

第一次作业

基于蓝鲸体系开发部署 SaaS 应用

学号	21312450
姓名	林隽哲

一、 实验目的

- (1) 掌握 SaaS 应用的创建和部署
- (2) 学会搭建 SaaS 应用本地开发环境
- (3) 掌握 SaaS 应用开发，实现小型朋友圈应用
- (4) 通过实践编码结合 pre-commit 等规范检查工具，提升代码质量与编程能力

二、 实验环境

- (1) 硬件环境需求： PC 或笔记本， 支持外网访问
- (2) 软件环境需求

系统： Windows, MacOS, Linux

安装 Python 3.6.12

安装 MySQL 8.3

安装 Git (最新版本即可)

安装 pre-commit 代码检查工具 (可选)

安装 VSCode, PyCharm 或其它 IDE

三、 实验内容

使用蓝鲸 PaaS 平台 (<https://ce.bktencent.com/>) 创建并部署 SaaS 应用。搭建蓝鲸应用框架本地开发环境。开发一个小型朋友圈 SaaS 应用, 该应用具备三个功能: 第一是展示登录用户信息, 第二是能够显示已发送朋友圈内容, 第三是具有发送朋友圈功能

四、 实验评分标准

整体要求: 请同学们采用迭代方式进行需求分析、面向对象设计和编程实现, 实训课报告中需包含相应的需求规约、设计规约, 项目开发说明

考点一: 通过平台开发者中心创建 SaaS 应用, 新建源码仓库并部署示例程序到 PaaS 平台。搭建好本地开发环境。部署后框架界面如图:



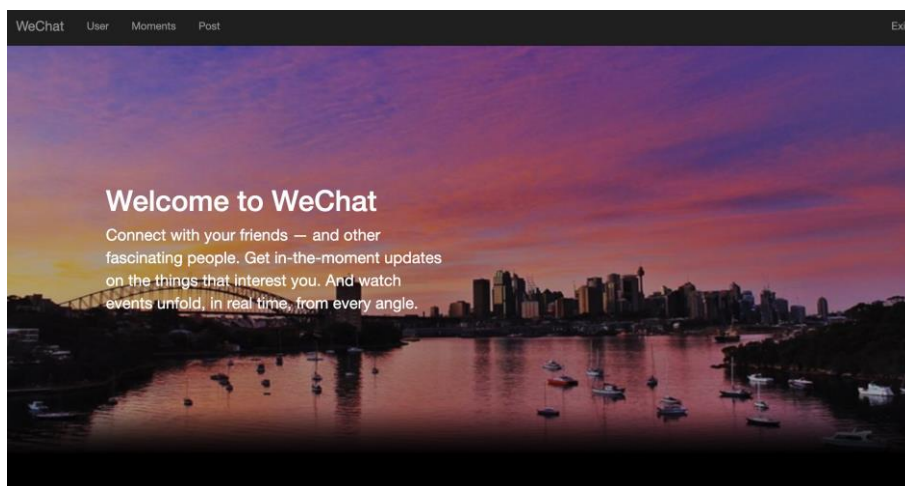
相关资料：

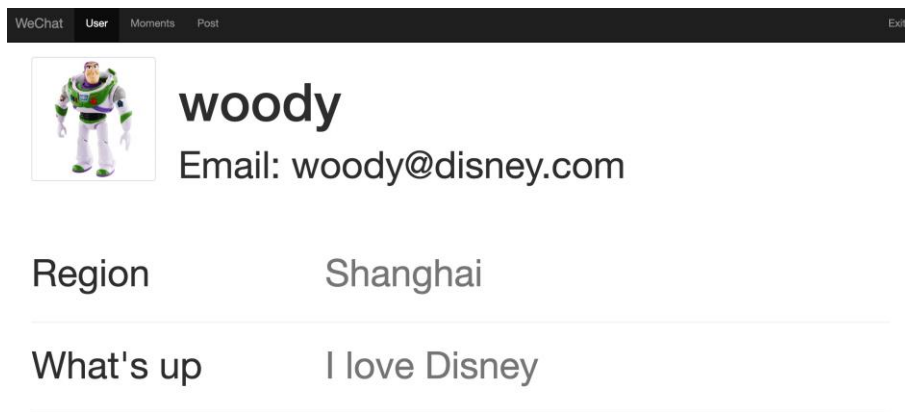
1. 手把手搭建蓝鲸开发框架环境-Windows 系统 教程文档
2. 文档中心 SaaS 开发者资料：

https://bk.tencent.com/docs/markdown/ZH/PaaS/DevelopTools/BaseGuide/quickstart/python/python_preparations.md

