

# 基于PaaS平台部署应用

 蓝鲸智云 *Tencent* 腾讯

# 目录

- 开发案例需求分析与设计
- 开发实战
- 实验评分标准

# // 实战开发需求

## ●课题目标：

- 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

## ●课题内容：

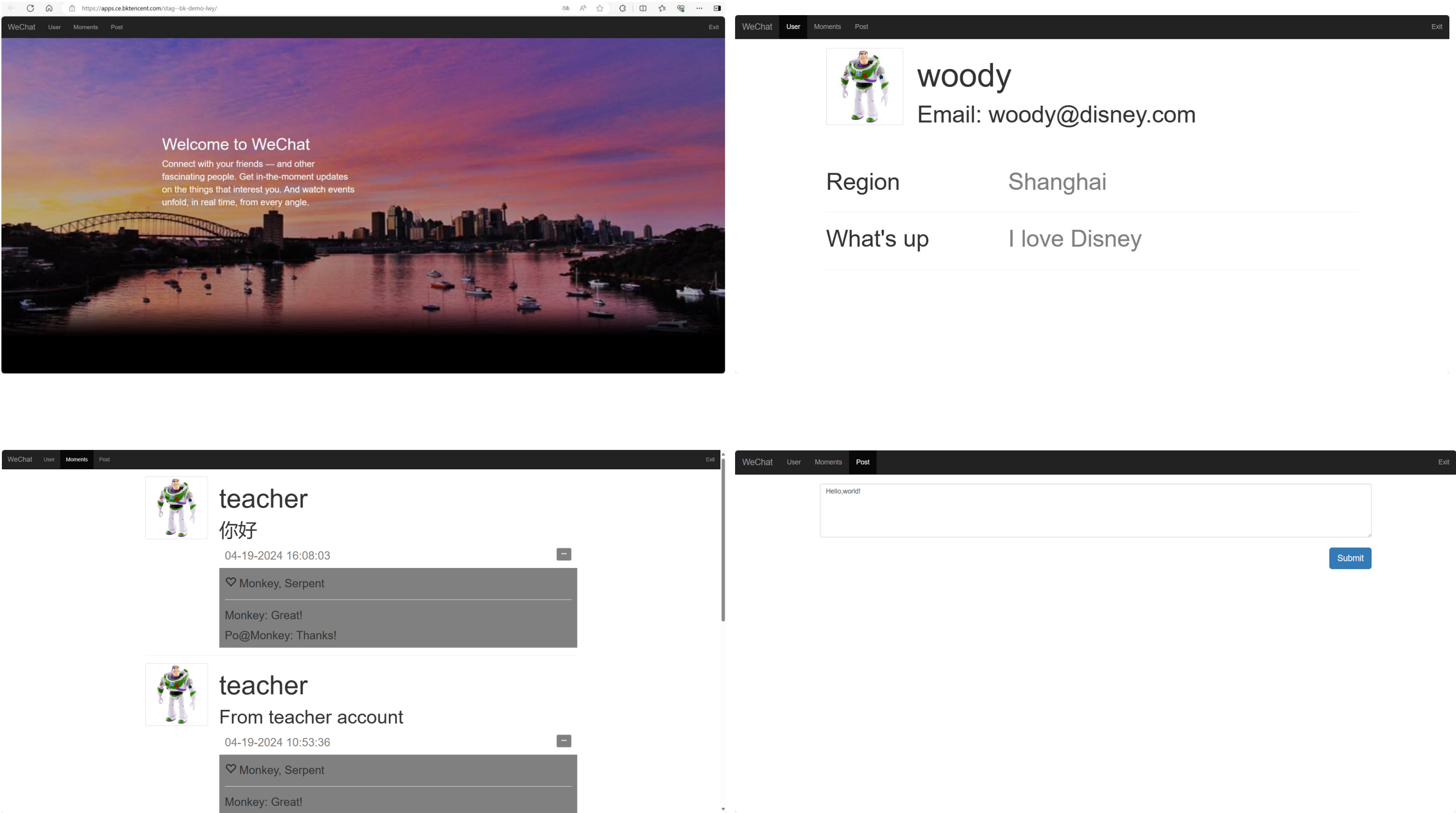
使用蓝鲸 PaaS 平台 (<https://ce.bktencent.com/>) 创建并部署 SaaS 应用。搭建蓝鲸应用框架本地开发环境。开发一个小型朋友圈 SaaS 应用，该应用具备三个功能：第一是展示登录用户信息，第二是能够显示已发送朋友圈内容，第三是具有发送朋友圈功能

