

L1 实验手册

L1 实验是基于蓝鲸开发者中心社区版实现简单的微信朋友圈应用，实现的效果和预习实验相似。首先需要先在开发者中心创建 SaaS 应用并搭建本地开发环境。

示例 DEMO：[基于蓝鲸开发者中心社区版实现简单的微信朋友圈应用](#)(此 DEMO 中用户访问 User/Post 页面如果报错【5000，系统异常,请联系管理员处理】，是符合预期的，需要应用管理员手动在 admin 页面添加用户为 WeChatUser 即可)

0. 蓝鲸开发者中心创建 SaaS 应用

- 前置知识：[手把手搭建蓝鲸开发框架环境-Windows 系统](#)
- 进入社区版蓝鲸开发者中心：<https://bkpaas.ce.bktencent.com/>，点击创建应用
- 应用类型：选择云原生应用；代码来源：选择代码仓库
- 填写应用 ID、应用名称
- 应用模板：选择 蓝鲸应用开发框架（Django3.x）
- 新建 gitee 仓库，并将 git 地址（<https://gitee.com/xxxuser/xxxproj.git>）填进源代码地址（如果你创建的 gitee 仓库是公开的，下方的 git 账号密码可以随便填）
- 点击下一步，然后点击创建应用
- 在创建成功界面下载初始化模板
- 按照页面提示初始化本地代码仓库（类似于下面命令行，具体操作根据实际显示命令），git 的安装和详细使用参考 [手把手搭建蓝鲸开发框架环境-Windows 系统](#) 文档

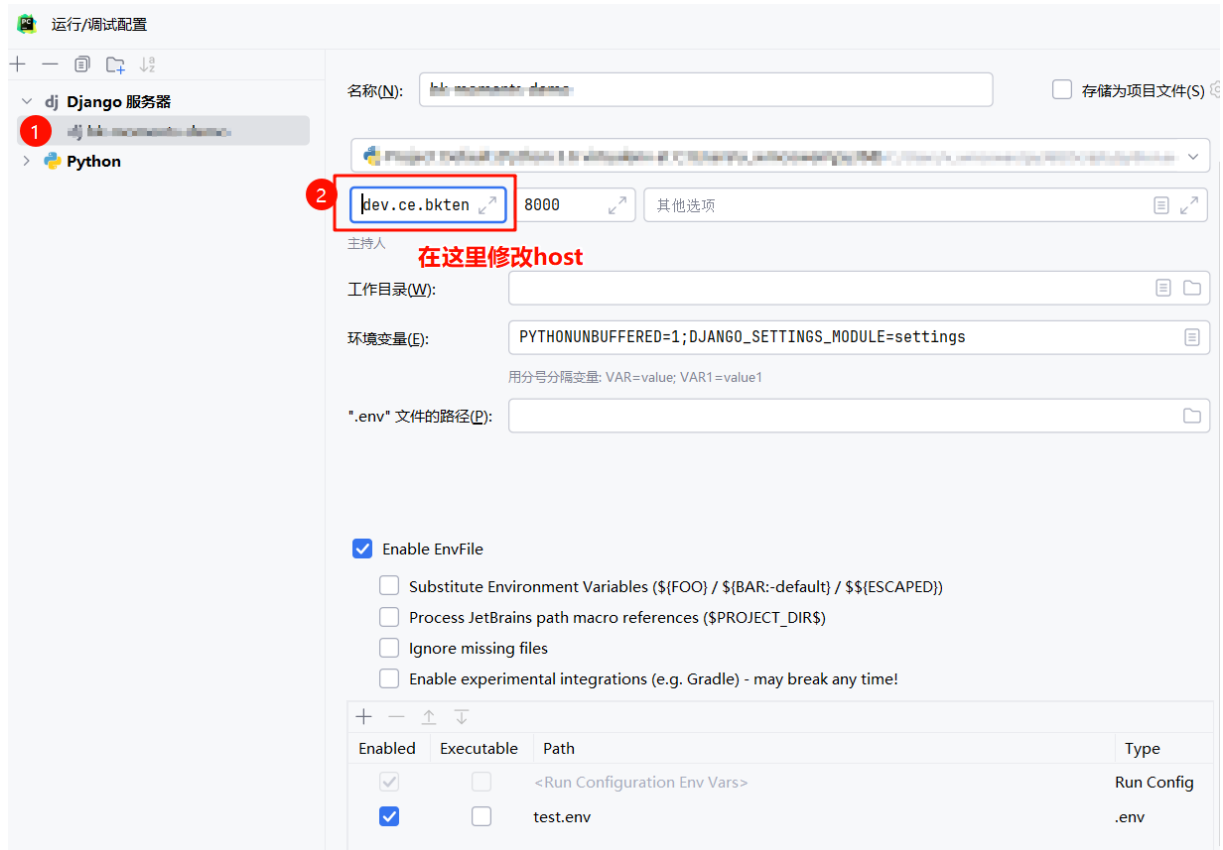
```
# 下载并解压代码到本地目录
mkdir wechat-saas
curl "https://bkrepo.ce.bktencent.com/generic/xxx" > wechat-
saas.tar.gz
tar -xf wechat-saas.tar.gz -C wechat-saas

# 添加远程仓库地址并完成推送
cd wechat-saas
git init
git add .
git commit -m "init repo"
git remote add origin https://gitee.com/xxxuser/xxxproj.git
git push -u origin main
```