考点二：添加 moments 应用，并实现 WeChatUser 和 Status 两个 model。生成数据库迁移脚本并使用 migrate 变更数据库。其中 WeChatUser 是蓝鲸开发用户（from blueapps.account.models import User）的拓展（OneToOneField），具备 motto、pic（头像，可直接使用 static/image 中的图像）、region 等字段。Status 具备 user、text、pics、pub_time 等字段。

考点三：编写 view 函数实现主页和用户信息展示（前端界面已提供），如下图所示：





考点四: 实现发送朋友圈和展示朋友圈功能, 效果如下图。使用其他账号登录并发送朋友圈, 可以在 /admin 界面给登录用户创建对应的 WeChatUser 数据。



其他评分项:

1. Python 代码符合 [PEP8 规范](#), 配置 pre-commit 等格式化工具, 可酌情加分
2. 系统边界考虑完善, 系统性能优良, 可酌情加分

五、 实验过程与结果

1. 部署 Git 环境

(1) 安装 git

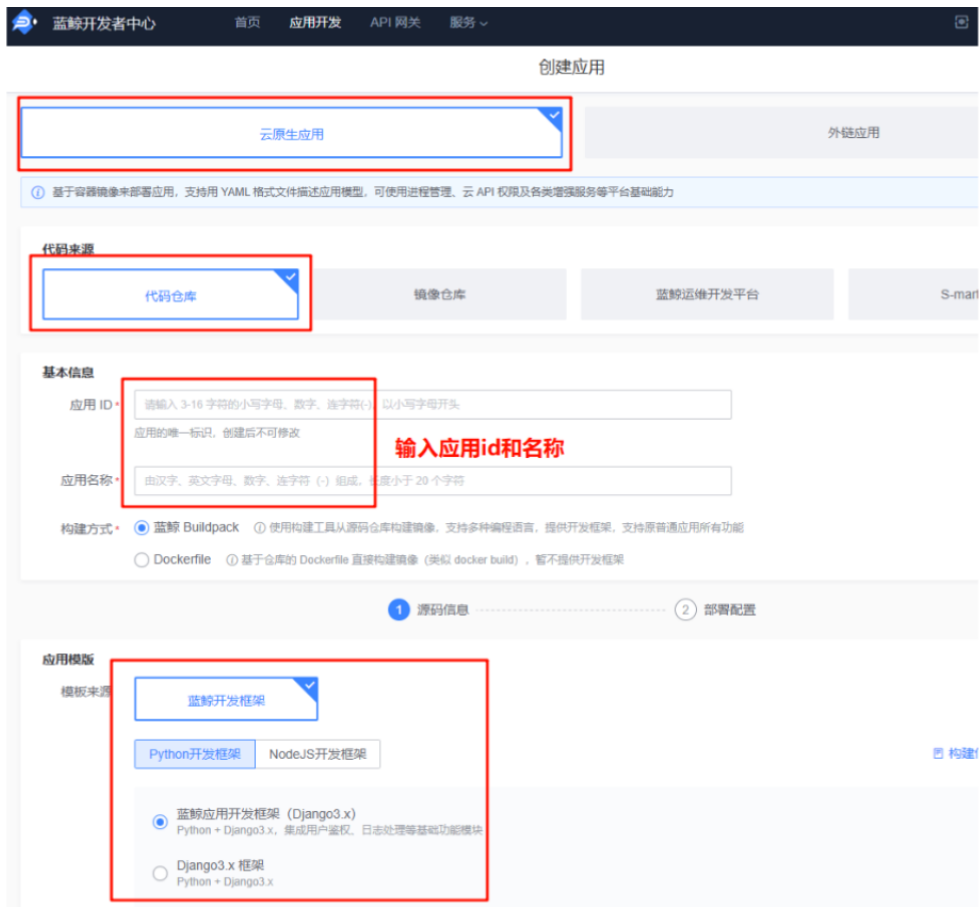
```
[K] 54.76ms $ git -v
git version 2.47.0.windows.2
```

