课程回顾

# web开发Django电商项目

生鲜

1. 卖家
   1. 上传商品
   2. 编辑商品
   3. 订单
   4. 店铺
2. 买家
   1. 购买商品
   2. 购物车
   3. 付款
   4. 订单
   5. 收货地址

# 电商项目工程

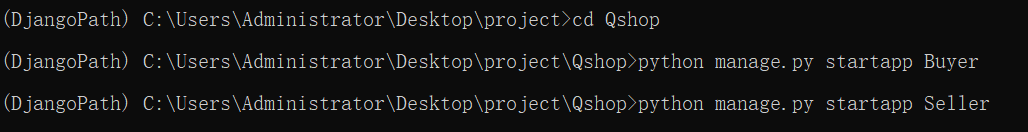
### 创建工程 Qshop

django-admin startproject Qshop

### 创建子应用 Buyer Seller

buyer 买家

seller 卖家

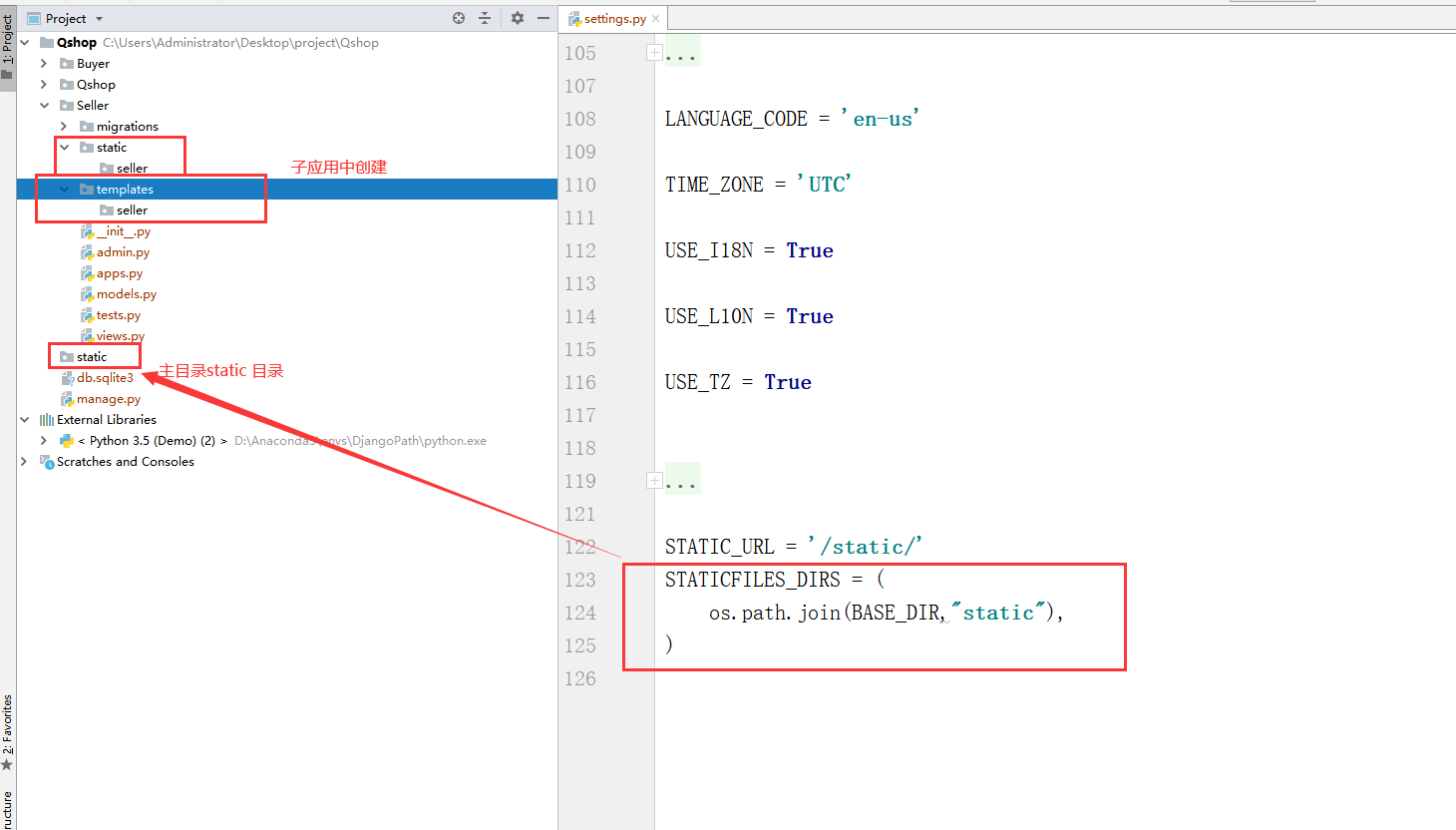


### 注册app



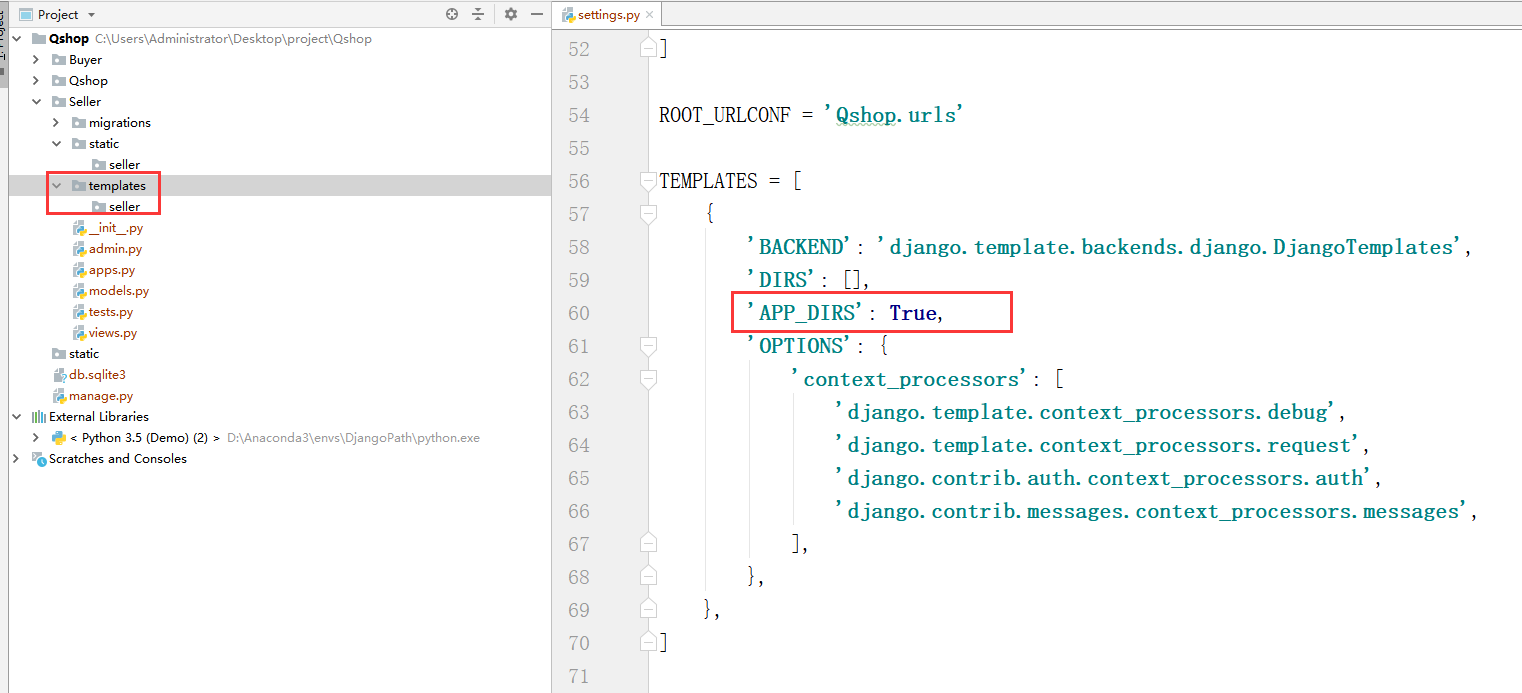
### 模板静态文件目录创建

/static/ 指向了 项目主目录中的static 目录



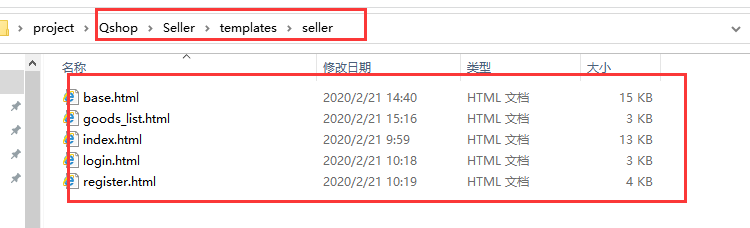
子应用中创建templates 目录，确保settings中的appdirs = True，

