

# 蒲公英（虚拟局域网操作）

- 1. 下载蒲公英
- 2. 登陆



### 3. 进入页面



bogon

172.16.1.252 本机



51568459:002

172.16.1.83



所属网络 xgl1129的网络

#### 4. 进入控制台

蒲公英 | 网络管理平台

主题模式 APP下载 代购 客服中心 意见反馈 消息 xgl1129

概览

星云 beta

成员管理

监控平台 提示

告警管理

SIM管理

日志中心

零信任

智能网络

服务配置

配置中心

服务管理

蒲公英体验版

超级成员 1/3 已使用 / 总

访问端成员 0/0 已使用 / 总

组网列表

xgl1129的网络 默认 对等网络 标准线路

硬件成员 0 在线 / 总

访问端成员 1/1 在线 / 总

查看详情 组网可视化 更多

增值服务

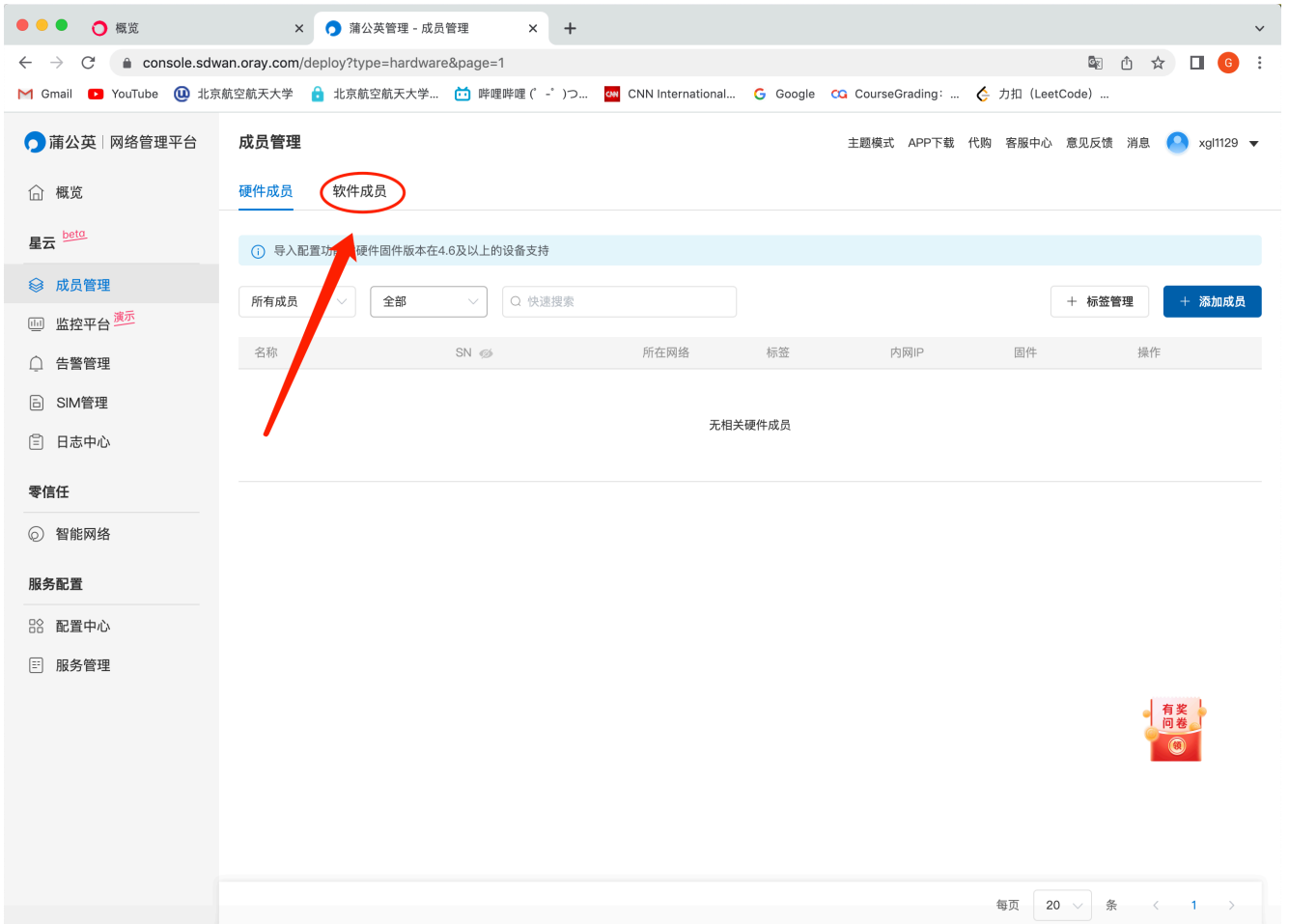
自定义IP 可修改组网内软件成员的IP地址，并且固定使用

