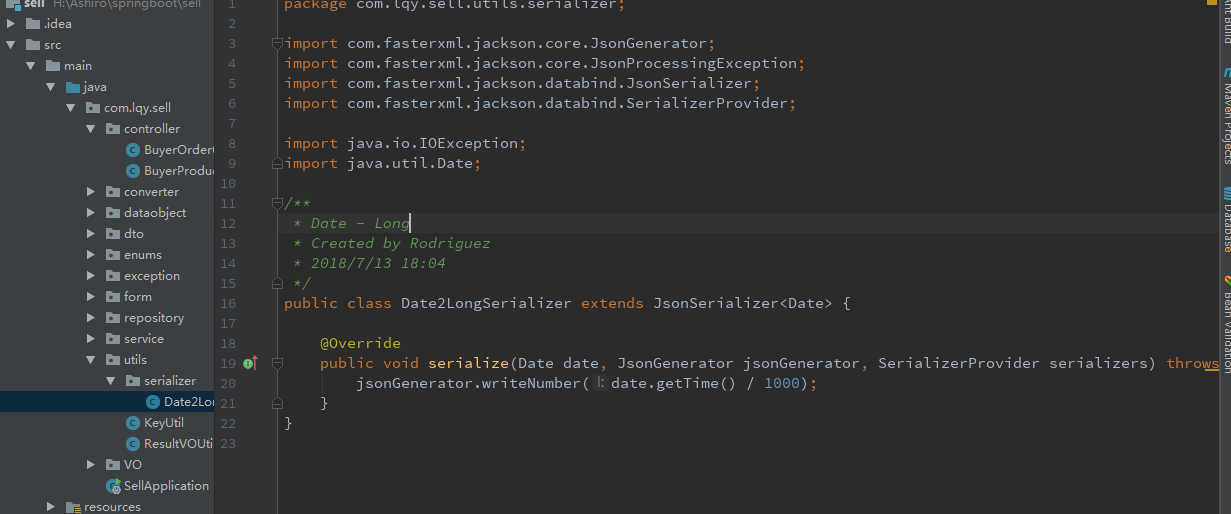


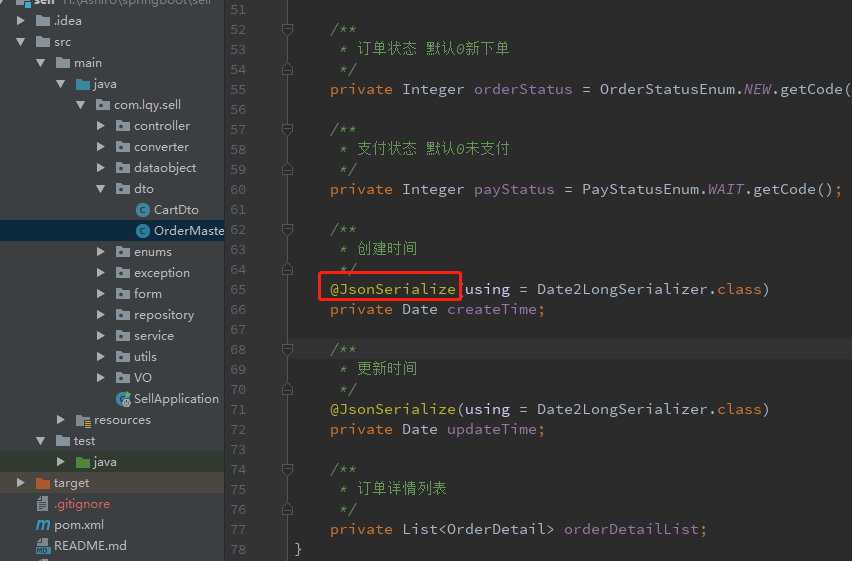
nginx -s reload: 重启nginx 服务器

修改本机的hosts文件

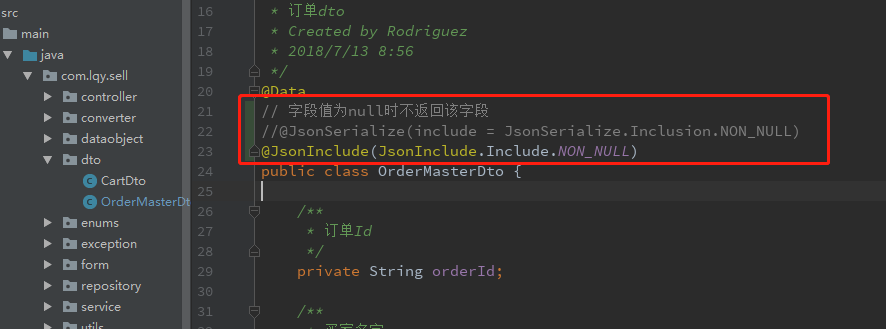
虚拟机ip地址 sell.com

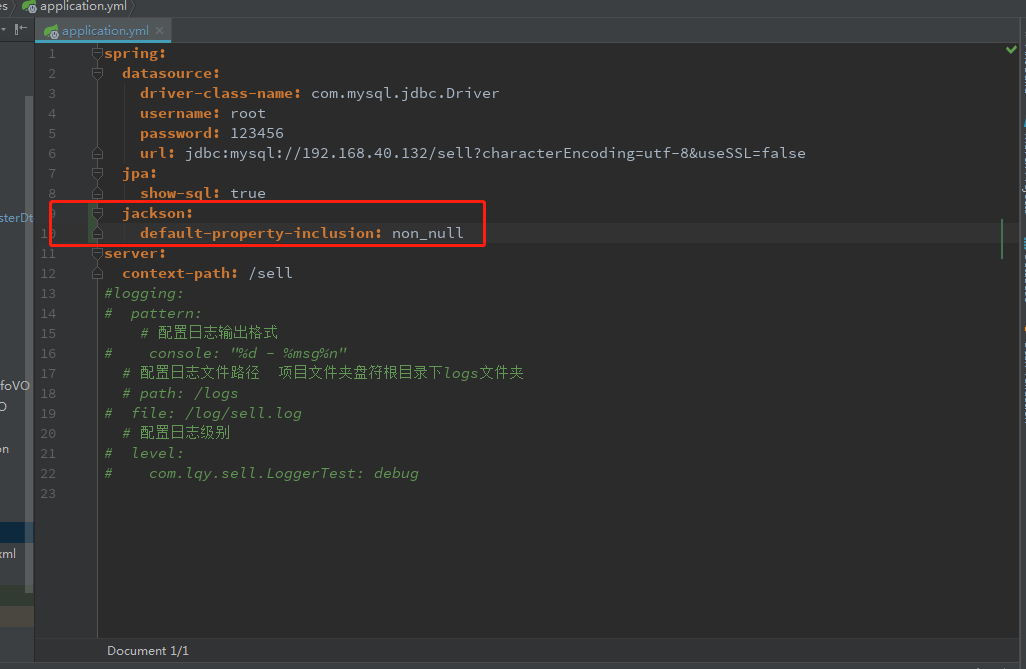
5.4.买家订单部分





字段值为null时不返回该字段





返回json必须带值，实体类字段后面赋初始值

1. 微信特性（重点和难点）

6.1.微信授权

微信文档要单独、仔细的查看一遍。6.1.1.微信网页授权

官方文档： <https://mp.weixin.qq.com/wiki>

微信支付文档： [https://pay.weixin.qq.com](https://mp.weixin.qq.com/wiki)