- 本地使用 Pycharm 打开项目文件夹，操作系统 hosts 文件中添加如下配置

```
127.0.0.1 dev.ce.bktencent.com
```

- 注意 PyCharm 中 Django 运行配置的 host 需要设置为 dev.ce.bktencent.com，还需要在下方启用 EnvFile 并添加相关的.env 文件



下载下来的代码与我们的实验环境要求有差异，所以需要修改 runtime.txt 文件和 requirements.txt 文件：

- runtime.txt 修改为如下内容

```
# -*- coding: utf-8 -*-
# 工程预装模块
# 需要额外的 python 包，可直接在文件后面添加
# 请确保指定的包和版本号，可通过 pip 安装

# blueapps requirement

redis==3.5.3
Django==3.2.4
celery==4.4.0
eventlet==0.31.0
greenlet==1.1.3
django-celery-beat==2.2.0
django-celery-results==2.0.1
django-cors-headers==3.7.0

# jwt
pyCryptodome==3.9.7
PyJWT==1.7.1
cryptography==2.7
mysqlclient==2.0.3
MarkupSafe==1.1.1
Mako==1.1.4
requests==2.25.1
python-json-logger==2.0.1
whitenoise==5.2.0
gunicorn==19.6.0

blueapps==4.8.0

pymysql==1.0.2
django-rest-framework==3.15.1
drf-yasg==1.21.8
```

1. 设置数据库

- 本实验需要使用 MySQL 数据库，需要先安装 MySQL 且配置好账号密码
- `config/dev.py` 设置数据库密码

2. 添加 moments 应用

- 创建 moments app
 - PyCharm 添加 python 运行配置，脚本选择 `manage.py`，参数填写 `startapp moments`，注意需要选择环境变量，运行该配置创建 app
- `config.py/default.py` 下的 `INSTALLED_APPS` 添加 moments