## ●样例展示：<https://apps.ce.bktencent.com/stag--bk-demo-lwy/>



# // 需求分析-功能点

- 功能一： 主页信息展示
- 功能二： 用户信息展示
- 功能三： 发送朋友圈
- 功能四： 展示朋友圈



# 目录

- 开发案例需求分析与设计
- 开发实战演示
- 作业布置



# 前言：开发框架环境配置

## // 开发框架—本地环境安装

- 开发基础环境准备：
  - python安装: python3.6.12, setuptools、pip
  - mysql安装:  $\geq$  mysql5.7

- APP创建, 下载开发框架

本次课程使用环境: <https://ce.bktencent.com/>

- 开发框架依赖包安装 (在项目根目录下) :
  - pip3 install -r requirements.txt

- 配置环境变量

[蓝鲸文档中心 > PaaS 平台 > 开发者资料库 > 快速开始 > Python > 运行项目](#)

## // 开发框架- 本地运行

1. 本地创建一个和APP\_ID同名的数据库

```
CREATE DATABASE `xxx` CHARACTER SET utf8 COLLATE utf8_general_ci;
```

2. 在项目根目录执行

```
python manage.py migrate
```

3. 配置开发用的host (可用switchHosts工具设置)

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

4. 在根目录启动

```
python manage.py runserver
```

5. 访问 <http://dev.ce.bktencent.com:8000>



# // 开发框架—安装pre-commit(可选)

## 1. 安装pre-commit

`pip install pre-commit`

## 2. 初始化hooks

`pre-commit install`

## 3. pre-commit配置文件

在项目根目录下加入`.pre-commit-config.yaml`文件

## 4. Flake8配置文件

在项目根目录下加入`.flake`文件

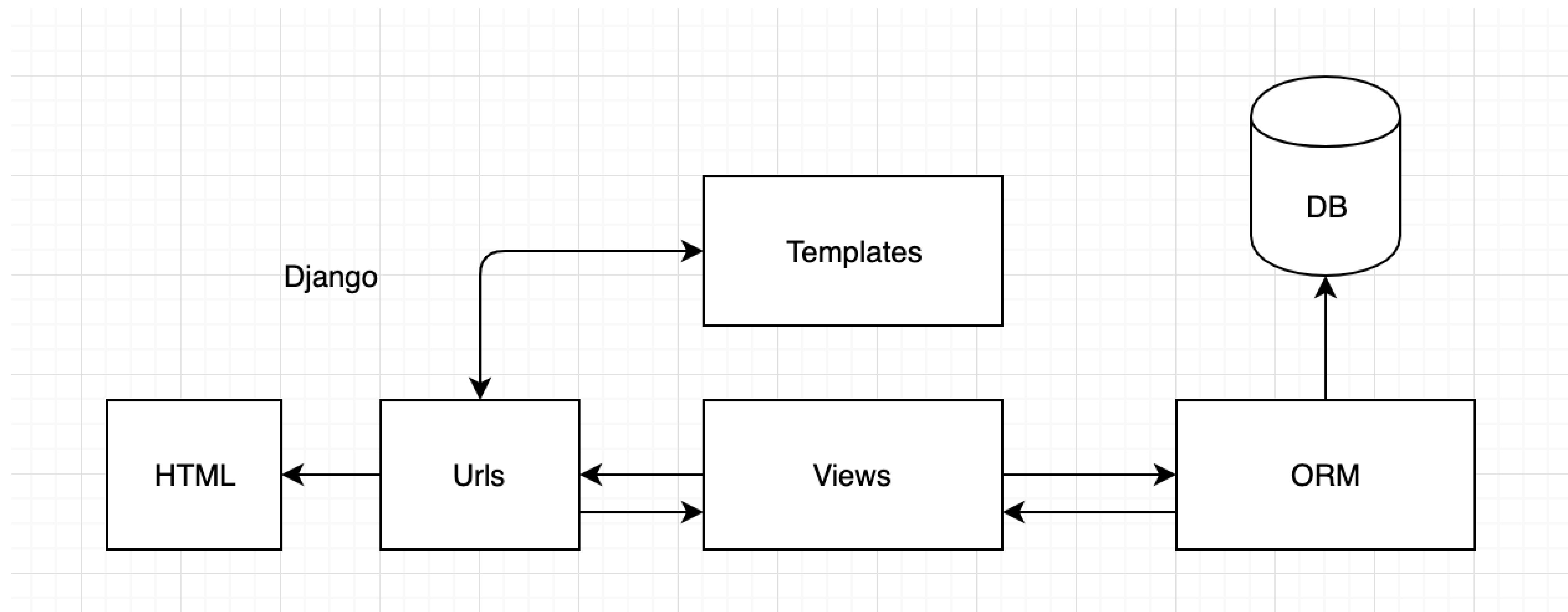
## 5. 执行代码检查

`pre-commit run -a`

```
default_stages: [commit]
repos:
