



链滴

# 商城笔记 1-- 运行环境搭建

作者: [wenyl](#)

原文链接: <https://ld246.com/article/1682409442385>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 环境搭建

### 虚拟机

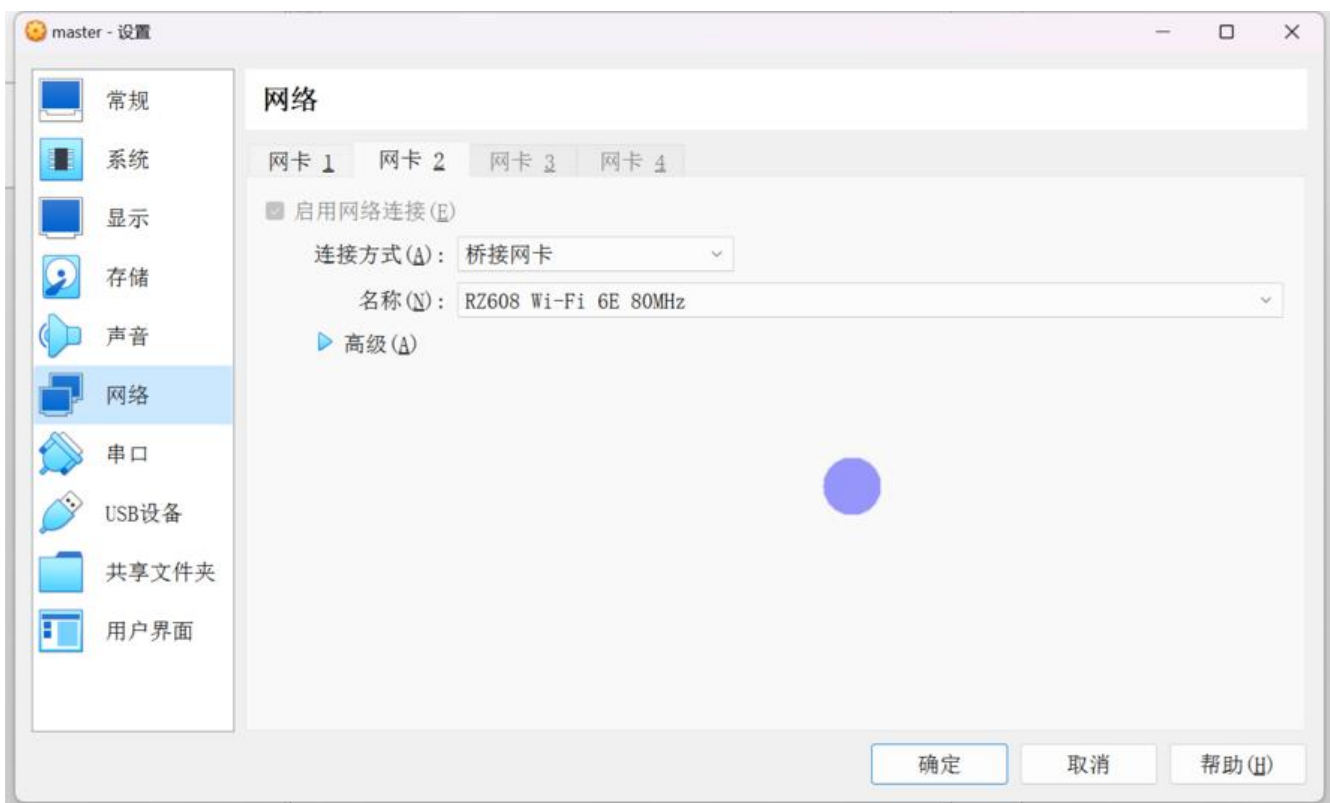
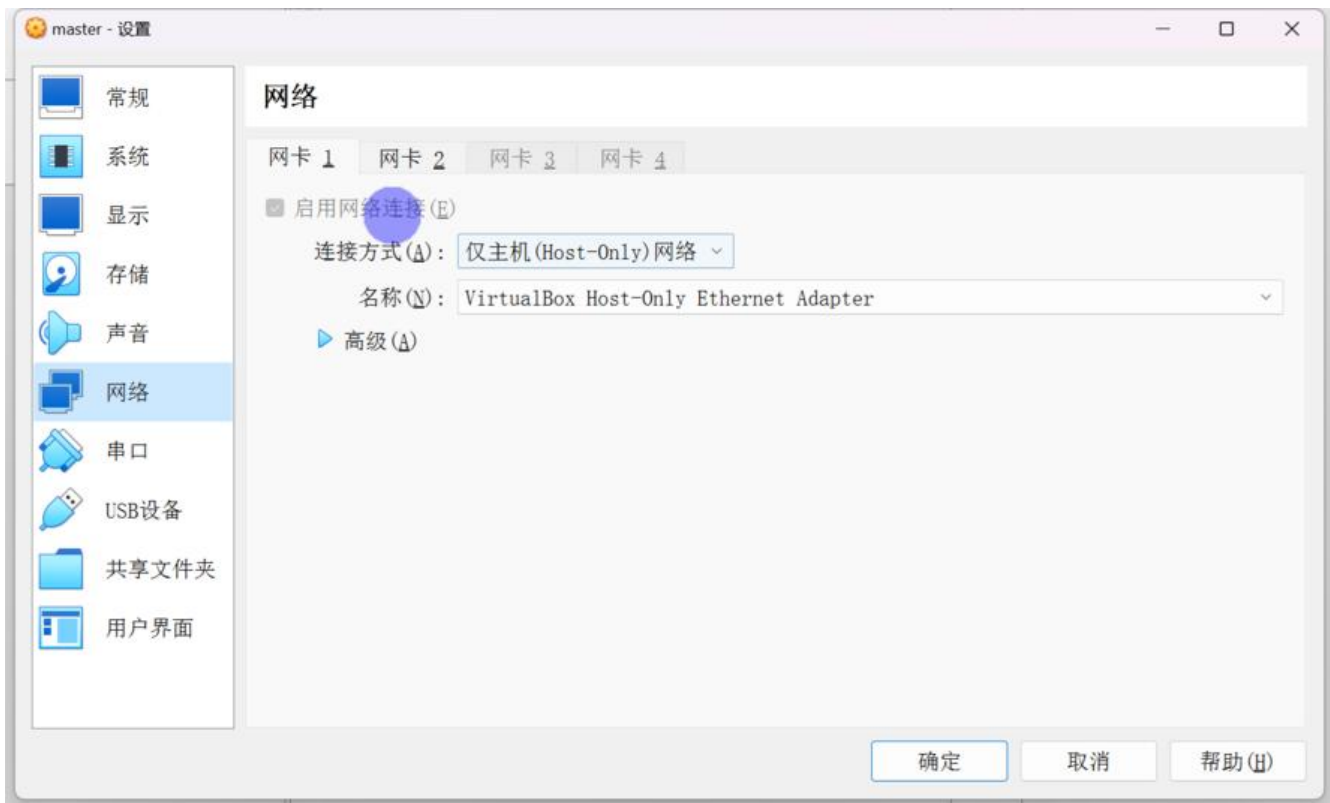
使用virturebox安装三个centos虚拟机