3. model.py

- `moments/models.py` 添加如下 model 代码
- 其中 `WeChatUser` 是蓝鲸开发框架用户模型的拓展，使用 `from blueapps.account.models import User` 引入 User 模型

```

from django.db import models
from blueapps.account.models import User

class WeChatUser(models.Model):
    user = models.OneToOneField(User, models.CASCADE)
    email = models.EmailField(unique=True, null=True)
    motto = models.CharField(max_length=100, null=True,
blank=True)
    pic = models.CharField(max_length=50, null=True, blank=True)
    region = models.CharField(max_length=50, null=True,
blank=True)

    def __str__(self):
        return self.user.username

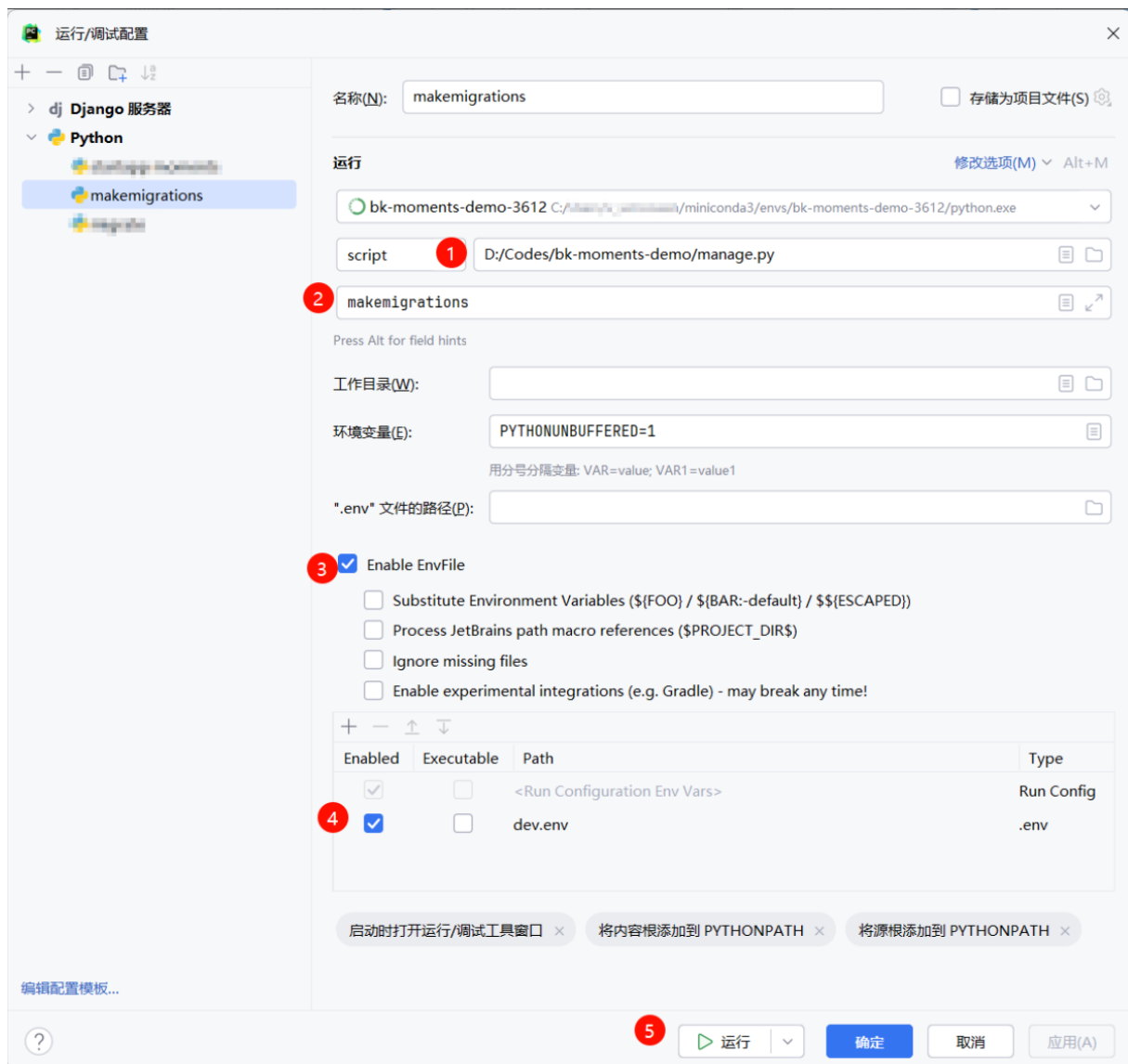
class Status(models.Model):
    user = models.ForeignKey(WeChatUser, models.CASCADE)
    text = models.CharField(max_length=280)
    pics = models.CharField(max_length=100, null=True,
blank=True)
    pub_time = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.text

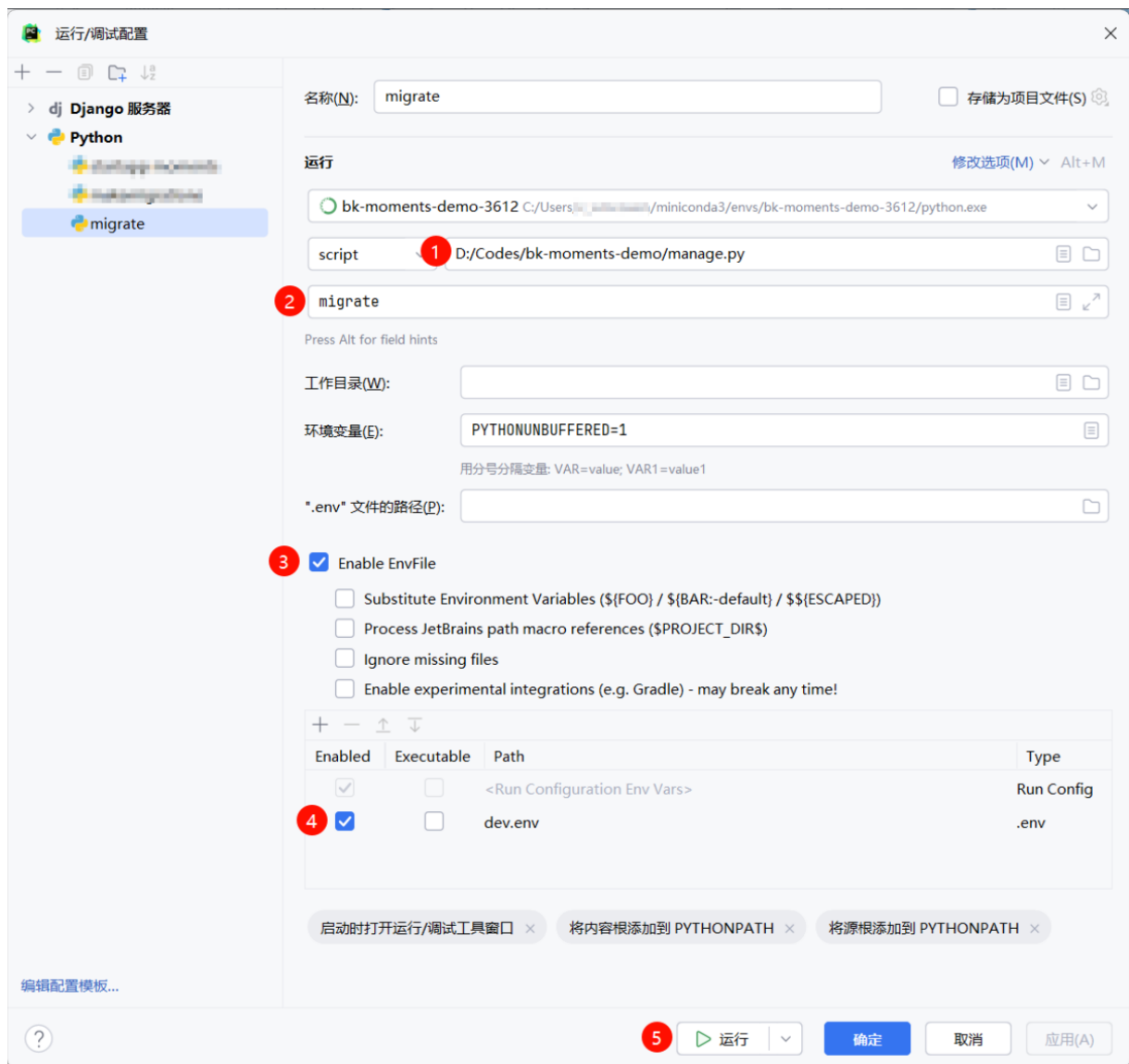
class Meta:
    ordering = ['-id']