  - repo: https://github.com/asottile/pyupgrade
    rev: v1.21.0
    hooks:
      - id: pyupgrade
  - repo: https://github.com/pre-commit/pre-commit-hooks
    rev: v2.1.0
    hooks:
      - id: check-merge-conflict
  - repo: https://github.com/psf/black
    rev: 22.8.0
    hooks:
      - id: black
        language_version: python3.6
        args:
          - "--line-length=120"
  - repo: https://github.com/pycqa/isort
    rev: 5.4.0
    hooks:
      - id: isort
        args:
          - "--profile"
          - "black"
  - repo: https://github.com/pre-commit/pre-commit-hooks
    rev: v2.1.0
    hooks:
      - id: flake8
        args:
          - "--config"
          - ".flake8"
```

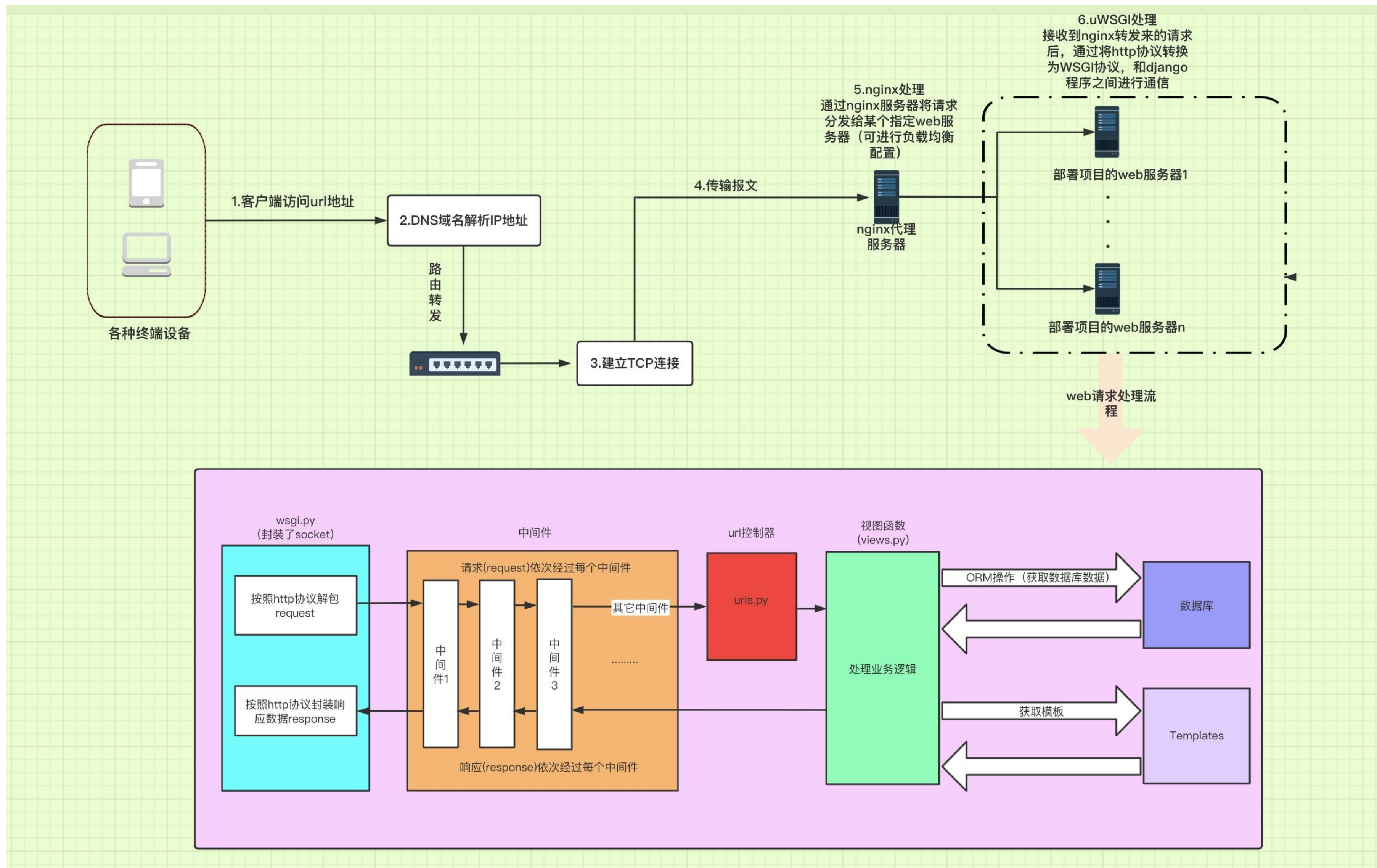
```
[flake8]
ignore = F405, W503, E203
exclude =
    *migrations*,
    # python related
    *.pyc,
    .git,
    __pycache__,
    */node_modules/*
    ./static/*
    ./blueking/*
    */blueking/*
    ./blueapps/*
    */blueapps/*
    ./pipeline/*
    ./sites/*
    ./config/default.py
    ./scripts_tools/*
    ./support-files/*
    ./upgrade.py
    ./iam/*

max-line-length=120
max-complexity=26
format=pylint
show_source=False
statistics=True
count=True
```