(2) 新建 Gitee 仓库



2. 部署蓝鲸环境

(1) 在蓝鲸开发者中心中新建应用



(2) 新建应用时代码源填写自己新建的 gitee 仓库

源码管理

代码源*  Git 代码库  SVN 代码库

源代码地址*

Git 账号* 输入你自己的git代码仓库地址、账号、密码

密码*

构建目录

(3) 初始本地 Git 仓

根据命令，在本地执行初始化操作

恭喜，应用“bkdemo45”创建成功

常用操作: [查看应用概览](#) | [部署应用](#) | [添加成员](#)

应用初始化模板地址: <https://bkrepo.ce.bktencent.com/generic/temporary/download/bkpaas/bkpaas3-slug-packages/app-template-instances/default/bk-demo45-default.tar.gz?token=0529487a693c4de09ea42918b65ca959>

使用 Git 命令推送代码到远程仓库

```
# 下载并解压代码到本地目录
mkdir bk-demo45
curl "https://bkrepo.ce.bktencent.com/generic/temporary/download/bkpaas/bkpaas3-slug-packages/app-template-instances/default/bk-demo45-default.tar.gz?token=0529487a693c4de09ea42918b65ca959" > bk-demo45.tar.gz
tar -xzf bk-demo45.tar.gz -C bk-demo45

# 添加远程仓库地址并完成推送
cd bk-demo45
git init
git add .
git commit -m "init repo"
git remote add origin https://gitee.com/CHD-EASY-GOING/blueking_demo.git
git push -u origin master
```

帮助文档

[本地 Python 开发环境搭建指南](#)

[了解如何部署蓝鲸应用](#)

3. 部署 python 环境

(1) 安装 miniconda3，并新建一个 python 环境

```
[K] 793.31ms 🏠 $ scoop list | findstr miniconda
Installed apps:
miniconda3      24.9.2-0      extras      2024-11-04 22:20:36
[K] 301.14ms 🏠 $ conda env list | findstr saas
saas            D:\ScoopApp\Scoop\apps\miniconda3\current\envs\saas
[K] 1466.43ms 🏠 $ conda activate saas
(saas) [K] 1022.5ms 🏠 $ python -V
Python 3.6.12 :: Anaconda, Inc.
```

(2) 编写 runtime.txt 与 requirements.txt 文件如下，并使用 pip install -r 命令安装依赖

```
# -*- 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
```

4. 部署 mysql 数据库

(1) 安装 mysql 数据库

```
[K] 37.82ms > scoop list | findstr mysql
Installed apps:
mysql 8.3.0 <auto-generated> 2024-12-04 22:18:02
```

(2) 启动 mysql 数据库

```
[K] 3ms $ mysql --console
2024-12-13T09:07:34.736322Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2024-12-13T09:07:35.362161Z 0 [System] [MY-010118] [Server] D:\ScoopApp\Scoop\apps\mysql\current\bin\mysqld.exe (mysqld 8.3.0) starting as process 21848
2024-12-13T09:07:35.354792Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-12-13T09:07:38.766013Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-12-13T09:07:39.873219Z 0 [System] [MY-010229] [Server] Starting XA crash recovery...
2024-12-13T09:07:39.878066Z 0 [System] [MY-010232] [Server] XA crash recovery finished.
2024-12-13T09:07:42.278922Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-12-13T09:07:42.278163Z 0 [System] [MY-013602] [Server] Channel mysql_native configured to support TLS. Encrypted connections are now supported for this
2024-12-13T09:07:42.855792Z 0 [System] [MY-010931] [Server] D:\ScoopApp\Scoop\apps\mysql\current\bin\mysqld.exe: ready for connections. Version: '8.3.0'
2024-12-13T09:07:42.856232Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060
```

