



YunsChou

既然选择了远方，便只顾风雨兼程

2017-11-24 • heroku,vapor,swift

# 将Vapor项目部署到Heroku

说明：

- 1、本操作仅为熟悉流程，一切从简，通俗易懂
- 2、本操作流程不需要本地安装Heroku和数据库，方便快捷

## 一、将本地vapor项目上传至GitHub

- 1、安装vapor
- 2、使用 `vapor new [项目名]` 创建应用
- 3、进入项目根目录，执行 `vapor xcode -y` 使用Xcode来运行
- 4、通过浏览器访问 `http://0.0.0.0:8080/hello`，返回内容说明运行成功
- 5、在GitHub上新建一个空仓库，并把本地项目上传到GitHub

在项目根目录，执行以下两行命令：

```
git remote add origin git@github.com:xxx/xxx.git
git push -u origin master
```

## 二、将GitHub项目关联到Heroku上

1、注册Heroku账号，自备梯子

2、在Heroku上创建一个新应用

3、在 `Settings->Buildpacks` 下，添加

<https://github.com/kylef/heroku-buildpack-swift>

因为Heroku中并没有提供swift语言的选项，所以我们要自行添加URL（告诉它build的时候使用swift）

4、在 `Deploy->Deployment method` 下，关联GitHub仓库和项目

5、在 `Deploy->Manual deploy` 下，选择正确的分支，并点击 `Deploy Branch` 按钮，Heroku会开始部署项目

如果是首次部署，可能需要多花一点时间安装配置，耐心等待部署完成.....

### 三、访问链接：

1、部署完成后，点击项目右上角的 `Open app` 按钮或手动输入域名，访问链接 <https://xxxxxx.herokuapp.com>（xxxxxx是Heroku中创建的应用名）

2、此时会提示 `Application error`，别慌！离最终成功已经很近了，请继续以下操作

3、进入项目根目录，执行 `touch Procfile` 创建名为Procfile的文件，并用编辑打开添加 `web: Run --env=production --port=$PORT`，然后push到GitHub

4、将Procfile文件提交到GitHub后，刷新Heroku控制面板，在Dyno处可以看到多了Procfile中的内容

5、重新进入 `Deploy->Manual deploy`，点击 `Deploy Branch` 按钮重新部署

6、再次访问链接，成功

## 四、注意

1、Procfile文件可以在新建项目后一并创建，提交至GitHub，此处后添加只为演示

2、在Heroku中可以选择自动部署，当选中的分支（执行自动部署的分支）每次有代码push上来的时候，都会执行自动部署。如果你是测试或练习项目，没有什么影响，如果是正式项目，请慎重选择并做好分支控制

3、本项目 [Heroku链接](#)，[GitHub链接](#)

---

更早的文章

## 【组件化-2】组件化的配置及命令行的使用

一、组件的.podspec文件存放1、远程：在代码托管平台上新建空一个项目

（YsPodSpec），用来存放组件

的.podspec 文件2、本地：使用命令在本地创建一个仓库，并使该仓库指向远程仓库

（YsPodSpec）`pod repo add [本地repo名称] [远程仓库http地址]`本地仓库

（repo）的作用：用来存放本地组件工程的.podspec文件，将本地.podspec文件上传（push）到远程仓库二、组件工程文件／代码存放1、远程：在代码托管平台上新建一个空项目（YsModul.....

2017-08-08 • 组件化 [继续阅读](#)