当 app\_dirs =True时，django会自动从子应用中寻找模板文件

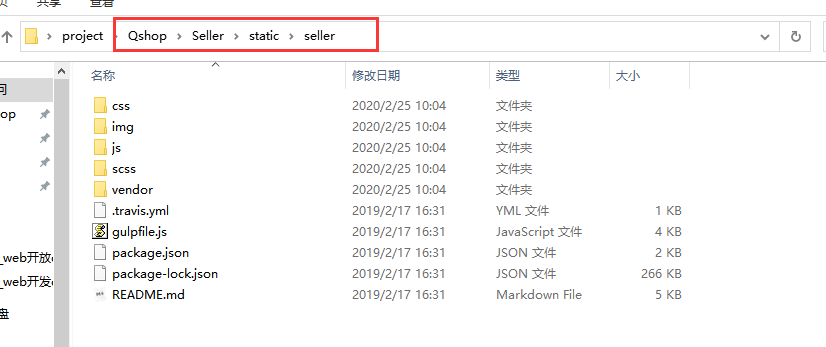


### 导入静态文件，模板文件

模板文件



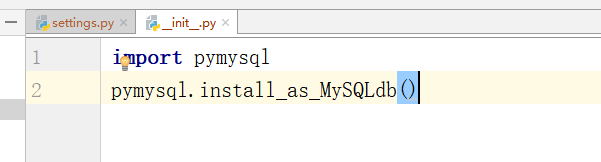
静态文件



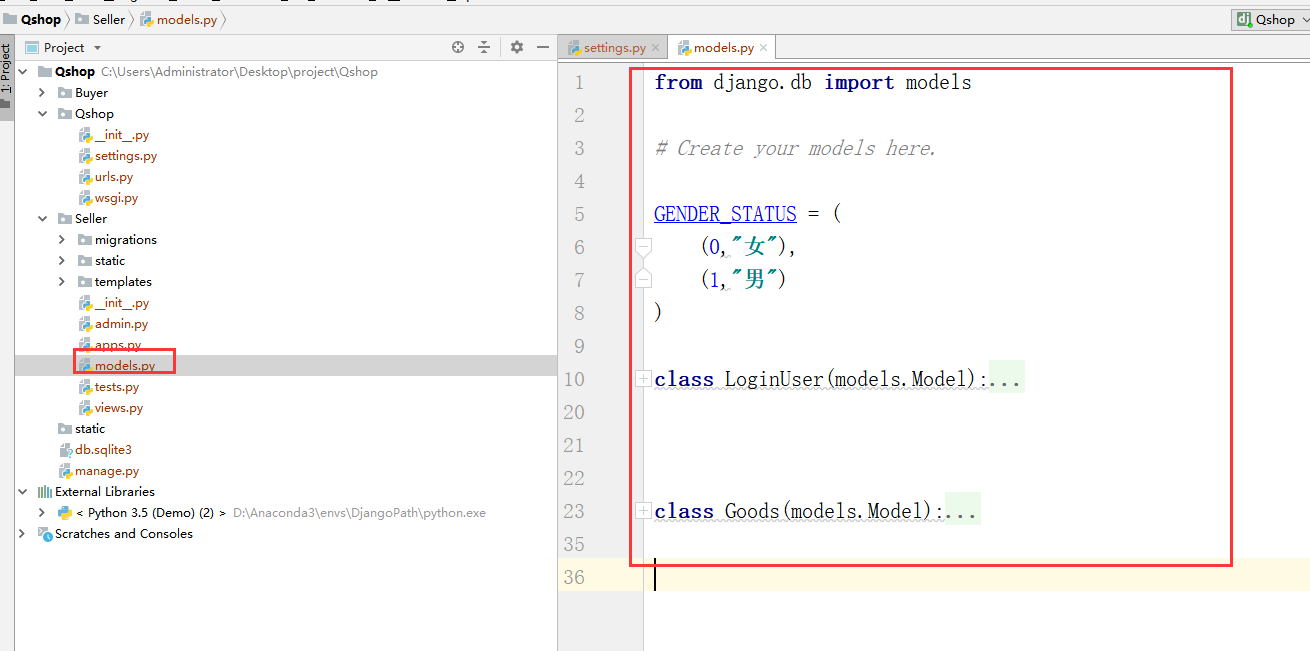
### 使用mysql数据库

settings中修改数据的配置

DATABASES = {  
 **'default'**: {  
 **'ENGINE'**: **'django.db.backends.mysql'**,  
 **'NAME'**: **'qshop'**,  
 **'USER'**:**'root'**,  
 **'PASSWORD'**:**'111111'**,  
 **'HOST'**:**'127.0.0.1'**,  
 **'PORT'**:**'3306'** }  
}



