|  |
| --- |
| 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](#_Toc425000050)

[1. 部署平台介绍 2](#_Toc425000051)

[2. 部署前的本地配置 2](#_Toc425000052)

[3. Heroku账号 2](#_Toc425000053)

[4. 部署到Heroku 3](#_Toc425000054)

[二．验证方法 3](#_Toc425000055)

[1. 让Heroku启动应用 3](#_Toc425000056)

[2. 同步数据库 3](#_Toc425000057)

[3. 访问网站后台 3](#_Toc425000058)

[4. 让Heroku停止应用 4](#_Toc425000059)

[三．常见问题及解决方案 4](#_Toc425000060)

[1. 如何配置相关资源的路径 4](#_Toc425000061)

[2. Django中的ImageField在Heroku中无法正常使用 5](#_Toc425000062)

## 一．安装的环境、步骤

### 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。

### 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。

### 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。

### 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

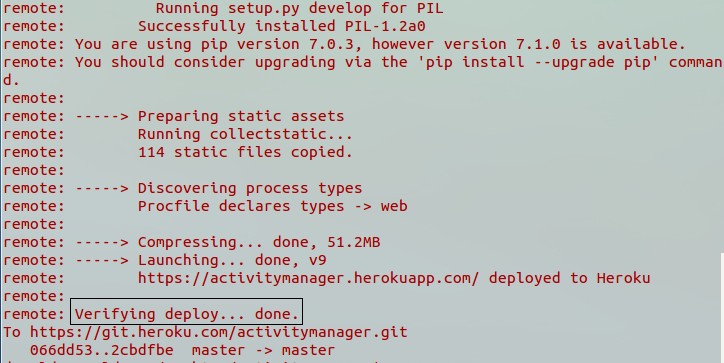
输入如下命令

|  |
| --- |
| 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



则代表项目已经成功部署

## 二．验证方法

### 让Heroku启动应用

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

这样便启动了一个应用

### 同步数据库

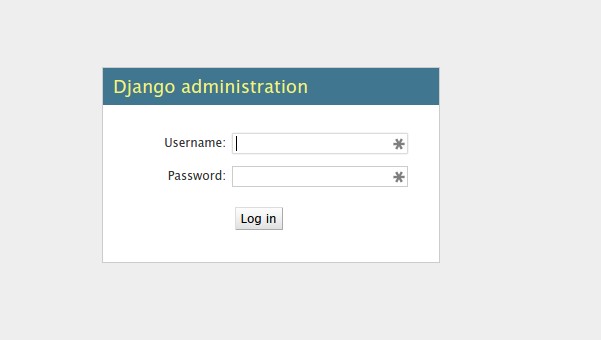
输入如下命令

|  |
| --- |
| heroku run python manage.py syncdb |

### 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

### 让Heroku停止应用

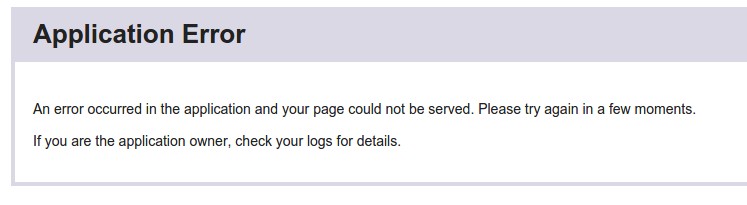
输入如下命令

|  |
| --- |
| 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

### 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',  ) |

### 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的链接来解决了这个问题，指令如下：

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