带宽加速 提供网络关键节点备用带宽，可自主分配

夜间带宽加速 在20:00pm至08:00am期间提升组网标准转发速度

软件定制 支持客户端个性化定制企业私有品牌

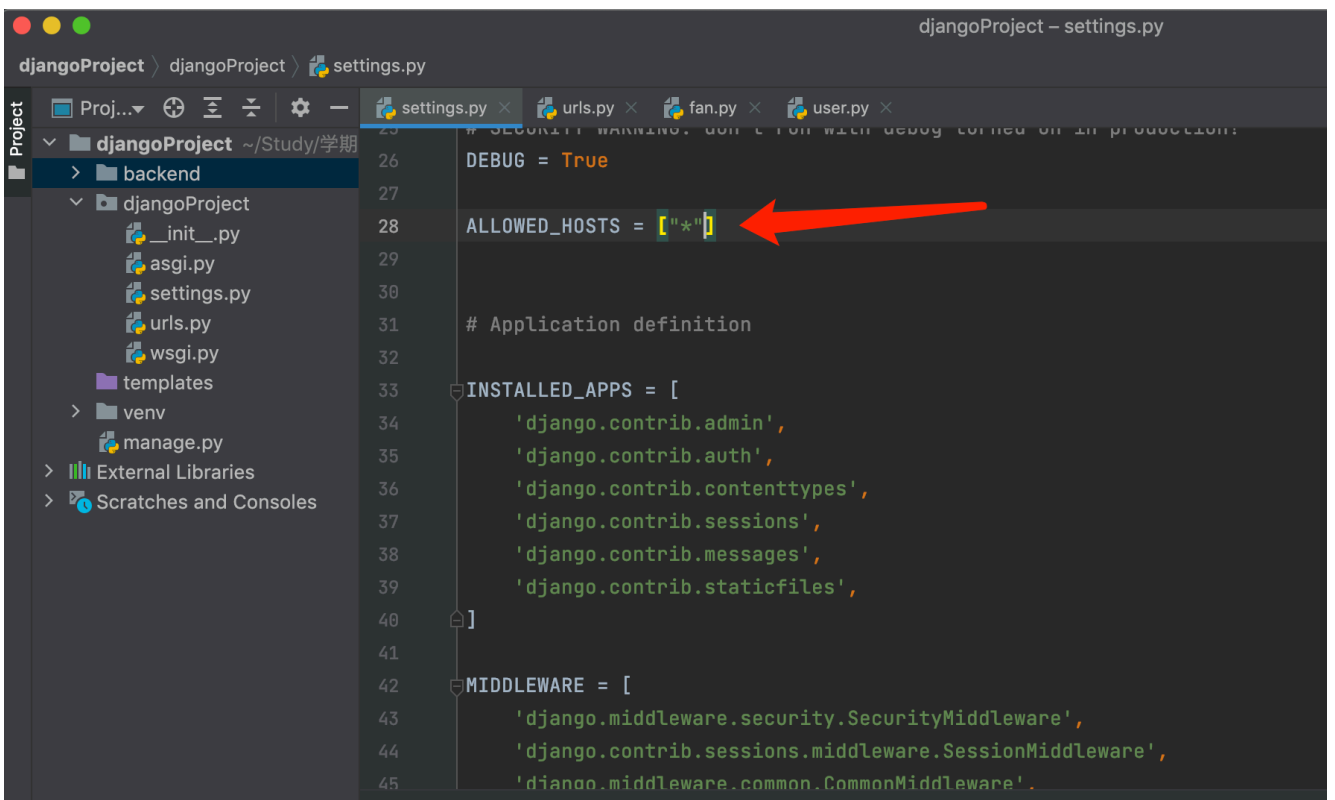
有奖问卷



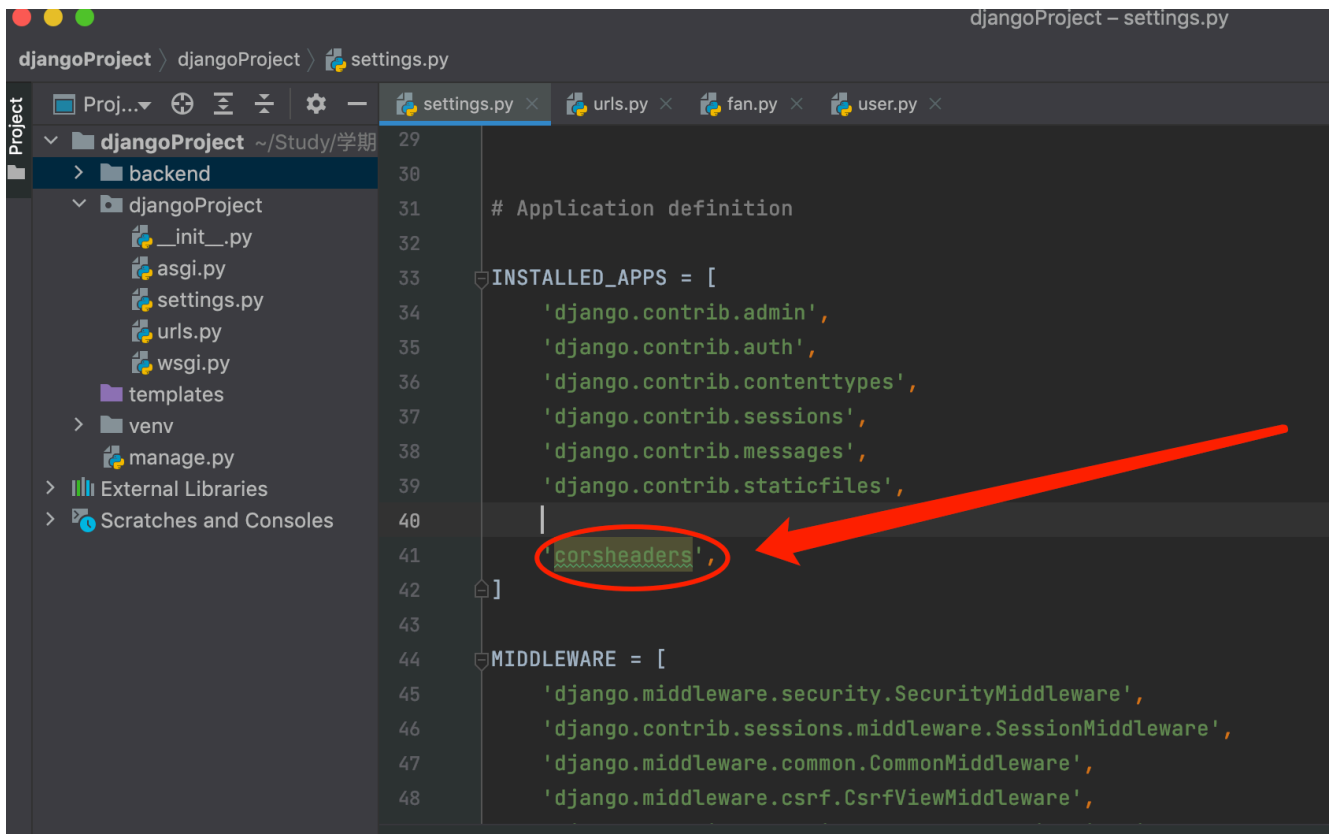
5. 只要登陆同一个账号密码，就可以自动在同一个虚拟局域网。

## 后端setting改动

1.