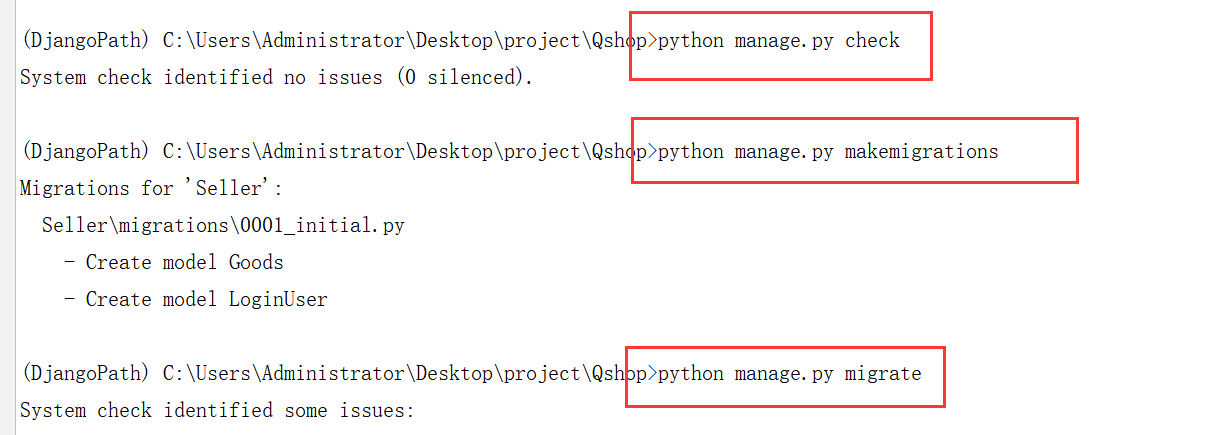
### 复制模型



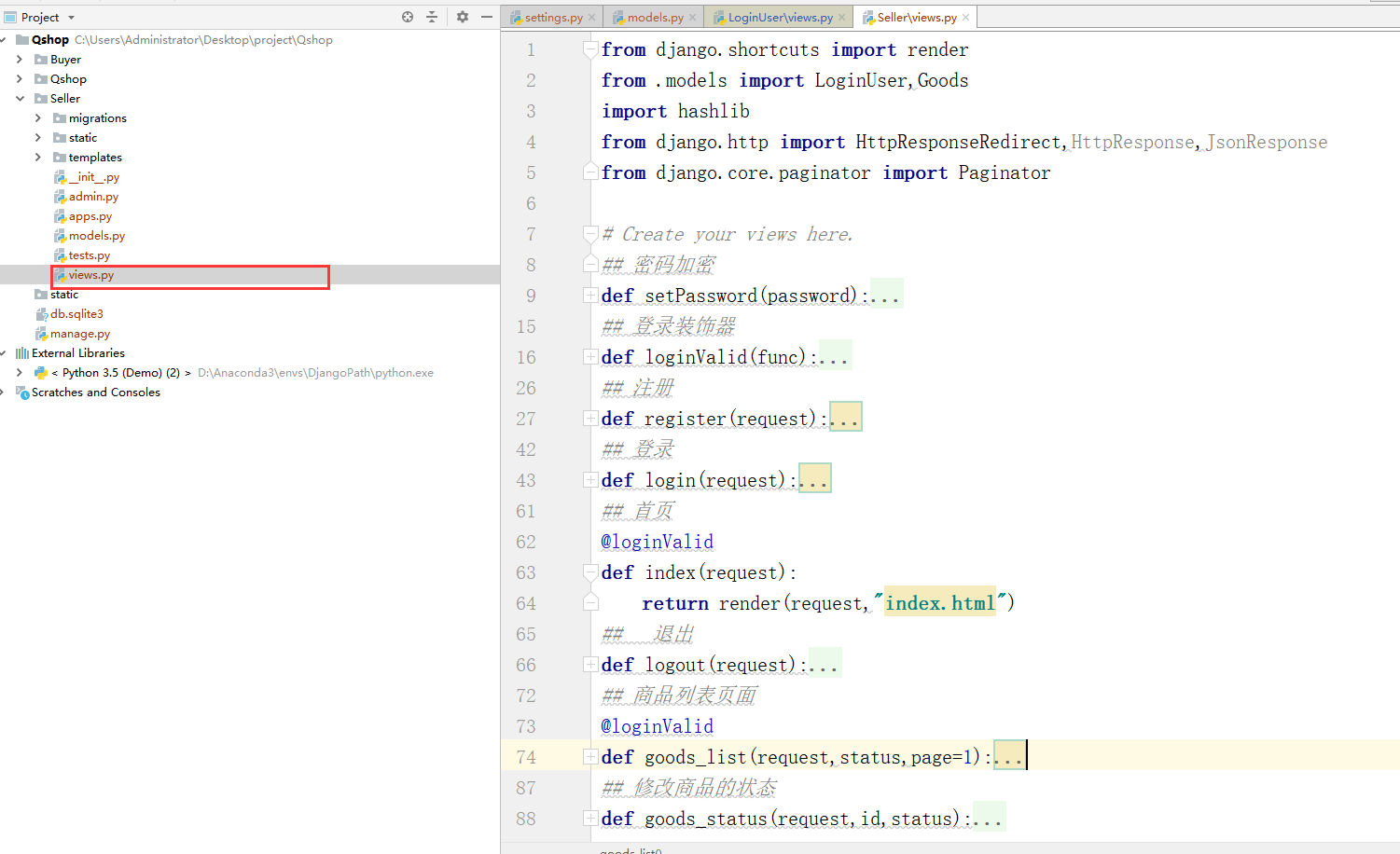
创建数据库

create database qshop charset=utf8;