调试：<https://natapp.cn>

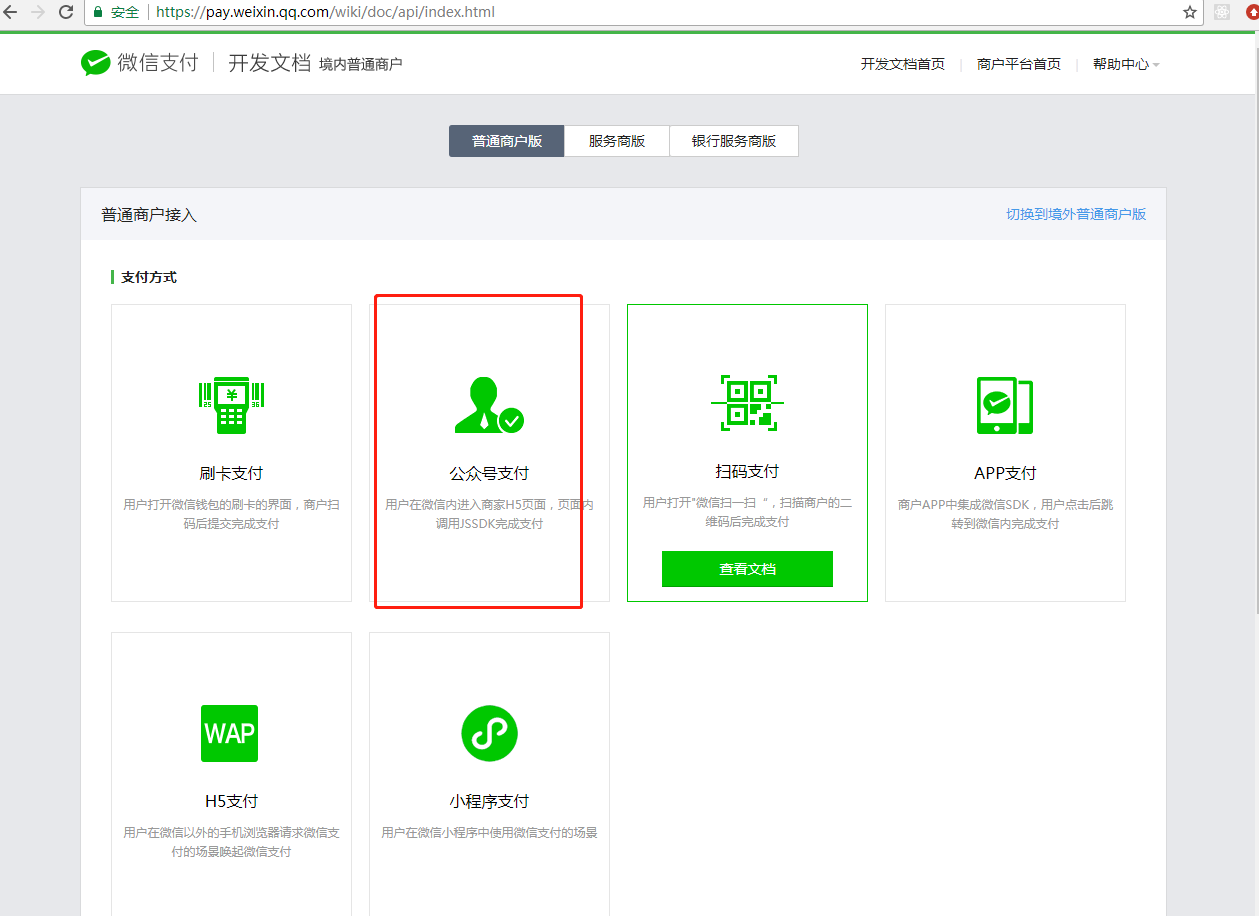
第三方SDK：<https://github.com/Wechat-Group/weixin-java-tools>