一句话描述Django：将数据库的东西通过ORM的映射取出来，通过view文件处理渲染返回给用户。

# // Django请求生命周期



## // 实战步骤

- 步骤一：添加moments应用
- 步骤二：创建model
- 步骤三：创建前端文件
- 步骤四：创建views视图
- 步骤五：设置路由
- 步骤六：添加admin
- 步骤七：部署运行项目



## // 步骤一：添加moments应用

- 蓝鲸开发框架默认的应用名称是home\_application，可以改成wechat，需要更改以下几个位置
  - home\_application 文件夹
  - config.py/default.py 下的 INSTALLED\_APPS
  - 项目根路径下 urls.py
- 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
- **注册moments app**
  - config.py/default.py 下的 INSTALLED\_APPS 添加 moments

**思考：如果没有注册moments app，在后续的步骤中会出现什么问题？**

## // 步骤二：创建model

- 在moments/models.py创建WeChatUser和Status模型

```
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)

    @wYour
    def __str__(self):
        return self.text

    @wYour
    class Meta:
        ordering = ["-id"]
```

```
class WeChatUser(models.Model):
    user = models.OneToOneField(User, models.CASCADE)
    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)

    @wYour
    def __str__(self):
        return self.user.username
```

- 生成数据库迁移脚本  
`python manage.py makemigrations`
- 变更数据库  
`python manage.py migrate`

```
makemigrations x