### 虚拟机网络配置

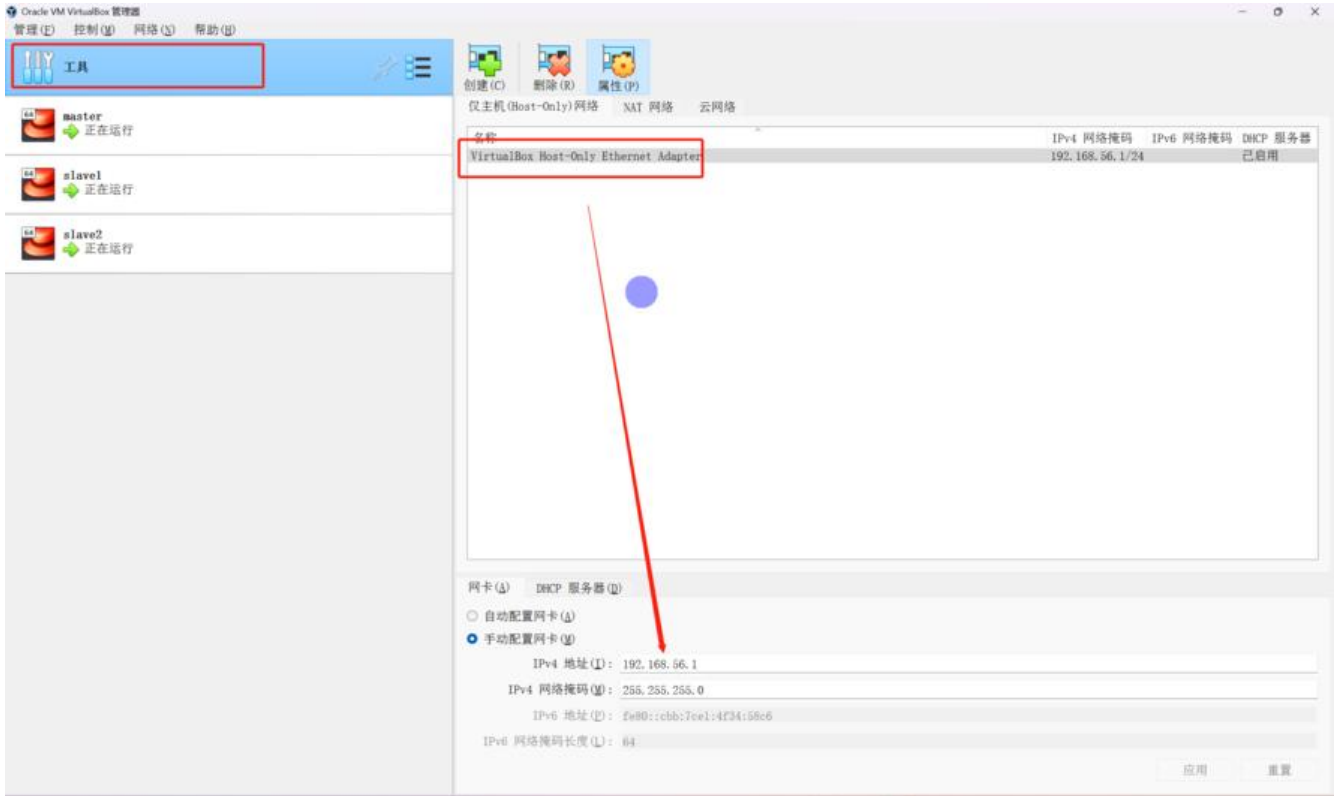
为了防止虚拟机网络随着我们主机网络的变化而变化，对虚拟机的网络做如下配置

### 网卡

给虚拟机建立两个网卡（网卡选定即可，使用默认配置，无需修改）



在工具中查看网卡地址信息，没有的话自己配置一个，记录这里的网关和子网掩码



## 虚拟机中配置网络

```
vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

打开后在最后新增配置

```
IPADDR=192.168.56.5  
NETMASK=255.255.255.0  
GATEWAY=192.168.56.1  
DNS1=114.114.114.114  
PREFIX=24
```

修改配置项信息

```
BOOTPROTO=static  
DEFROUTE=no  
ONBOOT=yes
```

## docker安装

### 1、卸载docker

```
sudo yum remove docker docker-client docker-client-latest docker-common docker-latest do  
ker-latest-logrotate docker-logrotate docker-engine
```

### 2、yum-utils安装

```
sudo yum install -y yum-utils
```

### 3、yum源设置

```
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.
epo
```

#### 4、安装docker

```
sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose
plugin
```

#### 5、配置阿里镜像源

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://h0fcunc7.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

## docker安装mysql

### 拉取镜像

```
docker pull mysql:5.7
```

### 运行容器

```
docker run -p 3306:3306 --name mysql \
-v /opt/mydata/mysql/log:/var/log/mysql \
-v /opt/mydata/mysql/data:/var/lib/mysql \
-v /opt/mydata/mysql/conf:/etc/mysql \
-e MYSQL_ROOT_PASSWORD=root \
-d mysql:5.7
```

### 修改配置

建立配置文件

