链滴

python 框架 Django 实战商城项目之用户模块创建

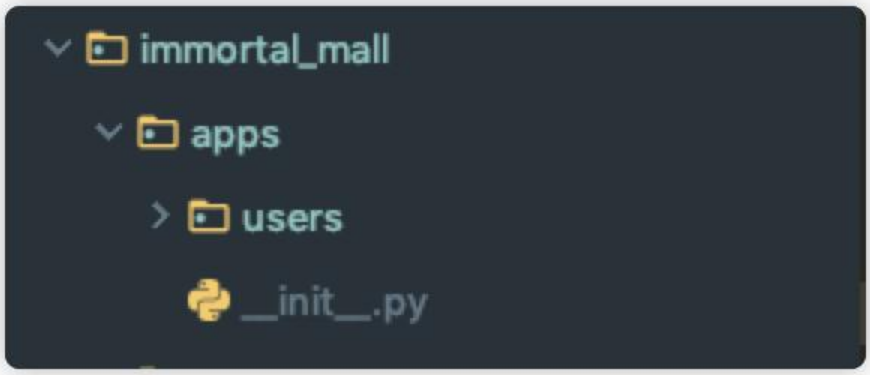
作者：zyjImmortal

原文链接：https://ld246.com/article/1583734683398

来源网站：链滴

创建用户APP

整个项目会存在多个应用，需要存放在一个单独的文件包了，所以新建一个apps目录，管理所有子应
。



在apps包目录下穿件users应用

```
python ../../manage.py startapp users
```

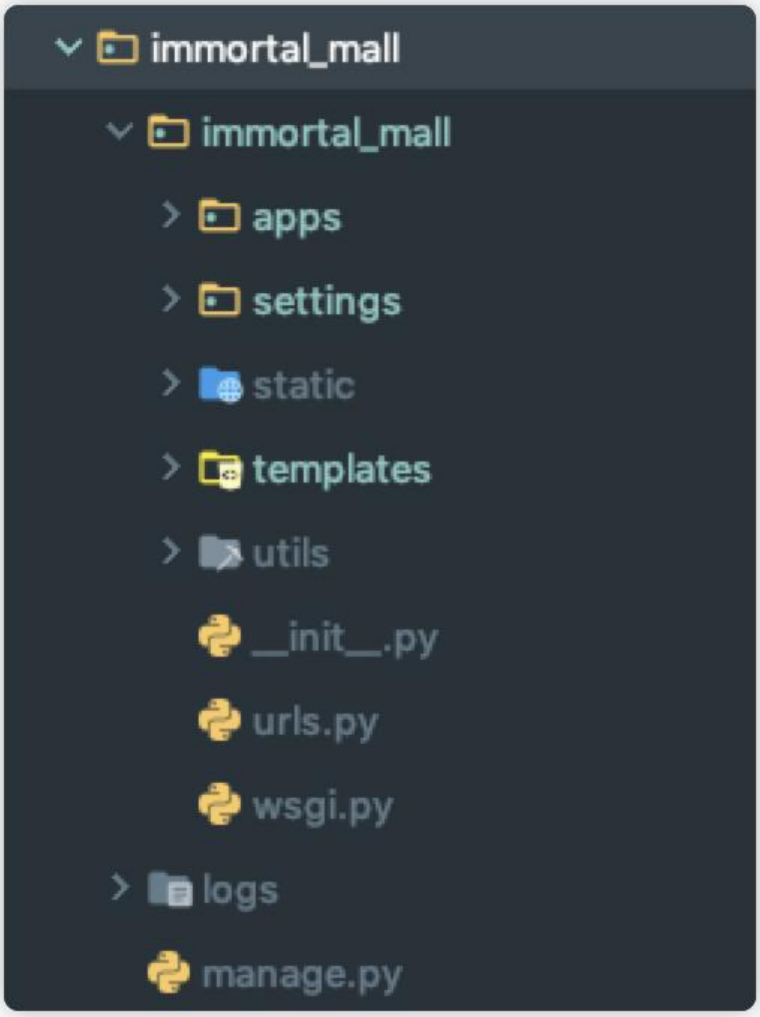
这个时候呢，需要我们将新创建的应用注册到django里，可是这里我们修改了应用的管理目录,与默的方式不同，如果还按照之前的方式注册APP肯定会报错，这个时候我们可以先查看一下django的导路径，在dev文件中输入

```
print(sys.path()) # 输出包的所有搜索路径

['/Users/xxxx/workspace/xxxx/mall/immortal_mall',
 '/Users/xxxx/workspace/xxxx/mall',
 '/Users/xxxx/workspace/xxxx/mall/venv/lib/python38.zip',
 '/Users/xxxx/workspace/xxxx/mall/venv/lib/python3.8',
```

```
 '/Users/xxxx/workspace/xxxx/mall/venv/lib/python3.8/lib-dynload',
 '/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8',
 '/Users/xxxx/workspace/xxxx/mall/venv/lib/python3.8/site-packages',
 '/Applications/PyCharm.app/Contents/helpers/pycharm_matplotlib_backend']
```

第一个路径就是我们django项目的主目录，



也就是说他会搜索主目录下的所有包，那么就可以定义APP的路径为

```
meiduo_mall.apps.users
```

这个时候运行程序，是可以运行成功的。但是呢，这样定义注册APP的方式太麻烦了，如果应用多的，每个都要这样写一遍，那不得烦死。所以必须得简化。怎么简化，直接往包的搜索路径中插入apps录的绝对路径，那不django可以搜索到了么。

```
sys.path.insert(0, os.path.join(BASE_DIR, 'apps'))
```