F:\Anaconda\Anaconda3\envs\bk_demo\python.exe "G:/蓝鲸课程资料/L0 课程资料/bk-demo-app/manage.py" makemigrations
No changes detected

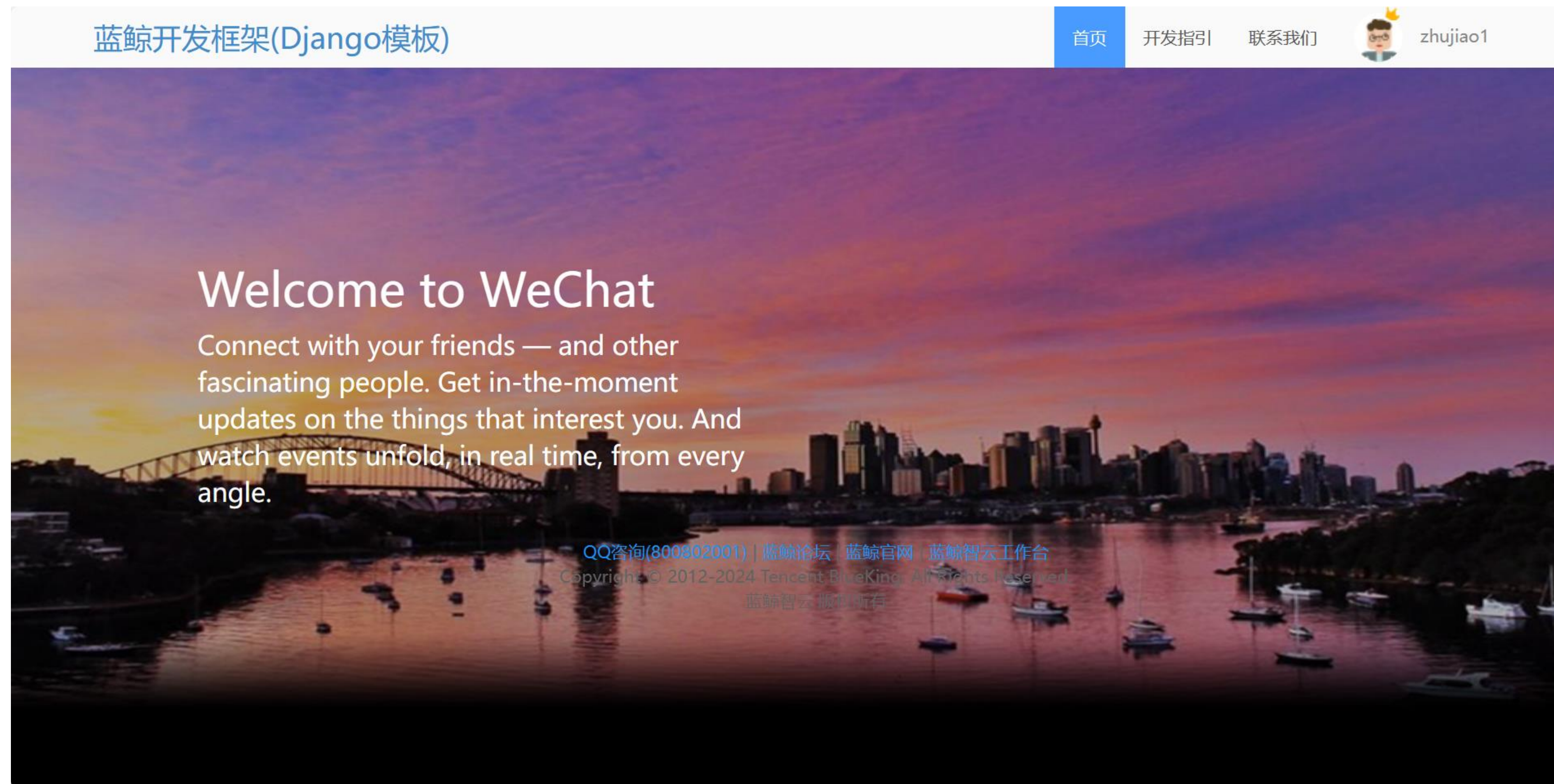
Process finished with exit code 0
```

**思考：什么原因造成的？**

**思考：makemigrations和migrate分别有什么作用/效果？**

## // 步骤三：创建前端文件

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





## // 步骤四：创建views视图

- moments/views.py添加路由函数
  - 展示主页
  - 展示用户信息
  - 展示朋友圈内容
  - 发送朋友圈

思考：Function Based View or Class Based View?

```
def home(request):
    return render(request, template_name="homepage.html")

2 个用法  👤 wYour
def show_user(request):
    po = {
        "name": "woody",
        "motto": "I love Disney",
        "email": "woody@disney.com",
        "region": "Shanghai",
        "pic": "buzz.png",
    }
    return render(request, template_name="user.html", context={"user": po})

2 个用法  👤 wYour
def show_status(request):
    statuses = Status.objects.all()
    return render(request, template_name="status.html", context={"statuses": statuses})

2 个用法  👤 wYour
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(f"/stag--{APP_CODE}/status")
    return render(request, template_name="my_post.html")
```