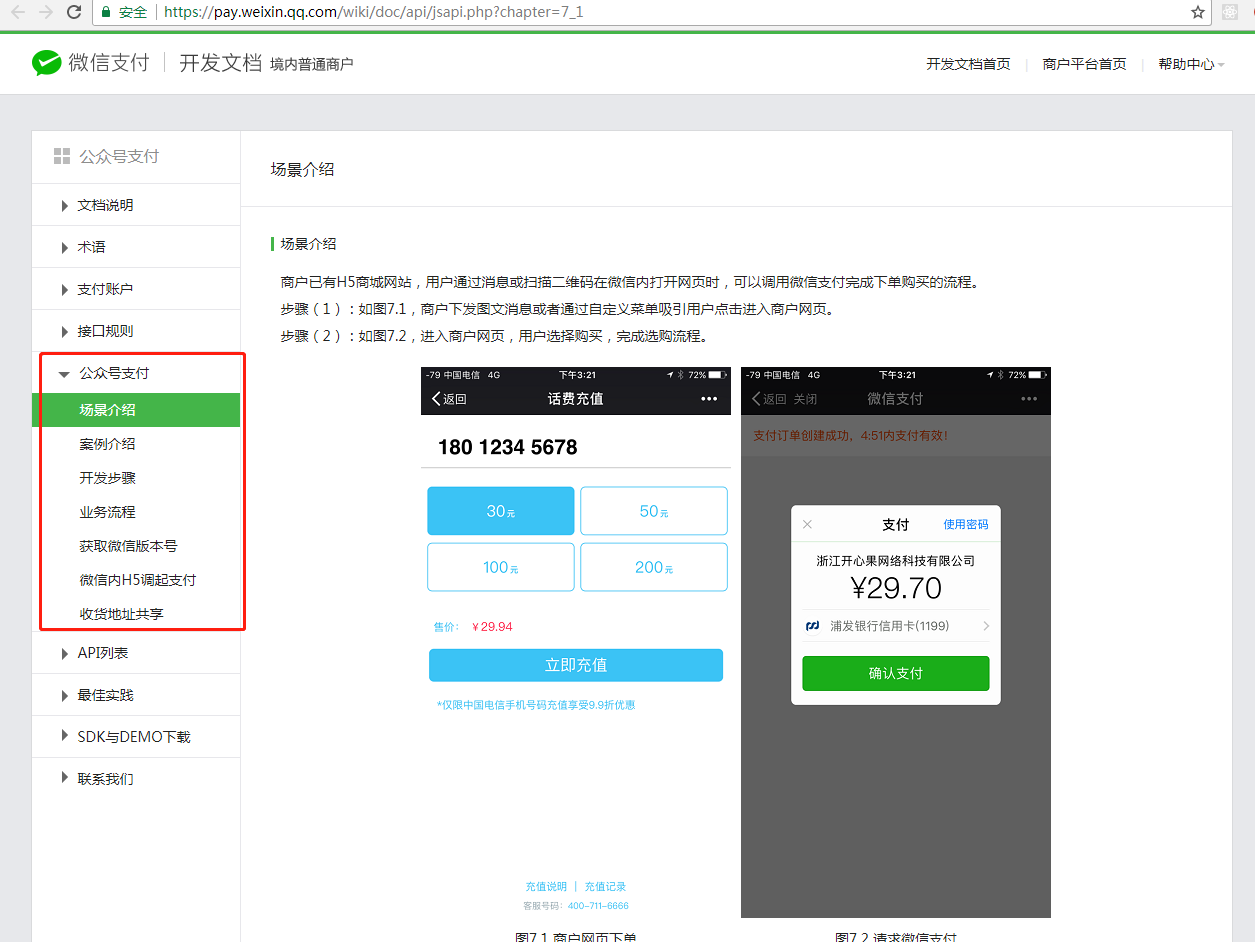
6.1.1.第一个关键点

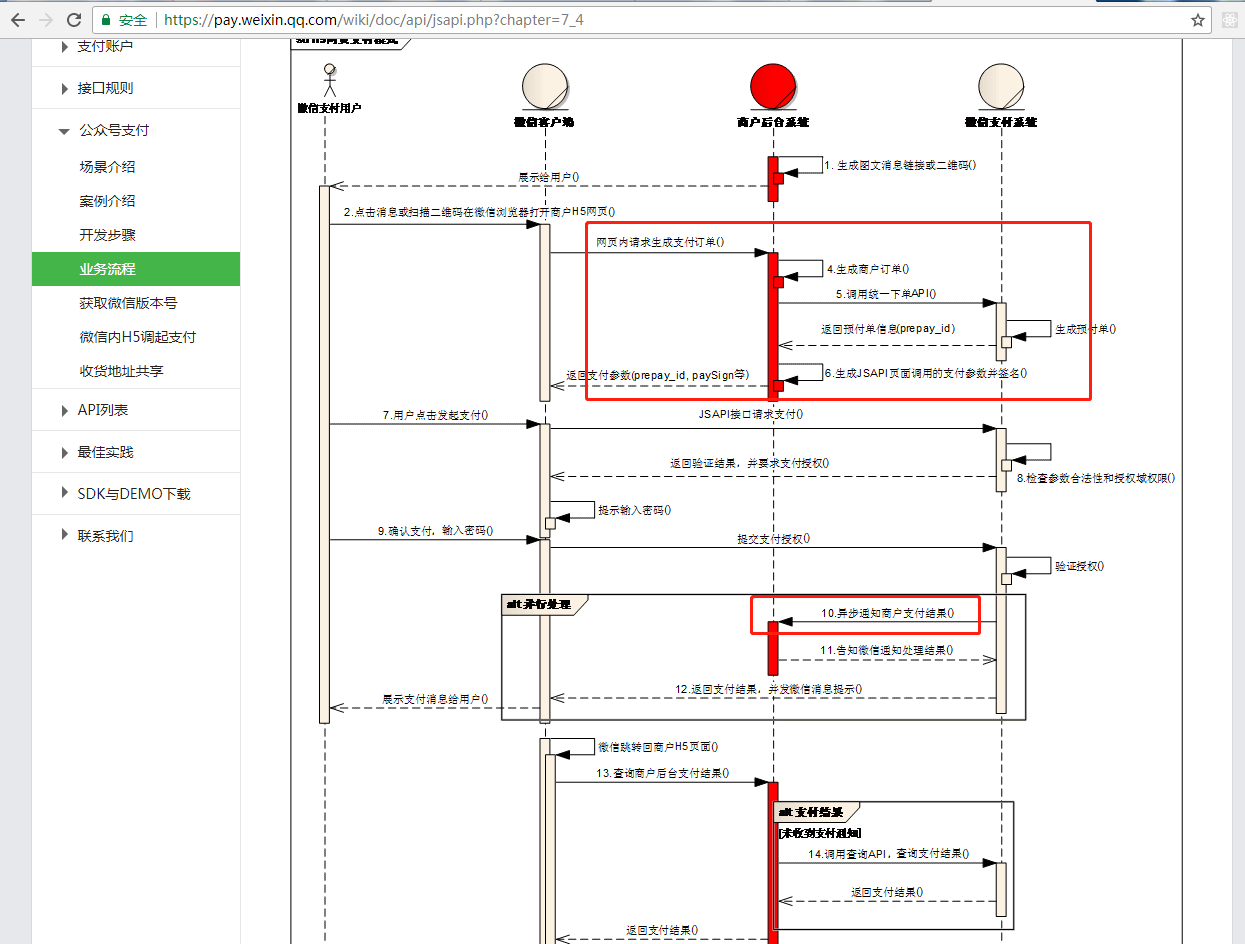
获取OpenId：手工方式、利用第三方SDK