```

- 执行 `python manage.py makemigrations` 指令生成数据库迁移脚本（因为执行 `manage.py` 脚本命令需要读取环境变量，所以不要通过终端的命令执行，而是通过 PyCharm 的运行配置去执行，下同）
- 记得要按照**手把手搭建蓝鲸开发框架环境-Windows 系统**的指引在本地项目根目录下创建 `dev.env` 文件。



- 执行 `python manage.py migrate` 变更数据库



- 上面两个命令使用 pycharm 的运行配置，注意添加环境变量，需要确保 config/dev.py 中配置的数据库存在。如数据库不存在，可以使用下面的 SQL 语句创建


```
CREATE DATABASE `... ..` default charset utf8 COLLATE  
utf8_general_ci;
```

4. template 静态文件

- 复制实验材料中的 static 和 templates 到 moments 目录中
- 删除项目根路径下的 templates 文件中的模版文件，防止和 moments 模版冲突

5. views 视图文件

- 默认需要登录态，不需要 login_required

```
from .models import WeChatUser, Status
from django.shortcuts import redirect, render


def home(request):
    return render(request, 'homepage.html')


def show_user(request):
    # 获取蓝鲸用户 id
    user_id = request.user.id
    # 获取 WeChatUser 对象
    wechat_user = WeChatUser.objects.get(user_id=user_id)
    return render(request, 'user.html', {'user': wechat_user})


def show_status(request):
    statuses = Status.objects.all()
    return render(request, 'status.html', {'statuses': statuses})


def submit_post(request):
    user = WeChatUser.objects.get(user=request.user)
    text = request.POST.get('text')
    if text:
        status = Status(user=user, text=text)
        status.save()
        return redirect('/status')
    return render(request, 'my_post.html')
```