## // 步骤五：设置路由

- 修改项目根目录下的urls.py如下

```
DoesNotExist at /post/
WeChatUser matching query does not exist.

Request Method: GET
Request URL: http://dev.ce.bktencent.com:8000/post/
Django Version: 3.2.4
Exception Type: DoesNotExist
Exception Value: WeChatUser matching query does not exist.
Exception Location: E:\Users\weiyulin\miniconda3\envs\bk-demo\lib\site-packages\django\db\models\query.py, line 437, in get
Python Executable: E:\Users\weiyulin\miniconda3\envs\bk-demo\python.exe
Python Version: 3.6.12
Python Path: ['D:\\Users\\weiyulin\\Documents\\School\\腾讯蓝鲸\\bk-demo-lwy-default',
'D:\\Users\\weiyulin\\Documents\\School\\腾讯蓝鲸\\bk-demo-lwy-default',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers\\pydev',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers\\third_party\\thriftpy',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers\\pydev',
'D:\\Users\\weiyulin\\Documents\\School\\腾讯蓝鲸\\bk-demo-lwy-default',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers\\pycharm_display',
'C:\\Users\\weiyulin\\AppData\\Local\\JetBrains\\PyCharm2023.3\\cythonExtensions',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers-pro\\pydevd_asyncio',
'E:\\Users\\weiyulin\\miniconda3\\envs\\bk-demo\\python36.zip',
'E:\\Users\\weiyulin\\miniconda3\\envs\\bk-demo\\DLLs',
'E:\\Users\\weiyulin\\miniconda3\\envs\\bk-demo\\lib',
'E:\\Users\\weiyulin\\miniconda3\\envs\\bk-demo',
'E:\\Users\\weiyulin\\miniconda3\\envs\\bk-demo\\lib\\site-packages',
'E:\\Program Files\\JetBrains\\PyCharm ',
'2023.3.3\\plugins\\python\\helpers\\pycharm_matplotlib_backend']

Server time: Wed, 17 Apr 2024 23:06:18 +0800

Traceback Switch to copy-and-paste view

E:\Users\weiyulin\miniconda3\envs\bk-demo\lib\site-packages\django\core\handlers\exception.py, line 47, in inner
47.         response = get_response(request)
    ► Local vars

E:\Users\weiyulin\miniconda3\envs\bk-demo\lib\site-packages\django\core\handlers\base.py, line 181, in _get_response
181.         response = wrapped_callback(request, *callback_args, **callback_kwargs)
    ► Local vars

D:\Users\weiyulin\Documents\School\腾讯蓝鲸\bk-demo-lwy-default\moments\views.py, line 26, in 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

## // 步骤六：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']`

如果不是第一次部署，[需要设置路由函数手动指定用户为超级管理员](#)

- 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')
```

## // 步骤六：添加admin

- urls.py 中添加 `path('set-su', set_super_user)`
- 添加 model 信息到 moments/admin.py  
`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
  - 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

## // 步骤七：部署运行项目

- git push 推送到git仓库，并在开发者中心的预发布环境部署应用
- 部署完成后访问应用，对功能进行测试
- 如果出现问题，则需要对问题进行排查，修改代码，并经过本地测试后，将代码重新推送到git仓库，部署和测试
- 测试通过，项目完成