数据迁移



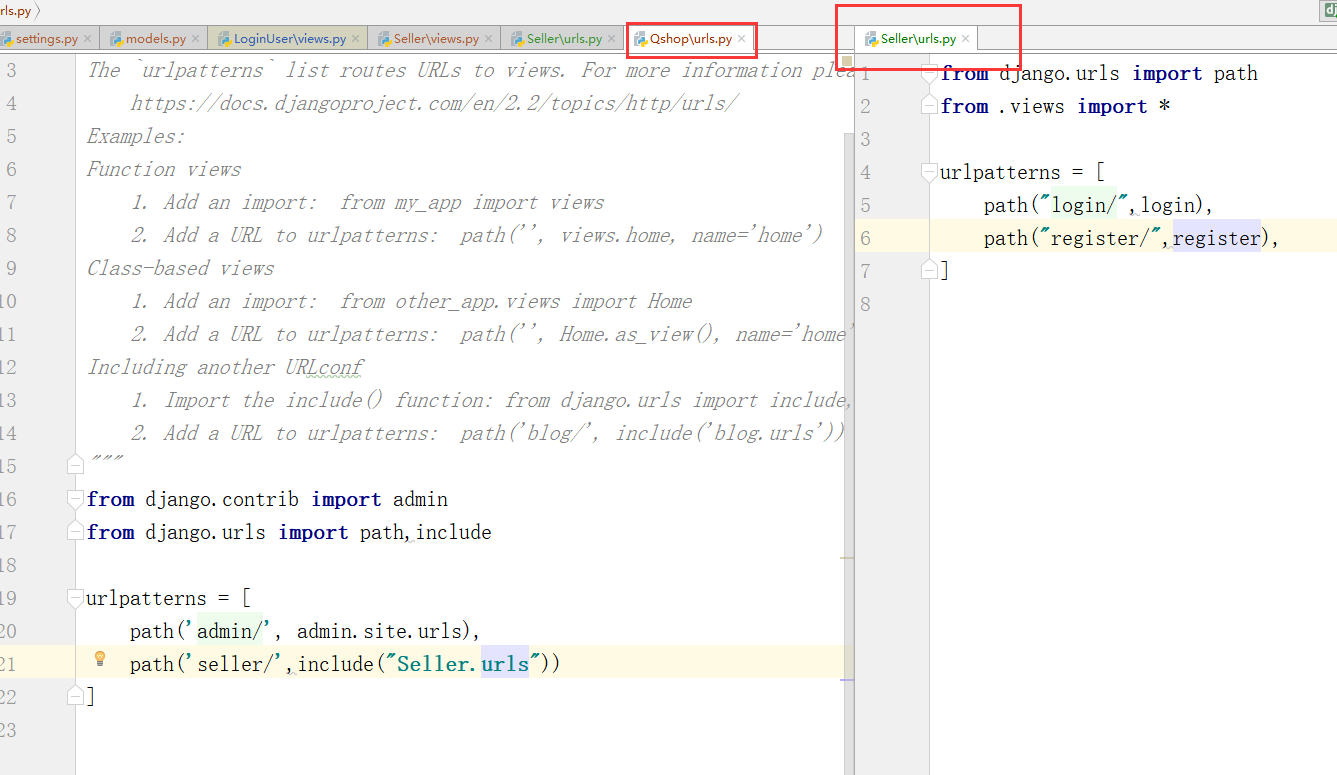
### 复制视图



# 子路由系统

项目，需要进行分模块化管理，如果将所有的路由写在同一个urls.py 中，不方便管理。

1. 在子应用中创建urls.py
2. 在主的urls中使用include，指向子应用的urls



# 子模板系统

目的： 方便管理，将各自的模板文件，归各自模块进行管理，

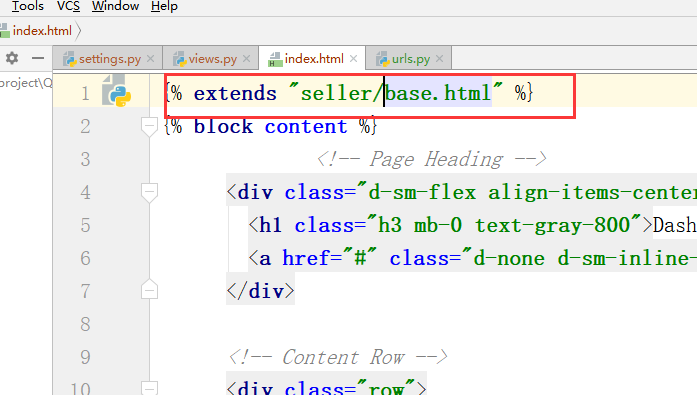
在子应用中创建templates目录，同在templates下面创建以子应用的名字命名的目录（通常情况，使用子应用名字的小写） templates/appname

保证settings中 的 app\_dirs = True

## 模板文件的调用



## 模板文件继承



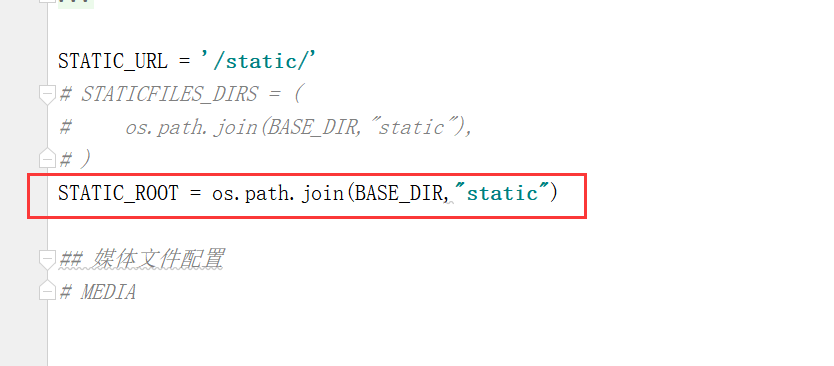
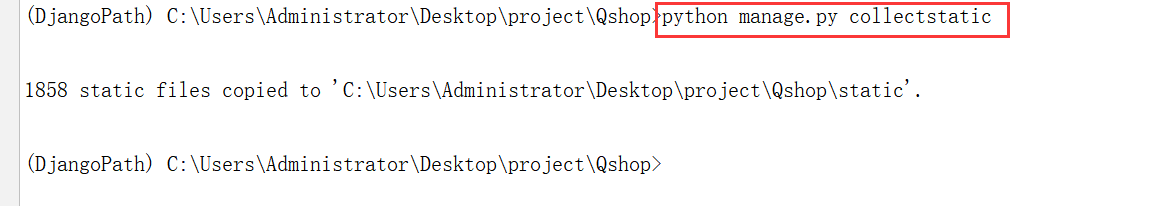
# 子静态系统

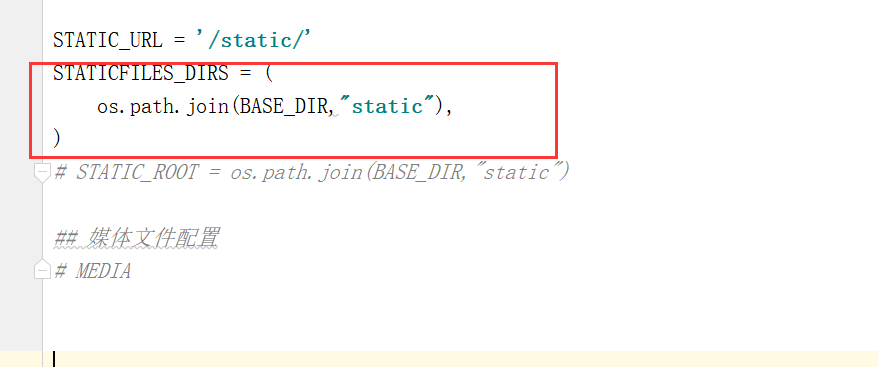
在子应用中创建了一个static目录，static目录下创建了appname（小写）目录，将静态文件放在这个目录下。目的：为了方便管理

