

中山大学软件学院 2012 级

# 部署文档

---

——公益活动管理系统 Philosopher

2015-7-18

项目小组成员

12330054 陈跃东

12330053 陈远宏

12330051 陈宇文

12330084 封伟城

12330293 田 澍

## 目录

一. 安装的环境、步骤.....	2
1. 部署平台介绍.....	2
2. 部署前的本地配置.....	2
3. Heroku 账号 .....	2
4. 部署到 Heroku.....	3
二. 验证方法.....	3
1. 让 Heroku 启动应用 .....	3
2. 同步数据库.....	3
3. 访问网站后台.....	3
4. 让 Heroku 停止应用.....	4
三. 常见问题及解决方案 .....	4
1. 如何配置相关资源的路径 .....	4
2. Django 中的 ImageField 在 Heroku 中无法正常使用 .....	5

## 一. 安装的环境、步骤

### 1. 部署平台介绍

本系统部署在 Heroku 平台上。



Heroku 是一个支持多种编程语言的云平台即服务，选用该平台的原因有三：

- Heroku 为 Web 应用提供了相当方便的部署服务
- Heroku 对于访问量不是很大的小应用是免费的
- 在普通的云平台上配置 Django 的运行环境是一件较为麻烦的事

基于此三点，对于本应用来说，部署到 Heroku 是个较为明智的选择。

### 2. 部署前的本地配置

#### ● requirements.txt

我们需要创建一个 requirements.txt 文件来告知 Heroku 需要在服务器上创建哪些 python 包。

不过首先，Heroku 需要我们在本地按照一些包。执行如下命令：

```
pip install dj-database-url gunicorn whitenoise
```

然后，在项目根目录下，也就是 manage.py 所在目录执行：

```
pip freeze > requirements.txt
```

这条命令会新建一个 requirements.txt 文件，里面包含了工程的所有依赖包。

然后打开 requirements.txt 文件，在文件最后加入

```
psycopg2==2.5.4
```

psycopg 应用能够在你的应用部署到 Heroku 时，自动为您匹配正确的数据库。

#### ● Procfile 文件

另一个需要创建的文件就是 Procfile，它告知 Heroku 在启动网站时需要执行的命令。同样在 manage.py 所在目录，新建一个文件名为 Procfile，添加如下内容：

```
web: gunicorn ActivityManager.wsgi
```

这句话的意思就是我们将要部署一个 web 应用程序，并且我们通过执行 gunicorn ActivityManager.wsgi 命令来完成。其中的 ActivityManager 是项目的名称，而 gunicorn 是一个类似 django 的 runserver 的程序，但是功能更强大。

### 3. Heroku账号

申请一个 Heroku 账号，地址：<https://id.heroku.com/signup/www-home-top>

接下来，需要现在本地环境上安装 Heroku toolbelt，教程地址：<https://toolbelt.heroku.com/>

本人使用的是 Ubuntu14.04 操作系统，直接使用如下命令即可：

```
wget -qO- https://toolbelt.heroku.com/install.sh | sh
sudo echo 'PATH="/usr/local/heroku/bin:$PATH"' >> /etc/profile
source /etc/profile
```

之后，在终端中输入

```
heroku login
```

便可登录 heroku 账号。

## 4. 部署到Heroku

- 创建一个工程

在 django 项目的最高目录中，输入如下命令

```
heroku create activitymanager
```

返回结果

```
Creating activitymanager... done, stack is cedar
```

```
https://activitymanager.herokuapp.com/ | https://git.heroku.com/activitymanager.git
```

请记住上面的地址 <https://git.heroku.com/activitymanager.git>

- 初始化 git 本地仓库提交，并添加上面的远程仓库地址

输入如下命令

```
git init
```

```
git add .
```

```
git commit -m 'activitymanager by django.'
```

```
git remote add origin https://git.heroku.com/activitymanager.git
```

- Push 项目工程

```
git push heroku master
```

最后得到如下返回结果

```
remote:      Running setup.py develop for PIL
remote:      Successfully installed PIL-1.2a0
remote: You are using pip version 7.0.3, however version 7.1.0 is available.
remote: You should consider upgrading via the 'pip install --upgrade pip' command.
remote:
remote: -----> Preparing static assets
remote:      Running collectstatic...
remote:      114 static files copied.
remote:
remote: -----> Discovering process types
remote:      Procfile declares types -> web
remote:
remote: -----> Compressing... done, 51.2MB
remote: -----> Launching... done, v9
remote:      https://activitymanager.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/activitymanager.git
   066dd53..2cbdfbe  master -> master
```