2.



```
djangoProject - settings.py
Project
  djangoProject ~/Study/学期
    backend
      djangoProject
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
        templates
      venv
        manage.py
    External Libraries
    Scratches and Consoles
  settings.py x
  urls.py x
  fan.py x
  user.py x
  29
  30
  31 # Application definition
  32
  33 INSTALLED_APPS = [
  34     'django.contrib.admin',
  35     'django.contrib.auth',
  36     'django.contrib.contenttypes',
  37     'django.contrib.sessions',
  38     'django.contrib.messages',
  39     'django.contrib.staticfiles',
  40
  41     'corsheaders',
  42 ]
  43
  44 MIDDLEWARE = [
  45     'django.middleware.security.SecurityMiddleware',
  46     'django.contrib.sessions.middleware.SessionMiddleware',
  47     'django.middleware.common.CommonMiddleware',
  48     'django.middleware.csrf.CsrfViewMiddleware',
```

加入 'corsheaders',

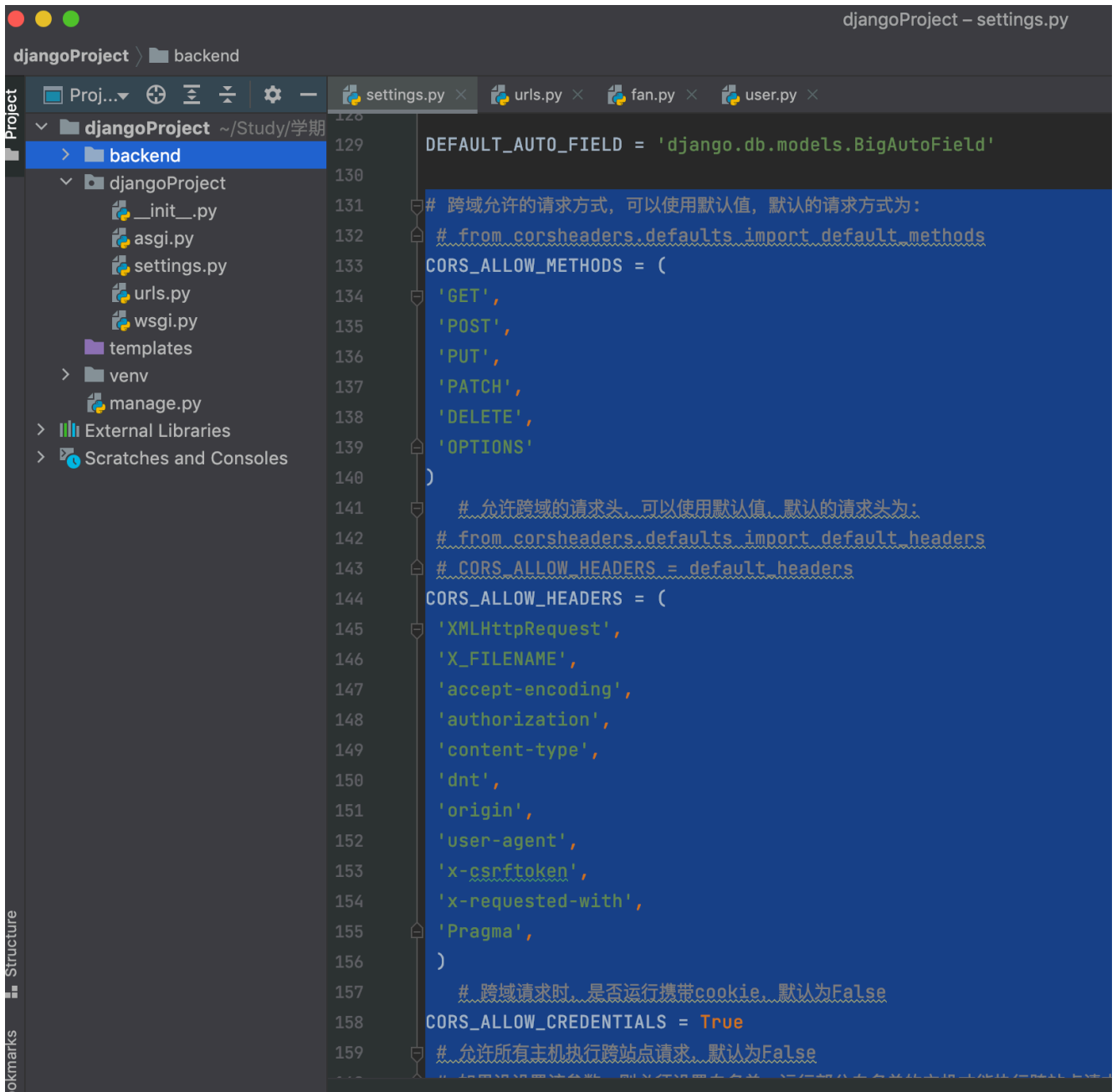


```
3
4 MIDDLEWARE = [
5     'django.middleware.security.SecurityMiddleware',
6     'django.contrib.sessions.middleware.SessionMiddleware',
7     'corsheaders.middleware.CorsMiddleware',
8     'django.middleware.common.CommonMiddleware',
9     # 'django.middleware.csrf.CsrfViewMiddleware',
10    'django.contrib.auth.middleware.AuthenticationMiddleware',
11    'django.contrib.messages.middleware.MessageMiddleware',
12    'django.middleware.clickjacking.XFrameOptionsMiddleware',
13 ]
14
15 ROOT_URLCONF = 'djangoProject.urls'
```