在主目录中创建了一个static目录

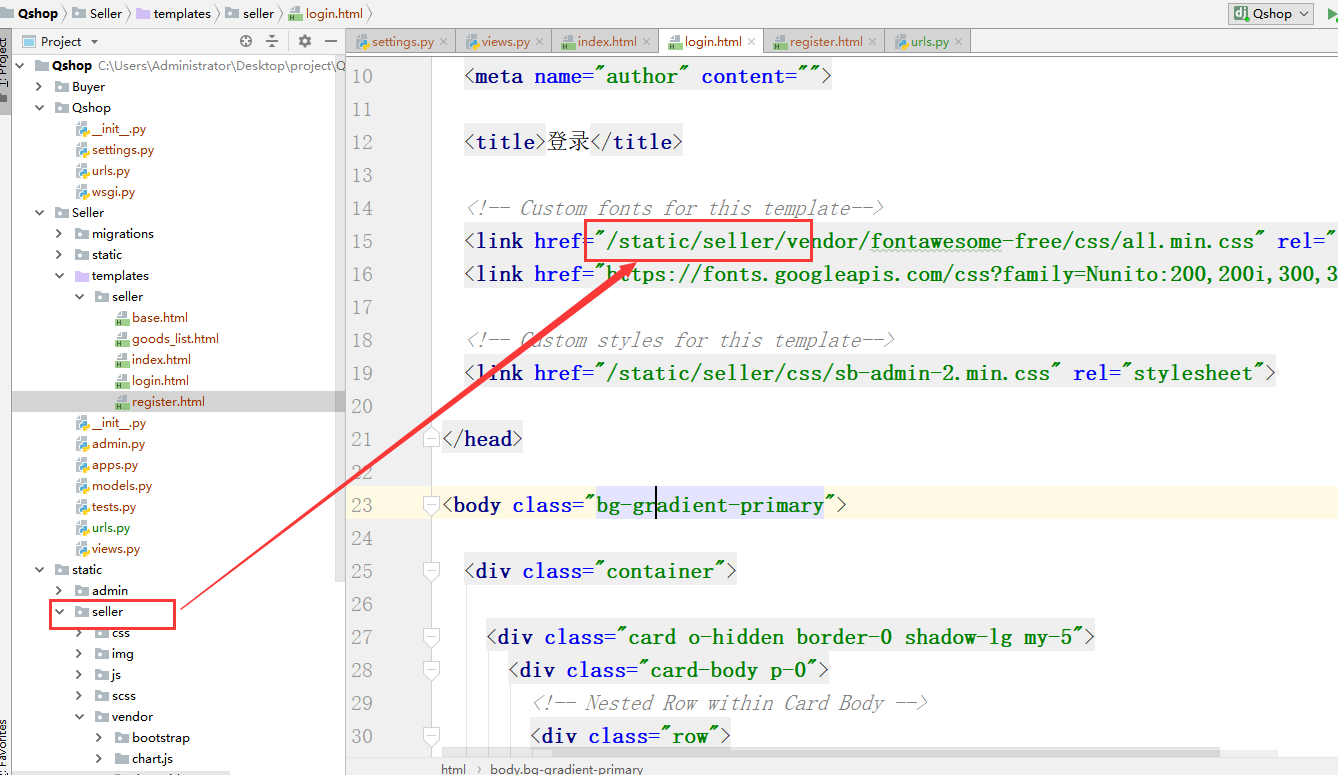
1. 在settings增加了静态文件的配置，
2. 项目在运行的时候使用的静态文件在 主目录的static目录下
3. 收集静态文件，到主目录的static目录下
4. 注意： 当收集完成之后，子应用中的静态文件已经没有作用，但是不要删除，为了不影响下次静态文件的收集。

## 收集静态文件

1. settings中增加配置
   1. STAITC\_ROOT 注意：这个配置跟STAITCFILES\_DIRS 个 媒体文件的配置，冲突
   2. 
2. 收集静态文件
   1. 
3. 复原，将settings文件内容恢复