则代表项目已经成功部署

## 二. 验证方法

### 1. 让Heroku启动应用

```
heroku ps:scale web=1 --app activitymanager
```

这样便启动了一个应用

### 2. 同步数据库

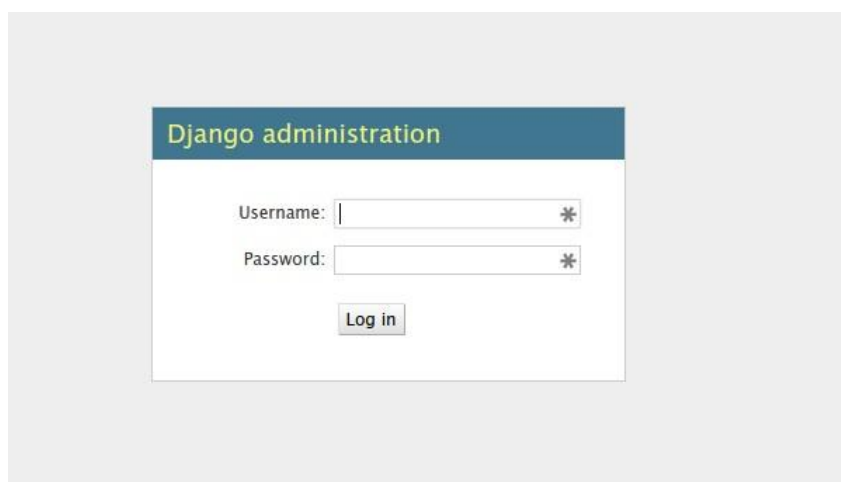
输入如下命令

```
heroku run python manage.py syncdb
```

### 3. 访问网站后台

进入网址 <http://activitymanager.herokuapp.com/admin/>

若出现如下页面，说明网站部署已经完全成功了。



这时，输入网址：<https://activitymanager.herokuapp.com/> 便可正常访问网站了

#### 4. 让Heroku停止应用

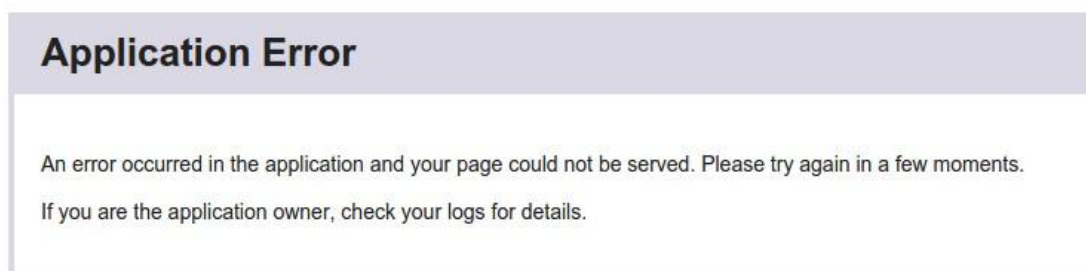
输入如下命令

```
heroku ps:scale web=0 --app activitymanager
```

若返回

```
Scaling dynos... done, now running web at 0:1X.
```

则说明网站已经停止运行，这时再输入网址将会出现如下错误页面



### 三. 常见问题及解决方案

#### 1. 如何配置相关资源的路径

Django1.4 中，media，template 等资源的路径都必须使用绝对路径标明。这样，当把应用部署到 Heroku 平台时，需要得知我们部署到该平台上时的绝对路径，那将是一件比较麻烦的事情。所以，我们可以通过修改 settings.py 文件来解决这一问题，代码如下：

```
import os.path
root = os.path.dirname(__file__).replace('\\', '/')

MEDIA_ROOT = root + '/../media/'
STATIC_ROOT = root + '/../static/'
TEMPLATE_DIRS = (
    root + '/../templates',
)
```

## 2. Django中的ImageField在Heroku中无法正常使用

本地环境中，配置一下 PIL 扩展即可解决，但是在 requirement 文件中指定 PIL 之后，却发现无法正常安装，最后，通过在 requirements.txt 中加入 PIL 的链接来解决这个问题，指令如下：

```
-e hg+https://bitbucket.org/etienned/pil-2009-raclette/#egg=PIL
```