```
cd /opt/mydata/mysql/conf/
vi my.cnf
```

配置信息如下

```
[client]
default-character-set=utf8
```

```
[default]default-character-set=utf8
```

```
[mysqld]
init_connect="SET collation_connection = utf8_unicode_ci"
init_connect="SET NAMES utf8"
character-set-server=utf8
```

```
collation-server=utf8_unicode_ci
skip-character-set-client-handshake
skip-name-resolve
```

## docker安装redis

### 拉取镜像

```
docker pull redis
```

### 运行容器

建立配置文件

```
mkdir -p /opt/mydata/redis/conf \
vi /opt/mydata/redis/conf/redis.conf
```

再redis.conf中填写配置信息

```
bind 0.0.0.0
appendonly yes
requirepass redis123
```

运行容器

```
docker run -p 6379:6379 --name redis \
-v /opt/mydata/redis/data:/data \
-v /opt/mydata/redis/conf/redis.conf:/etc/redis/redis.conf \
-d redis redis-server /etc/redis/redis.conf
```

## mavne镜像仓库配置

再setting.conf的配置文件mirrors节点下添加

```
<mirror>
  <id>nexus-aliyun</id>
  <mirrorOf>central</mirrorOf>
  <name>nexus maven</name>
  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
</mirror>
```

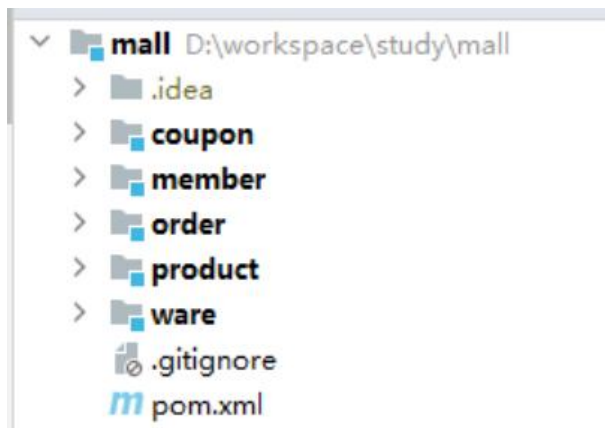
在setting.conf的配置文件profiles节点下添加，这里我本机jdk是1.8，版本根据自己jdk版本调整

```
<profile>
  <id>jdk-1.8</id>
  <activation>
    <activeByDefault>true</activeByDefault>
    <jdk>1.8</jdk>
  </activation>
  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
```

```
<maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>
</properties>
</profile>
```

## 新建git仓库，并建立项目

项目结构如下，spring boot版本根据自己需要选择



每个项目pom中引入

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>
```

## 数据库初始化

执行git项目的docs目录下的几个SQL文件，先执行init.sql，其他文件顺序任意[SQL文件地址点这里](#)

## 使用人人框架作为后台管理

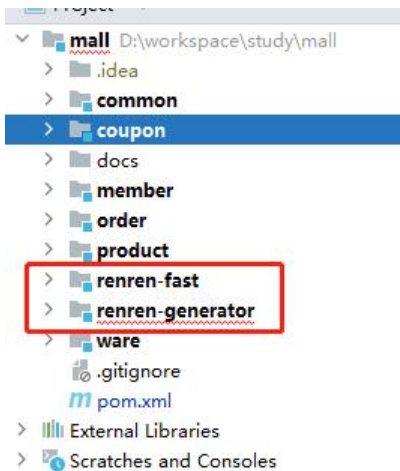
下载前后端三个git项目

<https://gitee.com/renrenio/renren-fast.git> 后台管理服务

<https://gitee.com/renrenio/renren-fast-vue.git> 后台页面

<https://gitee.com/renrenio/renren-generator.git> 代码生成器

下载成功后删除两个目录下的隐藏git文件，将renren-fast复制到mall目录下，作为一个模块添加到mall项目，修改application.yml中的数据库配置，在前面的数据库初始化环节，我将renren-fast的数据名修改为mall\_admin，所以这里要对应修改