第三行加入

第五行注释

3. 第三行加入 'corsheaders.middleware.CorsMiddleware',  
第五行注释



```
128 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
129
130
131 # 跨域允许的请求方式，可以使用默认值，默认的请求方式为：
132 # from corsheaders.defaults import default_methods
133 CORS_ALLOW_METHODS = (
134     'GET',
135     'POST',
136     'PUT',
137     'PATCH',
138     'DELETE',
139     'OPTIONS'
140 )
141
142 # 允许跨域的请求头，可以使用默认值，默认的请求头为：
143 # from corsheaders.defaults import default_headers
144 # CORS_ALLOW_HEADERS = default_headers
145 CORS_ALLOW_HEADERS = (
146     'XMLHttpRequest',
147     'X_FILENAME',
148     'accept-encoding',
149     'authorization',
150     'content-type',
151     'dnt',
152     'origin',
153     'user-agent',
154     'x-csrftoken',
155     'x-requested-with',
156     'Pragma',
157 )
158
159 # 跨域请求时，是否运行携带cookie，默认为False
160 CORS_ALLOW_CREDENTIALS = True
161
162 # 允许所有主机执行跨站点请求，默认为False
```

4. 在文件末尾加入一下代码：

```
# 跨域允许的请求方式，可以使用默认值，默认的请求方式为：
# from corsheaders.defaults import default_methods
CORS_ALLOW_METHODS = (
    'GET',
    'POST',
    'PUT',
    'PATCH',
    'DELETE',
    'OPTIONS'
)

# 允许跨域的请求头，可以使用默认值，默认的请求头为：
# from corsheaders.defaults import default_headers
# CORS_ALLOW_HEADERS = default_headers
CORS_ALLOW_HEADERS = (
```

```
'XMLHttpRequest',
'X_FILENAME',
'accept-encoding',
'authorization',
'content-type',
'dnt',
'origin',
'user-agent',
'x-csrf-token',
'x-requested-with',
'Pragma',
)
# 跨域请求时, 是否运行携带cookie, 默认为False
CORS_ALLOW_CREDENTIALS = True
# 允许所有主机执行跨站点请求, 默认为False
# 如果没设置该参数, 则必须设置白名单, 运行部分白名单的主机才能执行跨站点请求
CORS_ORIGIN_ALLOW_ALL = True
```

## 5. 运行指令

运行指令是

```
python manage.py runserver 0.0.0.0:8000
```