手动方式：设置域名、获取code、换取access\_token（拿到openid）、刷新access\_token、拉取用户信息

利用第三方SDK：









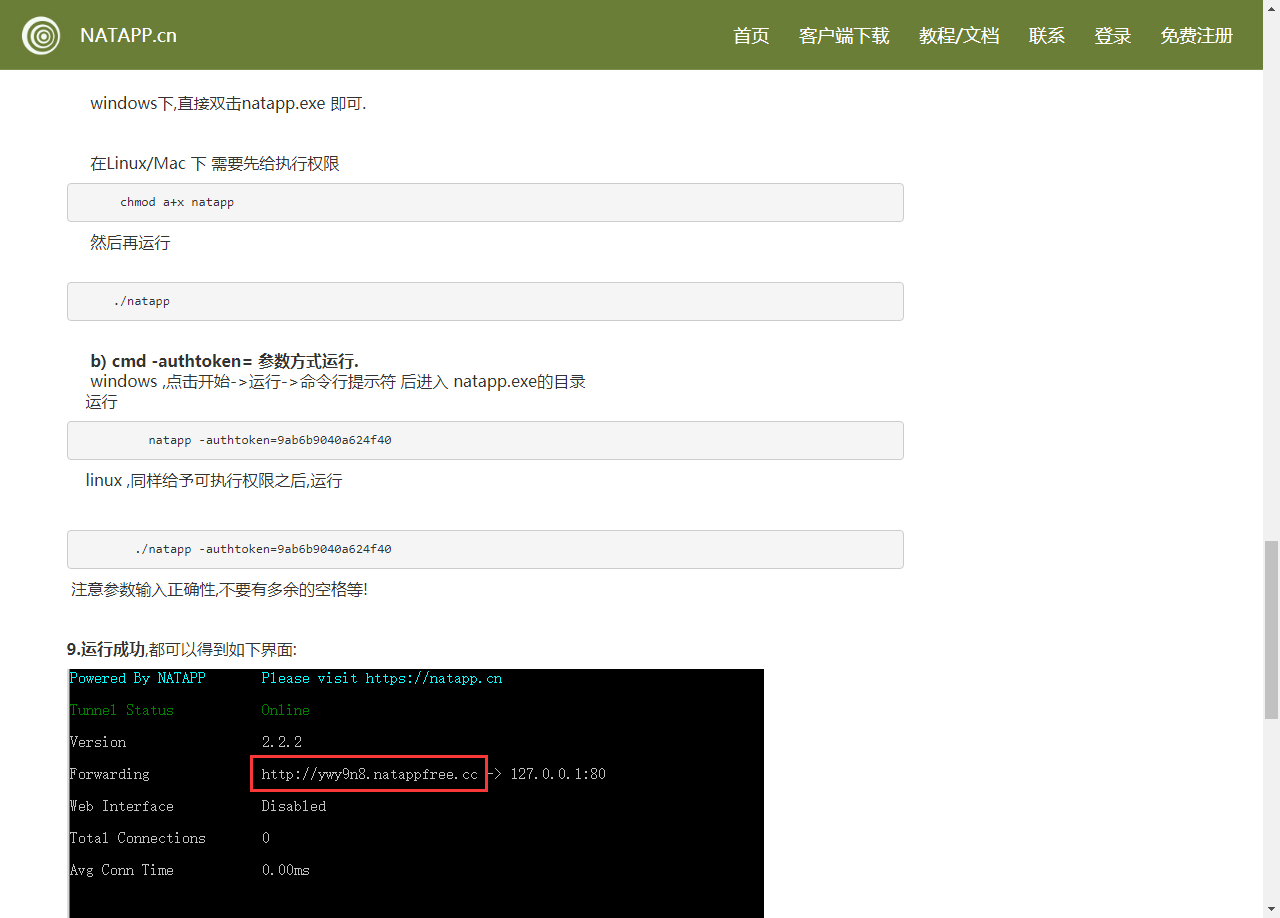




关于微信支付账号借用和开放平台账号借用的一些流程：<https://coding.imooc.com/learn/questiondetail/44160.html>