# // 步骤七：部署运行项目-排查问题（1）

原因分析：访问静态资源路径不对

思考：项目的路径是怎么构成的

`https://apps.ce.bktencent.com/stag--{APP_ID}/`

本地的路径

`/static/bootstrap-3.3.4-dist/css/bootstrap.min.css`

解决方法：

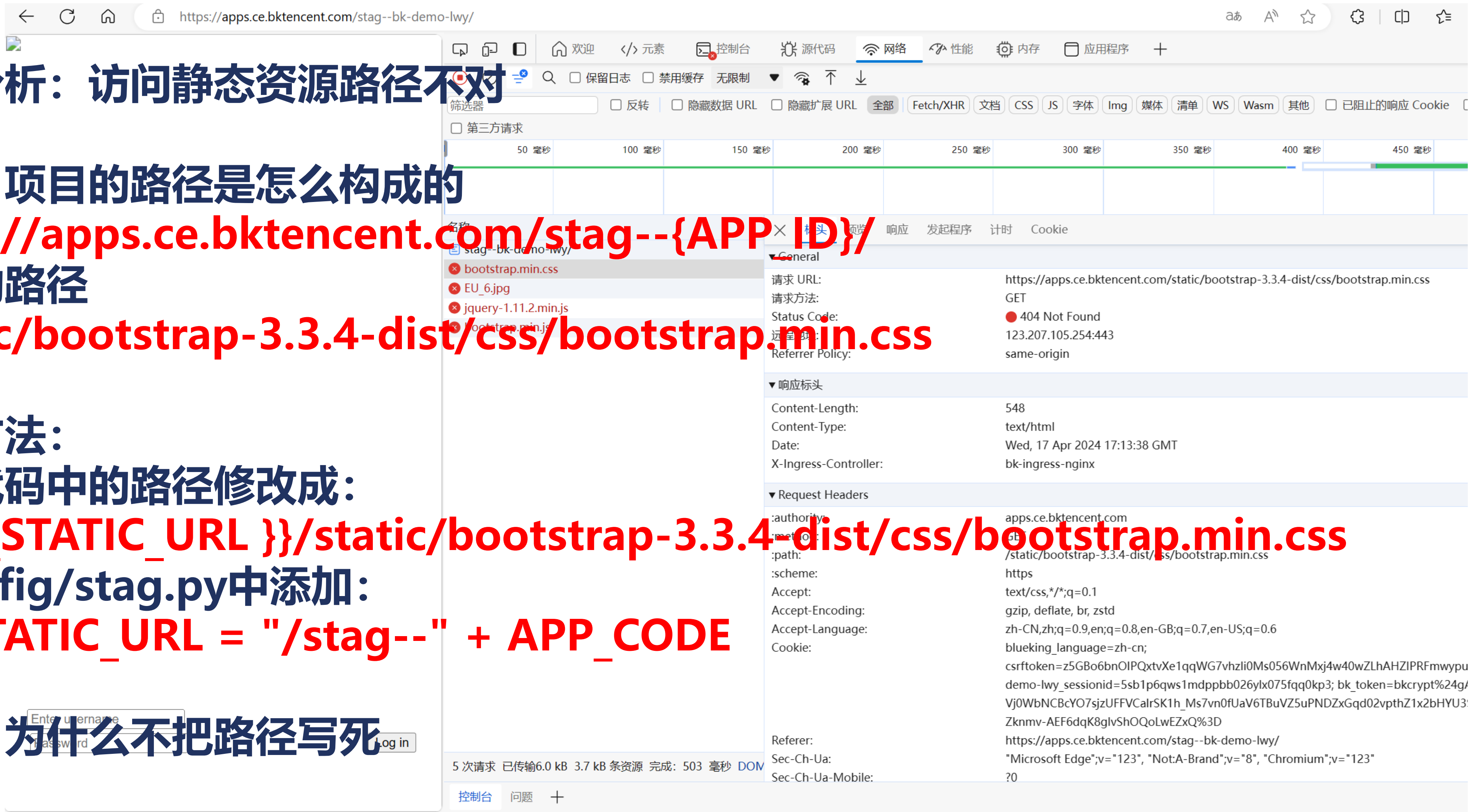
前端代码中的路径修改成：

`{{ BK_STATIC_URL }}/static/bootstrap-3.3.4-dist/css/bootstrap.min.css`

在config/stag.py中添加：

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

思考：为什么不把路径写死



## // 步骤七：部署运行项目-排查问题（2）

尝试在 /post 界面发送朋友圈内容，发送后出现 404



解决方法：通过settings.py的ENVIRONMENT获取目前环境，根据环境来调整url。如下图：

```
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('/status')
        elif ENVIRONMENT == 'stag':
            return redirect(f'/stag--{APP_CODE}/status')
    return render(request, template_name: 'my_post.html')
```