(3) 在 mysql 新建一个数据库，随后修改 config/dev.py 中的数据库配置信息

```
45 # 本地开发数据库设置
46 # USE FOLLOWING SQL TO CREATE THE DATABASE NAMED APP_CODE
47 # SQL: CREATE DATABASE 'bk_demo' DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci; # note:
48 DATABASES = {
49     "default": {
50         "ENGINE": "django.db.backends.mysql",
51         "NAME": APP_CODE, # noqa
52         "USER": "root",
53         "PASSWORD": "",
54         "HOST": "localhost",
55         "PORT": "3306",
56     },
57 }
58 }
```

5. 编写 moments 应用

(1) 编写 powershell 脚本 set-env.ps1

```
# Set environment variables for a BK (BlueKing) application

# Set the application ID and secret
$env:BKPAAS_APP_ID = "saas-expr-1"
$env:BKPAAS_APP_SECRET = "S0UPMZgfTpyr5izNgbz4M3I5dRc7ldR7mFBq"

# Set the major version of BKPAAS
$env:BKPAAS_MAJOR_VERSION = "3"

# Set URLs for various BK services
$env:BK_PAAS2_URL = "https://ce.bktencent.com"
$env:BK_COMPONENT_API_URL = "https://bkapi.ce.bktencent.com"
$env:BKPAAS_LOGIN_URL = "https://ce.bktencent.com/login/"

# Set CORS allowed origin
# Note: For production, change this to https://apps.ce.bktencent.com
$env:CORS_ALLOWED_ORIGIN = "http://dev.ce.bktencent.com:5000"

# Output the set environment variables (optional, for verification)
Get-ChildItem Env: | Where-Object { $_.Name -like "BK*" -or $_.Name -like "CORS*" } | Format-Table -AutoSize

Write-Host "Environment variables have been set successfully." You, 3 weeks ago • 2024-11-28-01-24-29
```

(2) 项目根目录下执行 ./set-env.ps1 初始化环境变量

```
[K] 33.5ms saas-expr-1 > .\set-env.ps1
```

Name	Value
BK_COMPONENT_API_URL	https://bkapi.ce.bktencent.com
BK_PAAS2_URL	https://ce.bktencent.com
BKPAAS_APP_ID	saas-expr-1
BKPAAS_APP_SECRET	S0UPMZgfTpyr5izNgbz4M3I5dRc7ldR7mFBq
BKPAAS_LOGIN_URL	https://ce.bktencent.com/login/
BKPAAS_MAJOR_VERSION	3
CORS_ALLOWED_ORIGIN	http://dev.ce.bktencent.com:5000

(3) 项目根目录下执行 `python manage.py startapp moments` 创建 moments 应用。随后在 `config/default.py` 中的 `INSTALLED_APPS` 添加 moments