1. 子静态文件的使用



# 保证项目能够正常运行

视图render 返回页面的路径

路由

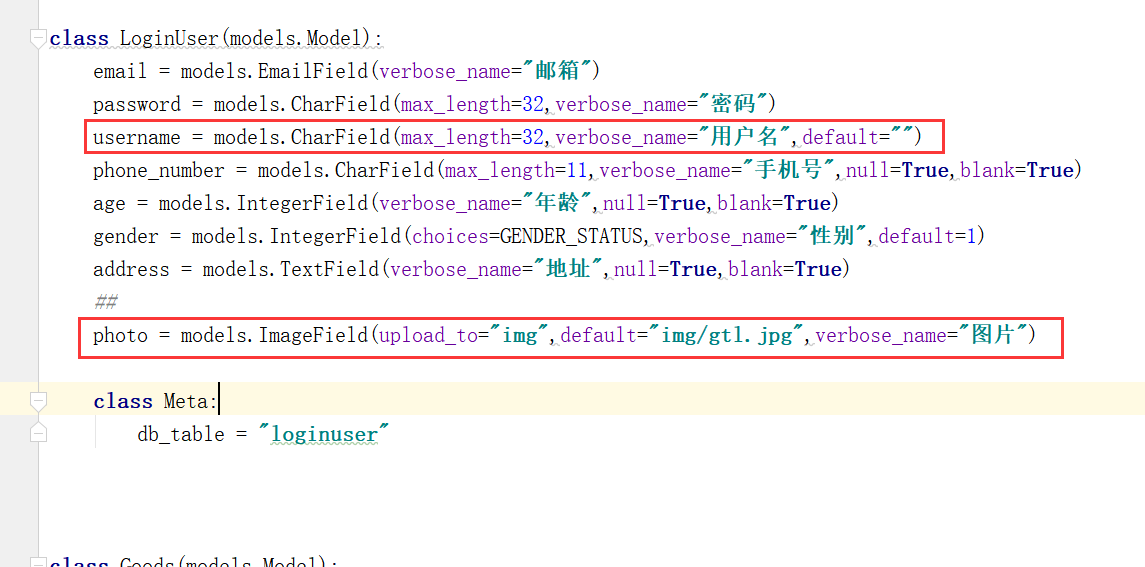
重定向的地址

模板继承的 extends

a标签的地址

# 个人中心页面

增加ImageField字段 增加username



进行数据迁移

## 个人中心静态页面

## 调用静态页面

## 动态数据渲染

视图

*## 个人中心*@loginValid  
**def** user\_profile(request):  
 *## 返回用户的信息  
 ## 从session 或者 cokkie 这种获取登录的用户* userid = request.COOKIES.get(**"userid"**)  
 user = LoginUser.objects.get(id=userid)  
 *## 处理post请求* **if** request.method == **"POST"**:  
 data = request.POST  
 user.email = data.get(**"email"**)  
 user.phone\_number = data.get(**"phone\_number"**)  
 user.username = data.get(**"username"**)  
 user.age = data.get(**"age"**)  
 user.gender = data.get(**"gender"**)  
 user.address = data.get(**"address"**)  
 **if** request.FILES.get(**"img"**):  
 user.photo = request.FILES.get(**"img"**)  
 user.save()  
 **return** render(request,**"seller/user\_profile.html"**,locals())

模板

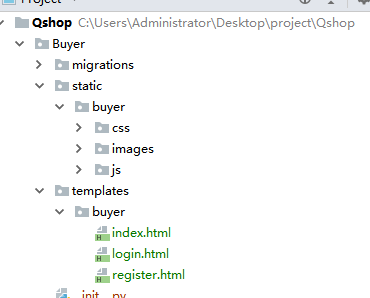
{% **extends "seller/base.html"** %}  
{% **block title** %}  
 个人中心  
{% **endblock** %}  
{% **block content** %}  
 <**h1 class="h3 mb-4 text-gray-800"**>个人中心</**h1**>  
 <**form action="" method="post" enctype="multipart/form-data"**>  
 {% **csrf\_token** %}  
 <**table class="table"**>  
 <**tr**>  
 *<!-- rowspan 跨行-->* <**td rowspan="3" style="width**: 300**px**;**"**>  
 <**img src="/static/**{{ **user**.**photo** }}**" style="height**: 300**px**;**width**: 300**px**;**" alt=""**>  
 <**input type="file" name="img"**>  
 </**td**>  
 <**th**>邮箱</**th**>  
 <**td**>  
 <**input type="text" name="email" value="**{{ **user**.**email** }}**" disabled**>  
 </**td**>  
 <**th**>用户名</**th**>  
 <**td**>  
 <**input type="text" name="username" value="**{{ **user**.**username** }}**"**>  
 </**td**>  
 </**tr**>  
 <**tr**>  
 <**th**>手机号</**th**>  
 <**td**>  
 <**input type="text" name="phone\_number" value="**{{ **user**.**phone\_number** }}**"**>  
 </**td**>  
 <**th**>年龄</**th**>  
 <**td**>  
 <**input type="text" name="age" value="**{{ **user**.**age** }}**"**>  
 </**td**>  
 </**tr**>  
 <**tr**>  
 <**th**>性别</**th**>  
 <**td**>  
*{# <input type="text" name="gender" value="{{ user.gender }}">#}* <**select name="gender" id=""**>  
 <**option value="0"**>女</**option**>  
 <**option value="1"**>男</**option**>  
 <**option value="**{{ **user**.**gender** }}**" selected hidden**>{{ **user**.**get\_gender\_display** }}</**option**>  
 </**select**>  
 </**td**>  
 <**th**>地址</**th**>  
 <**td**>  
 <**input type="text" name="address" value="**{{ **user**.**address** }}**"**>  
 </**td**>  
 </**tr**>  
 <**tr**>  
 <**td colspan="5"**>  
 <**input type="submit" value="完善个人信息" class="btn btn-primary btn-user btn-block"**>  
 </**td**>  
 </**tr**>  
  
  
 </**table**>  
 </**form**>  
  
  
{% **endblock** %}

图片上传

1. imagefield 字段属性
2. form表单
   1. input type = file
   2. <form action="" method="post" enctype="multipart/form-data">
3. 视图：
   1. 获取图片
      1. img =request.FILES.get("img")
   2. 保存
      1. save