测试微信公众号使用：

<http://coding.imooc.com/learn/questiondetail/17686.html>





获取code

手机微信访问下面地址[https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx67ea954b523c12ee&redirect\_uri=http://lqyhmb.natapp1.cc/sell/weixin/auth&response\_type=code&scope=snsapi\_base&state=STATE#wechat\_redirect](https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx67ea954b523c12ee&redirect_uri=http:/lqyhmb.natapp1.cc/sell/weixin/auth&response_type=code&scope=snsapi_base&state=STATE#wechat_redirect)

得到：<http://lqyhmb.natapp1.cc/sell/weixin/auth?code=xxxxxxxx>

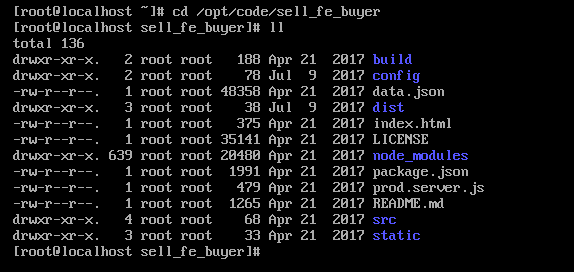
获取code后，请求以下链接获取access\_token和openid： [https://api.weixin.qq.com/sns/oauth2/access\_token?appid=wx67ea954b523c12ee&secret=3cd4b136aebb4450b16255ee0fb96a55&code=0712NLJU0B5z9X15wZIU0ykGJU02NLJR&grant\_type=authorization\_code](https://api.weixin.qq.com/sns/oauth2/access_token?appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code)