(4) 编写 `moments/models.py` 如下：

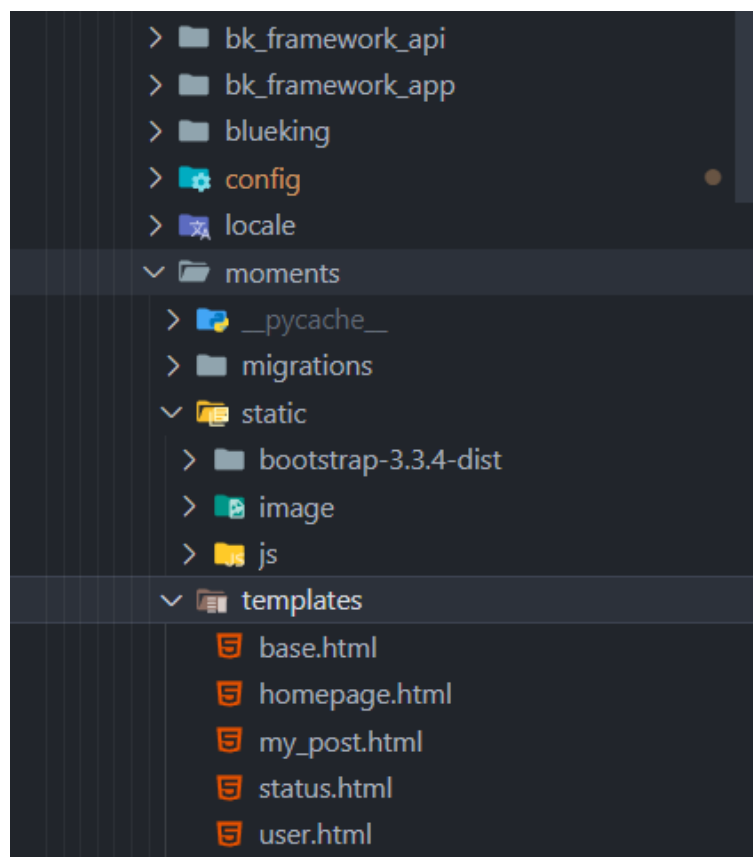

```

1  from django.db import models
2
3  from django.db import models
4  from blueapps.account.models import User
5
6  class WeChatUser(models.Model):
7      user = models.OneToOneField(User, models.CASCADE)
8      email = models.EmailField(unique=True, null=True)
9      motto = models.CharField(max_length=100, null=True, blank=True)
10     pic = models.CharField(max_length=50, null=True, blank=True)
11     region = models.CharField(max_length=50, null=True, blank=True)
12
13     def __str__(self):
14         return self.user.username
15
16     class Status(models.Model):
17         user = models.ForeignKey(WeChatUser, models.CASCADE)
18         text = models.CharField(max_length=280)
19         pics = models.CharField(max_length=100, null=True, blank=True)
20         pub_time = models.DateTimeField(auto_now_add=True)
21
22         def __str__(self):
23             return self.text
24
25         class Meta:
26             ordering = ['-id']

```

(5) 分别执行 `python manage.py makemigration` 与 `python manage.py migrate` 进行数据库迁移与变更

(6) 将实验材料中的 `static` 和 `templates` 复制到 `moments` 目录中



(7) 编写 moments/views.py 如下:

```
1  from .models import WeChatUser, Status
2  from django.shortcuts import redirect, render
3  from django.conf import settings
4  from django.http import HttpResponse
5  from blueapps.account import get_user_model
6  from config import APP_CODE
7  from settings import ENVIRONMENT
8
9  def set_super_user(request):
10     """
11     Add user as administrator
12     """
13     user = get_user_model()
14     for name in settings.INIT_SUPERUSER:
15         user.objects.update_or_create(
16             username=name,
17             defaults={'is_staff': True, 'is_active': True, 'is_superuser': True}
18         )
19     return HttpResponse('Success')
20
21 def home(request):
22     return render(request, 'homepage.html')
23
24 def show_user(request):
25     # Get monitored user id
26     user_id = request.user.id
27     # Get WeChatUser object
28     wechat_user = WeChatUser.objects.get(user_id=user_id)
29     return render(request, 'user.html', {'user': wechat_user})
30
31 def show_status(request):
32     statuses = Status.objects.all()
33     return render(request, 'status.html', {'statuses': statuses})
34
35 def submit_post(request):
36     user = WeChatUser.objects.get(user=request.user)
37     text = request.POST.get('text')
38     if text:
39         status = Status(user=user, text=text)
40         status.save()
41         if ENVIRONMENT == 'dev':
42             return redirect(f'/status')
43         elif ENVIRONMENT == 'stag':
44             return redirect(f'/stag--{APP_CODE}/status')
45     return render(request, 'my_post.html')
```

(8) 修改 urls.py 如下:

```
1  from django.contrib import admin
2  from django.urls import path
3  from moments.views import home, show_user, show_status, submit_post, set_super_user
4  from django.contrib.auth.views import LogoutView
5
6  urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('', home),
9     path('user/', show_user),
10    path('status/', show_status),
11    path('post/', submit_post),
12    path('set-su/', set_super_user)
13 ]
```