更优雅的写法？

# 目录

- 开发案例需求分析与设计
- 开发实战演示
- 实验评分标准

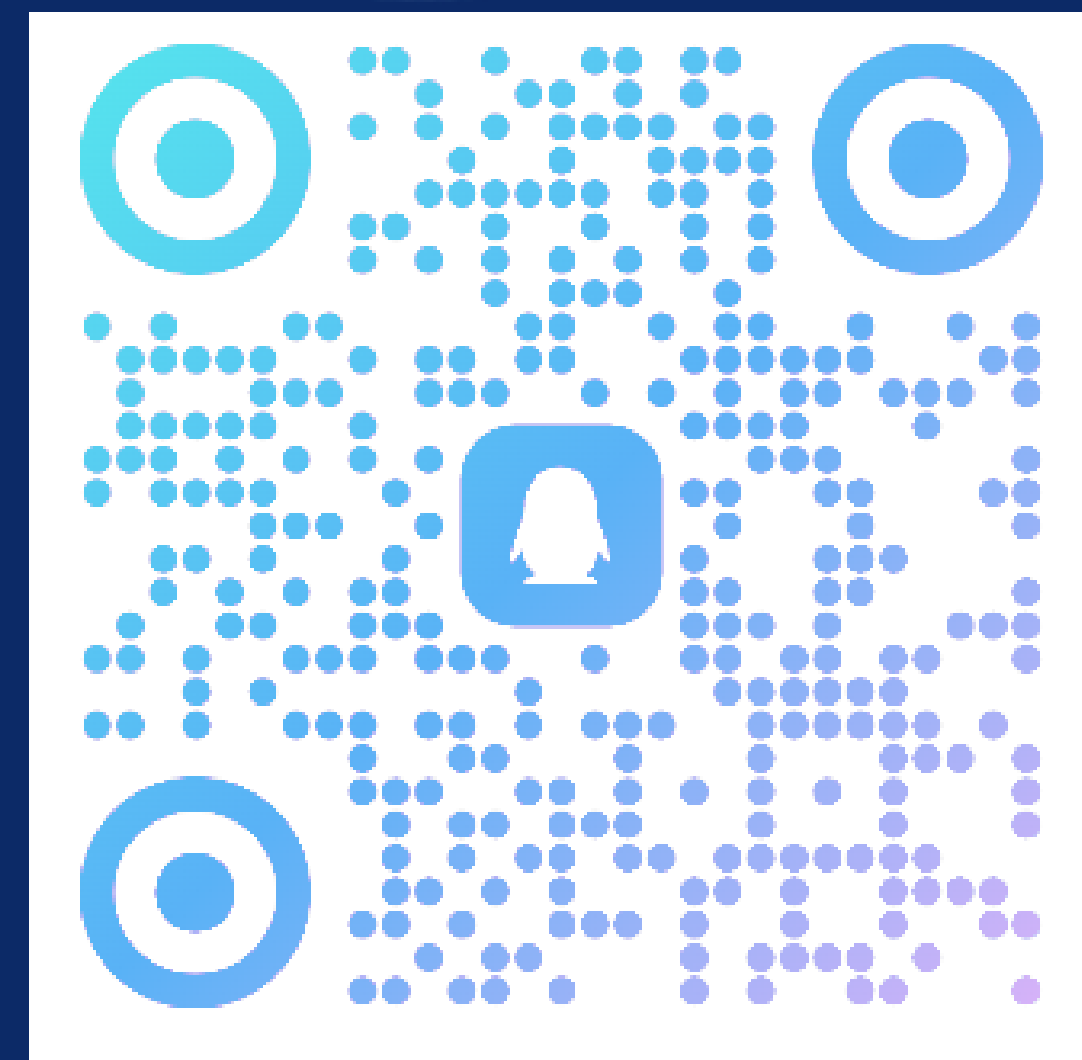
# // 实验评分标准

整体要求	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
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
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）
考点四	实现发送朋友圈和展示朋友圈功能。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





官方微信公众号



蓝鲸高校培训群