## 运行前后台

启动renren-fast后台

在renren-fast-vue中执行npm install,npm run dev启动即可

## 前端启动问题汇总：

1、node版本

node版本尝试了12、14、16、18，推荐使用node14版本，其他版本使用过程中会遇到各种各样的问题

2、权限问题

给nodejs安装目录完全控制权限，否则会出现windows系统安装过程中

3、python编码问题

python安装的是2.7版本，npm install过程中遇到 `UnicodeDecodeError: 'ascii' codec can't decode byte 0xc4 in position 7: ordinal not in range(128)`类似的问题，参考这篇文章处理 [【python】pip 装报错UnicodeDecodeError: 'ascii' codec can't decode byte 0xc4 in position 7: ordinal not in range\(128\) - 放飞的回忆 - 博客园 \(cnblogs.com\)](#)

前后端启动后就可以正常使用系统了

## 代码生成器

修改renren-generator中的数据库配置信息，要生成哪个库里表得代码就改成哪个

```
spring:
  datasource:
    type: com.alibaba.druid.pool.DruidDataSource
    #MySQL配置
    driverClassName: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://192.168.8.5:3306/mall_sms?useUnicode=true&cf
    username: root
    password: root
```



修改generator.properties配置，mainPath何package就是src下得路径，我的包路径是com.mall.coupon，所以将mainPath,package设置为com.mall.coupon,moduleName设置为coupon，生成的类会在com.mall.coupon路径下，tablePrefix用于去除表前缀，我的数据库表名称为sms\_product,配了tablePrefix属性后，生成的类名中就不会包含sms\_这个前缀

```
mainPath=com.mall
#\u5305\u540D
package=com.mall
moduleName=coupon
#\u4F5C\u8005
author=WenYL
#Email
email=3424675994@qq.com
#\u8868\u524D\u7F00(\u7C7B\u524D\u7F00)
tablePrefix=sms_
```

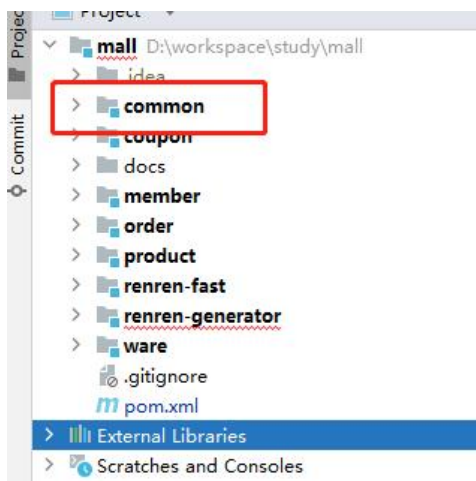
代码生成器中使用的是shiro，如果要替换成spring security或其他，需要改模板中的配置

修改renren-generator工程中，template目录下得controller.java.vm文件

将涉及RequirePermissions得地方都进行注释

## common模块处理

renren-fast是一个单服务框架，我们生成的代码里引用了很多他自己封装的防范，放在我们自己的代中，由于和renren-fast不是一个服务下，所以需要建立common模块，将公用的服务依赖放在common模块中，新词在根目录下建立一个common模块



建立好后，去看之前生成的代码，那些类没有引用的，就从renren-fast中引入到common服务

common处理好后，在其他工程中引入这个common依赖，然后复制生成的代码到对应的工程即可

## 项目配置yml文件

配置数据库连接信息，配置mybatis-plus信息

```
server:
  port: 9004
spring:
  datasource:
```

```
username: root
password: root
url: jdbc:mysql://192.168.8.5:3306/mall_pms?useUnicode=true&characterEncoding=UTF-8
useSSL=false&serverTimezone=Asia/Shanghai
driverClassName: com.mysql.cj.jdbc.Driver
```

```
mybatis-plus:
  mapperLocation: classpath*/mapper/**/*.xml
  globalConfig:
    dbConfig:
      idType: auto
```

## application启动类配置

启动类添加mapperScan扫描mybatis接口

```
@MapperScan("com.mall.product.dao")
```