(9) 在 `config/default.py` 中设置特定用户为超级管理员:

```
1 # 初始化管理员列表, 列表中的人员将拥有预发布环境和正式环境的管理员权限
2 # 注意: 请在首次提测和上线前修改, 之后的修改将不会生效
3 INIT_SUPERUSER = ['21312450X']
```

(10) 编写 `moments/admin.py`

```
1 from django.contrib import admin
2 from .models import WeChatUser, Status
3
4 admin.site.register(WeChatUser)
5 admin.site.register(Status)
```

6. 本地启动项目

(1) 更改本机 `host` 信息

在 Windows 下, `hosts` 文件位于 `C:\Windows\System32\drivers\etc` 目录中。
要修改 `hosts` 文件, 请按照以下步骤操作:

- 1、打开文件资源管理器, 导航到 `C:\Windows\System32\drivers\etc` 目录。
- 2、右键点击 `hosts` 文件, 选择“打开方式”, 然后选择“记事本”或其他文本编辑器。
如果没有看到“打开方式”的选项, 那么请先将鼠标右键点击记事本图标, 然后选择“以管理员身份运行”, 在记事本中打开

`C:\Windows\System32\drivers\etc\hosts` 文件。

- 3、在打开的 `hosts` 文件中, 可以增加或修改相应的域名和 IP 地址映射。每行应包含一个 IP 地址, 后跟一个空格或制表符, 然后是域名。例如:

```
# CE 环境
127.0.0.1 appdev.ce.bktencent.com
127.0.0.1 dev.ce.bktencent.com
```

- 4、完成修改后, 保存并关闭 `hosts` 文件。

- 5、为使更改生效, 可能需要清除 DNS 缓存。按下 `Win + R` 键, 输入 `cmd` 并按下 `Enter` 键打开命令提示符。在命令提示符中输入以下命令并执行

```
ipconfig /flushdns
```

现在, 已成功修改了 Windows 下的 `etc/hosts` 配置。



在 hosts 文件中添加以下两行配置：

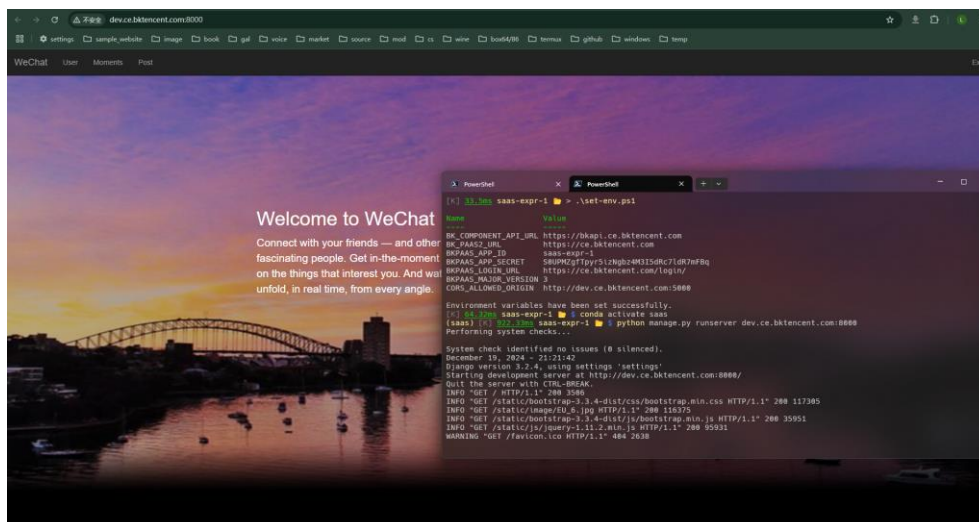
```
# CE 环境
127.0.0.1 appdev.ce.bktencent.com
127.0.0.1 dev.ce.bktencent.com
```



- (2) 通过 `python manage.py runserver dev.ce.bktencent.com:8000` 命令运行应用
- (3) 访问 `http://dev.ce.bktencent.com:8000/set-su` 设置管理员，此时返回 SUCCESS 说明成功添加管理员
- (4) 进入 `http://dev.ce.bktencent.com:8000/admin` 管理页面，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