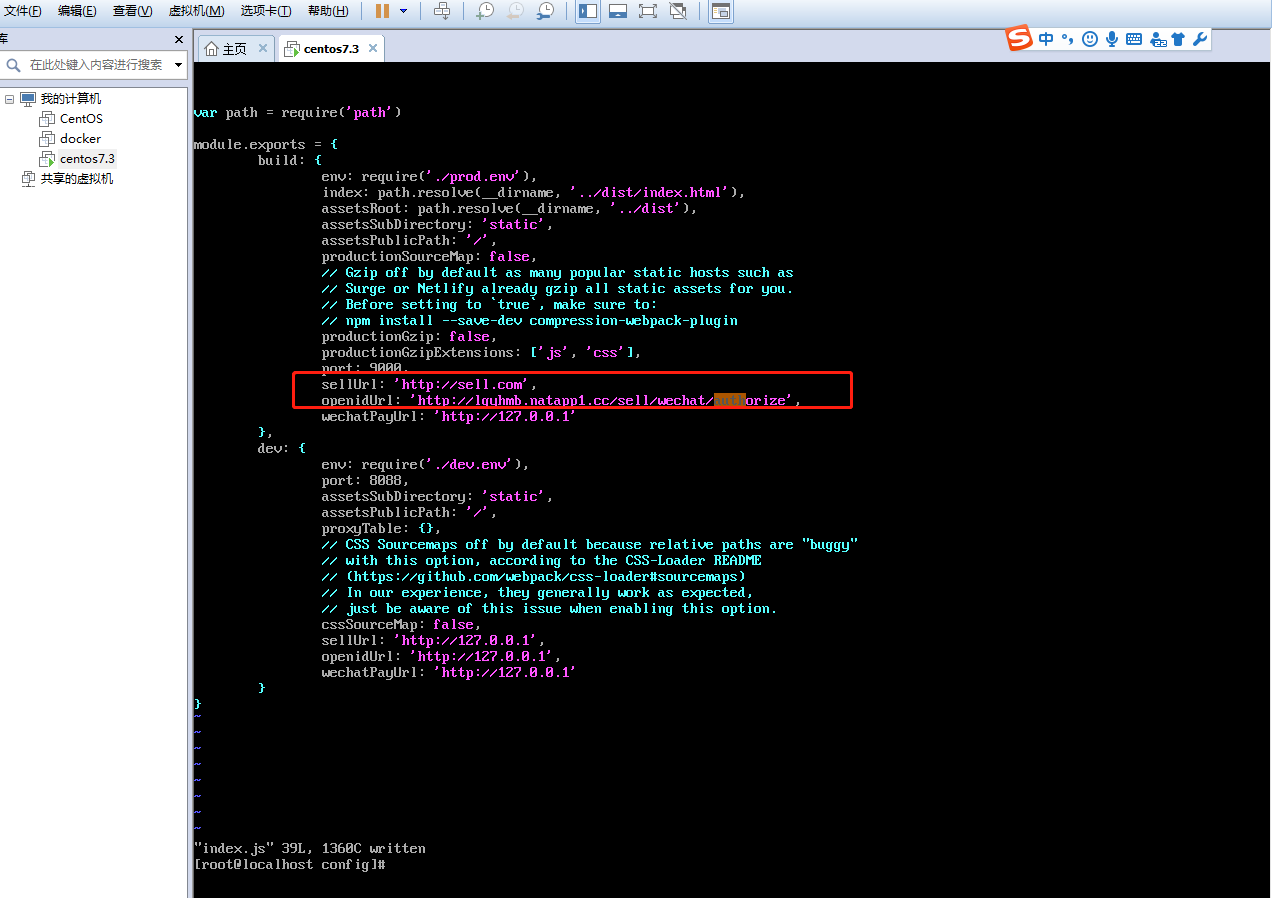
利用第三方sdk：

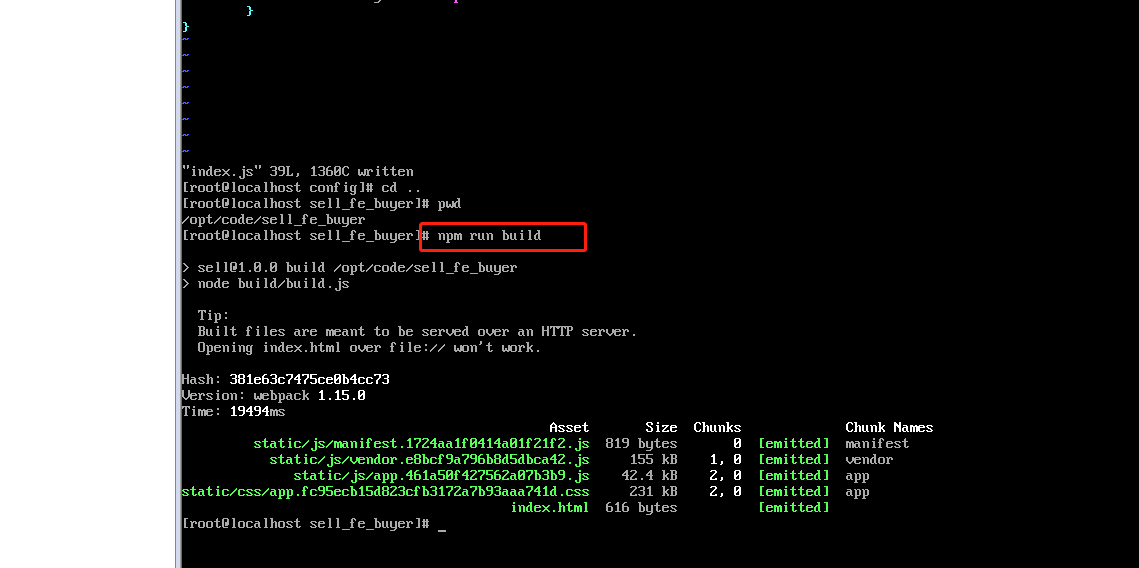
<https://github.com/wechat-group/weixin-java-tools/wiki>



虚拟机配置前端项目：







\cp -r dist/\* /opt/data/wwwroot/sell

抓包工具： fidder

6.2.微信支付

6.3.微信退款