6. 设置 urls.py

- 修改项目根目录下的 `urls.py` 为如下

```
from django.contrib import admin
from django.urls import path
from moments.views import home, show_user, show_status,
submit_post
from django.contrib.auth.views import LogoutView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', home),
    path('user/', show_user),
    path('status/', show_status),
    path('post/', submit_post),
]
```

- 此时本地运行项目可以发现，/post 页面是会出错的，因为还没添加 WeChatUser 用户，需要在数据库中创建用户表项

7. 添加 admin

- 设置特定用户为超级管理员，文件 config/default.py 中

```
INIT_SUPERUSER = ['设置成自己用户名']
```

- 由于不是第一次部署，需要设置路由函数手动指定用户为超级管理员，参考 [文档中心 | 蓝鲸智云](#)
- moments/views.py 添加如下路由函数

```
from django.conf import settings
from django.http import HttpResponse
from blueapps.account import get_user_model

def set_super_user(request):
    """
    添加用户为管理员
    """
    user = get_user_model()
    for name in settings.INIT_SUPERUSER:
        user.objects.update_or_create(
            username=name,
            defaults={'is_staff': True, 'is_active': True,
                    'is_superuser': True}
        )
    return HttpResponse('Success')
```

- 项目根目录下的 urls.py 中的 urlpatterns 添加 path('set-su/', set_super_user)
- 添加 model 信息到 moments/admin.py

```
from django.contrib import admin
from .models import WeChatUser, Status

admin.site.register(WeChatUser)
admin.site.register(Status)
```

- 运行项目并调用 /set-su 接口，设置指定用户为超级管理员，此时返回 Success 说明成功添加管理员
- 进入 <http://dev.ce.bktencent.com:8000/admin> 管理界面，添加 WeChatUser 用户，并且和当前用户关联，点击 MOMENTS > We chat users，点击按钮 ADD WE CHAT USER，按如下填写
 - User 选择自己的账号
 - Email 随便填写
 - Motto 随便填写
 - Pic 填写 buzz.png 或 static 文件夹中的其他图片名称
 - Region 随便填写
- 点击 SAVE 按钮保存用户信息
- 此时 /post 可以正常访问，测试发送朋友圈

8. 运行项目

- git push 推送到 git 仓库，并在开发者中心部署应用
- 部署完成后访问应用，会看到网页中很多静态资源加载不出来，如下图所示，这是因为社区版本每个 SaaS 应用会在一个特定的路径下，格式为 https://apps.ce.bktencent.com/stag--{APP_ID}/，所以需要在静态资源中添加路径前缀

```
BK_STATIC_URL = "/stag--" + APP_CODE
```

- templates 下的模版文件中搜索 href 和 src 等使用到该项目路径的地方，添加 {{ BK_STATIC_URL }}，实验材料的模版文件已经默认添加了

```
<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li {% block user %}{% endblock %}><a href="{{ BK_STATIC_URL }}/user">User</a></li>
    <li {% block status %}{% endblock %}><a href="{{ BK_STATIC_URL }}/status">Moments</a></li>
    <li {% block post %}{% endblock %}><a href="{{ BK_STATIC_URL }}/post">Post</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="https://ce.bktencent.com/console/accounts/logout">Exit</a></li>
  </ul>
</div>
```

- 访问 https://apps.ce.bktencent.com/stag--{APP_ID}/set-su 设置指定用户为超级管理员
- 和本地环境一样添加 WeChatUser 用户并保存到数据库
- 尝试在 /post 界面发送朋友圈内容，发送后出现 404，因为 submit_post 路由函数是重定向到根目录的 /status，需要根据环境去调整跳转的目标路径：

← → ↻ 🏠 apps.ce.bktencent.com/status

404 Not Found

nginx

- 修改 views.py 中 submit_post 中的 redirect 路径为，如下所示：

```
from config import APP_CODE
from settings import ENVIRONMENT

def submit_post(request):
    user = WeChatUser.objects.get(user=request.user)
    text = request.POST.get('text')
    if text:
        status = Status(user=user, text=text)
        status.save()
        if ENVIRONMENT == 'dev':
            return redirect(f'/status')
        elif ENVIRONMENT == 'stag':
            return redirect(f'/stag--{APP_CODE}/status')
    return render(request, 'my_post.html')
```

- 尝试使用别的账号登录，发现 /post 界面会出错，这种情况需要先在 /admin 界面创建对应的 WeChatUser 用户