就这样完事儿，然后注册APP

```
INSTALLED_APPS = [
 'django.contrib.admin',
 'django.contrib.auth',
 'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # 'immortal_mall.apps.users',
    'users'
]
```

返回注册页面

准备注册所使用的的模板，放入预先新建好的templates文件夹中

定义用户注册视图类：

```
class RegisterView(View):
    """用户注册视图类"""

    def get(self, request):
        '''获取注册页面'''
        return render(request, 'register.html')
```

定义用户注册路由

```
# 总路由
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(('users.urls', 'users'), namespace='users'))
]
```

这里要划重点了，include函数的第一个参数是一个元祖，第一个参数没得说，就是指定了子应用的由，第二个参数是app_name，这里必须制定app_name，如果不指定这个参数，写成include('users.rls', namespace='users')是会报错的。

当然还有一种指定方式就是在子应用的urls文件中中指定app_name='users'.

在users应用目录下新建一个urls.py文件，然后写入路由信息

```
urlpatterns = [
    path('register/', views.RegisterView.as_view(), name='register')  # name添加命名空间
]
```

启动应用，浏览器请求http://127.0.0.1:8989/register/，返回注册页面。

用户模型类

项目使用的是django自带的用户认证系统，先来了解一下都有哪些功能。

Django默认用户认证系统

django自带用户认证系统可以处理用户账号、组、权限以及基于cookie的用户会话，位于django.contrib.auth

包中。

auth包是django内置的一个APP，和admin一样，可以同时处理认证和授权，认证就是验证一个用是不是系统的人，授权决定了一个认证的用户可以被允许做什么。

Django认证系统中提供了用户模型类User保存用户的数据，User对象是认证系统的核心**：**

```
class User(AbstractUser):
    """
    Users within the Django authentication system are represented by this
    model.

    Username and password are required. Other fields are optional.
    """
    class Meta(AbstractUser.Meta):
        swappable = 'AUTH_USER_MODEL'
```

Userl类没什么，看看父类AbstractUser中的东西，里面定义用户的一些字段，里面包括user类一些填的字段username、password，还有其他一些非必填的字段，is_active,is_staff等，关于用户认证方法都AbstractUser的父类AbstractBaseUser中，

不过AbstractUser类中持有了UserManager的实例叫做objects，这个类提供了创建用户的方法，比：

```
user = User.objects.create_user(username, email, password, **extra_fields)
```

自定义用户模型类

图片

这是用户注册信息表单，有一个手机号的字段，但是Django提供放入用户模型中是没有这个字段的需要我们自己定义。

```
class User(AbstractUser):
    """自定义用户模型类"""
    mobile = models.CharField(max_length=11, unique=True, verbose_name="手机号")

    class Meta:
        db_table = 'tb_user' # 自定义表名
        verbose_name = "用户"  # 站点显示
        verbose_name_plural = verbose_name # 复数显示
```

自定义的用户模型类需要继承AbstractUser类，然后指定新添加的字段。添加完后运行项目，会报一错：

```
  File "/Users/zhouyajun/workspace/boxuegu/meiduo/venv/lib/python3.8/site-packages/django/core/management/base.py", line 441, in check
    raise SystemCheckError(msg)
django.core.management.base.SystemCheckError: SystemCheckError: System check identified some issues:

ERRORS:
auth.User.groups: (fields.E304) Reverse accessor for 'User.groups' clashes with reverse accessor for 'User.groups'.
    HINT: Add or change a related_name argument to the definition for 'User.groups' or 'User.groups'.
auth.User.user_permissions: (fields.E304) Reverse accessor for 'User.user_permissions' clashes with reverse accessor for 'User.user_permissions'.
    HINT: Add or change a related_name argument to the definition for 'User.user_permissions' or 'User.user_permissions'.
users.User.groups: (fields.E304) Reverse accessor for 'User.groups' clashes with reverse accessor for 'User.groups'.
    HINT: Add or change a related_name argument to the definition for 'User.groups' or 'User.groups'.
users.User.user_permissions: (fields.E304) Reverse accessor for 'User.user_permissions' clashes with reverse accessor for 'User.user_permissions'.
    HINT: Add or change a related_name argument to the definition for 'User.user_permissions' or 'User.user_permissions'.

System check identified 4 issues (0 silenced).
```

这是django系统默认的的认证对象配置，我们使用了自定义的对象，但是这个对象没有被指定给系统，

```
##################
# AUTHENTICATION #
##################

AUTH_USER_MODEL = 'auth.User'

AUTHENTICATION_BACKENDS = ['django.contrib.auth.backends.ModelBackend']

LOGIN_URL = '/accounts/login/'

LOGIN_REDIRECT_URL = '/accounts/profile/'

LOGOUT_REDIRECT_URL = None

# The number of days a password reset link is valid for
PASSWORD_RESET_TIMEOUT_DAYS = 3
```

所以要在自己的dev配置文件中重新指定

AUTH_USER_MODEL = 'users.User'

然后创建迁移文件，执行迁移命令，完成表的创建。

 python manage.py makemigrations

 python manage.py migrate