- (6) 点击 SAVE 按钮保存用户信息
- (7) 此时/post 可以正常访问，测试发送朋友圈
- (8) 本地运行效果如下：



7. 部署应用到蓝鲸平台

(1) 在 config/stag.py 中添加如下设置:

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

(1) 上传本地项目到 gitee 仓库

```
[K] 310.94ms $ git add . ; git commit -m '2024-12-19-21-11-11' ; git push
```

(2) 部署应用

> 预发布环境部署日志 21312450X于2024-11-28 01:09:03部署

<ul style="list-style-type: none"> 准备阶段 13秒 解析应用进程信息 4秒 上传仓库代码 5秒 配置资源实例 < 1秒 构建阶段 27秒 初始化构建环境 < 1秒 分析构建方案 < 1秒 检测构建工具 < 1秒 构建应用 10秒 上传镜像 10秒 部署阶段 < 1秒 部署应用 1秒 执行部署前置命令 7秒 检测部署结果 11秒 	<div> ① 仅展示准备阶段、构建阶段日志 </div> <pre> Preparing to build bkapp-saas-expr-1-stag ... Starting build app: bkapp-saas-expr-1-stag 0 info 01:09:19.608869 Setup Build Environ 0 info 01:09:19.608961 Loading registry credentials... 0 info 01:09:19.608989 Verifying accessibility to container image registry... 0 info 01:09:20.152970 Verifying accessibility to cache registry... 0 info 01:09:20.240417 Initializing platform env... 0 info 01:09:20.242128 Setting buildpacks execute order... 0 info 01:09:20.242322 Fetching source code... 0 info 01:09:20.242336 Downloading bkrepo.ce.bktencent.com/generic/temporary/download/bkpaas/bkpaas3-slug- packages/default/home/bkapp-saas-expr-1- stag:master:22ae65624ea970a7ebadb42189c7f943fcbdlb96/tar... 0 info 01:09:20.456936 Download bkrepo.ce.bktencent.com/generic/temporary/download/bkpaas/bkpaas3-slug- packages/default/home/bkapp-saas-expr-1- stag:master:22ae65624ea970a7ebadb42189c7f943fcbdlb96/tar to path "/app" successfully 0 info 01:09:20.474795 ===== 0 info 01:09:20.474879 Starting builder... 0 info 01:09:20.474933 --> Analyzing optimization plan... Timer: Analyzer started at 2024-11-28T01:09:21+08:00 Image with name "docker.ce.bktencent.com/bkpaas/docker/saas-expr-1/default:master- 2411280109" not found Timer: Analyzer ran for 288.957786ms and ended at 2024-11-28T01:09:21+08:00 0 info 01:09:21.906946 Step Analyze done, duration 1.432002593s </pre>
---	--

(3) 最终测试结果

WeChat
 User
 Moments
 Post

21312450X
 kobayashi

11-28-2024 01:10:48

❤ Monkey, Serpent

Monkey: Great!

Po@Monkey: Thanks!

21312450X
 kobayashi

11-28-2024 01:07:36

❤ Monkey, Serpent

Monkey: Great!

六、 实验心得与体会

通过这次基于蓝鲸体系开发部署 SaaS 应用的实验,我深入了解了 SaaS 应用的创建和部署流程。从搭建本地开发环境到实现小型朋友圈应用,每一步都让我受益匪浅。特别是在使用 Git 进行版本控制、配置 Python 环境和 MySQL 数据库方面,我掌握了很多实用技能。

实现 WeChatUser 和 Status 模型,以及编写视图函数来展示用户信息和朋友圈内容,让我更好地理解 Django 框架的工作原理。这此项目不仅提高了我的编程能力,也让我认识到了良好编码习惯的重要性。

将本地开发的应用部署到蓝鲸平台的过程,让我体会到了从开发到部署的完整流程。这次实验不仅增强了我的技术能力,也让我对 SaaS 应用开发有了更全面的认识。