# 电商前台 Buyer

## 导入静态文件 导入模板文件



## 收集静态文件

## 还原settings配置文件

## 调用模板文件



## 修改静态文件

<http://127.0.0.1:8000/buyer/index/>

<http://127.0.0.1:8000> 访问到首页

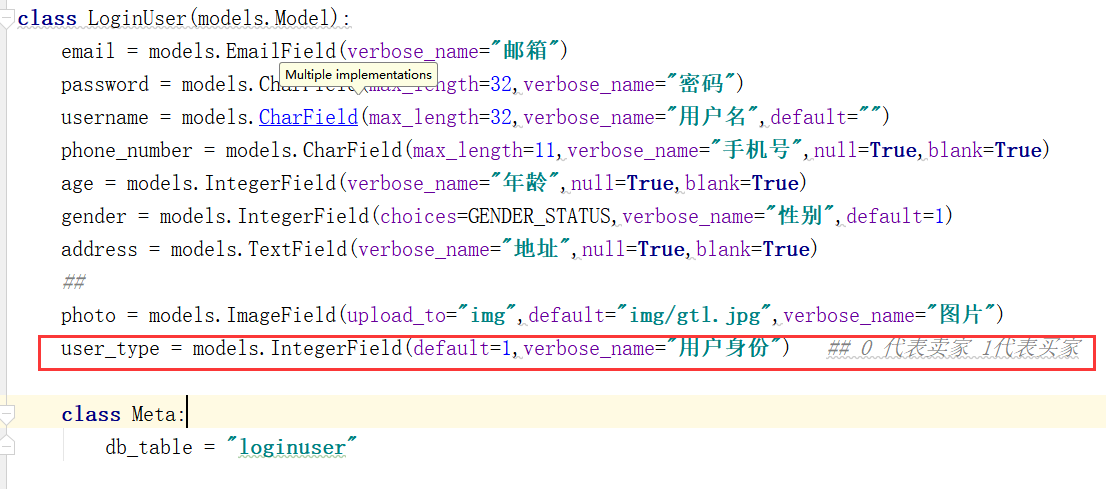
<http://127.0.0.1:8000>/login

<http://127.0.0.1:8000>/register



# 登录注册

增加字段，标识用户的身份



数据迁移

修改seller 注册视图，



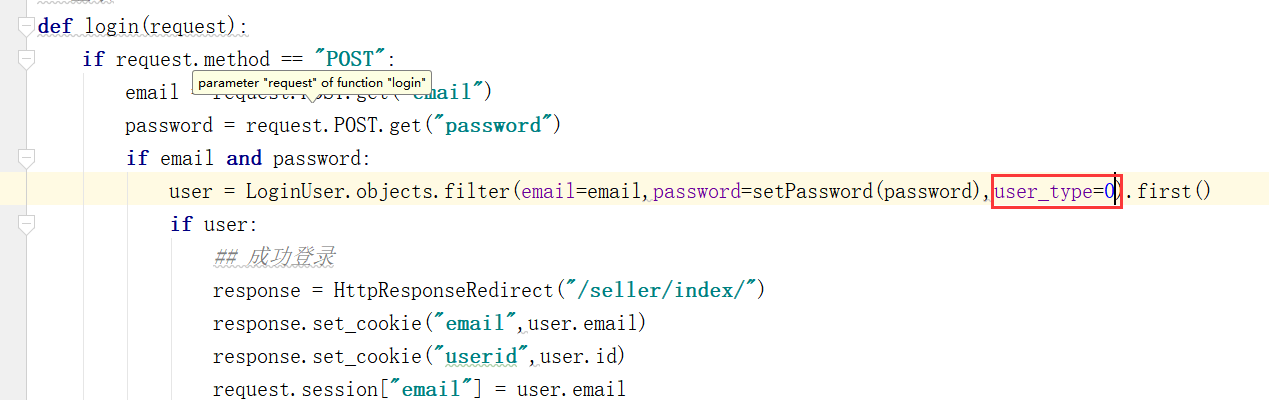
buyer 注册视图

**def** register(request):  
 *## 接收参数* **if** request.method == **"POST"**:  
 username = request.POST.get(**"user\_name"**)  
 password = request.POST.get(**"pwd"**)  
 repassword = request.POST.get(**"cpwd"**)  
 email = request.POST.get(**"email"**)  
 *## 校验数据* **if** email **and** password **and** password == repassword:  
 *## 有数据* LoginUser.objects.create(email=email,password=setPassword(password),username=username)  
 **return** HttpResponseRedirect(**"/login/"**)  
 **else**:  
 *## 参数为空* message = **"参数为空"  
  
 return** render(request,**"buyer/register.html"**)

buyer 登录视图

**def** login(request):  
 **if** request.method == **"POST"**:  
 username = request.POST.get(**"username"**)  
 password = request.POST.get(**"pwd"**)  
 **if** username **and** password:  
 user = LoginUser.objects.filter(username=username,password=setPassword(password),user\_type=1).first()  
 **if** user:  
 *## 成功登录* response = HttpResponseRedirect(**"/"**)  
 response.set\_cookie(**"email"**,user.email)  
 response.set\_cookie(**"username"**,user.username)  
 response.set\_cookie(**"userid"**,user.id)  
 request.session[**"email"**] = user.email  
 **return** response  
 **else**:  
 message = **"账号密码不正确"  
 else**:  
 message = **"参数为空"  
  
 return** render(request,**"buyer/login.html"**,locals())

seller登录视图



登出



视图：

**def** logout(request):  
 resp = HttpResponseRedirect(**"/login/"**)  
 resp.delete\_cookie(**"email"**)  
 resp.delete\_cookie(**"username"**)  
 resp.delete\_cookie(**"userid"**)  
 **del** request.session[**"email"**]  
 **return** resp

后端和前台 cookie 和session 冲突

在设置cookie的时候，修改cookie和session的名字