

# 用 Travis-CI 搭建 Github Pages 网站

目录

- [用 Travis-CI](#)
- [用 Travis-CI 部署](#)
- [用 Travis-CI 搭建 Github](#)
- [用 Web 部署静态网站](#)

2015年2月21日

本文档记录了我用 Travis-CI 搭建 Github Pages 网站的过程。

用 Github Pages 部署静态网站，使用 [Jekyll](#) 作为静态网站生成器。Github Pages 使用 [Jekyll](#) 作为静态网站生成器，通过 push 到 Github Pages 仓库，Github Pages 会自动生成静态网站。

用 [Pelican](#) 作为静态网站生成器，使用 [pelican](#) 生成静态网站。Pelican 使用 [python](#) 编写，使用 [pelican](#) 生成静态网站。

Windows Linux/OSX/Unix-like wordpress web Android SL4A python pelican Android git

Continuous integration Travis-CI

# Travis-CI

Agile Development Extreme Programming  
git commit

Travis-CI github repo github

# Travis-CI

https://travis-ci.org/ Github https://travis-ci.org/repositories repo push

farseerfc/farseerfc

ON

Travis-CI Github Repo

repo .travis.yml .travis.yml

```
1 language: python
2
3 python:
4   - "2.7"
5
6 before_install:
7   - sudo apt-add-repository ppa:chris-lea/node.js -y
8   - sudo apt-get update
9   - sudo apt-get install nodejs ditaa doxygen parallel
10
11 install:
12   - sudo pip install pelican
13   - sudo pip install jinja2
14   - sudo pip install babel
15   - sudo pip install beautifulsoup4
16   - sudo pip install markdown
17   - sudo npm install -g less
18   - wget "http://downloads.sourceforge.net/project/plantuml
/plantuml.jar?r=&ts=1424308684&use_mirror=jaist" -O plantu
ml.jar
19   - sudo mkdir -p /opt/plantuml
20   - sudo cp plantuml.jar /opt/plantuml
21   - echo "#! /bin/sh" > plantuml
22   - echo 'exec java -jar /opt/plantuml/plantuml.jar "$@"' >>
plantuml
23   - sudo install -m 755 -D plantuml /usr/bin/plantuml
24   - wget https://bintray.com/artifact/download/byvoid/openc
c/openc-1.0.2.tar.gz
25   - tar xf openc-1.0.2.tar.gz
26   - cd openc-1.0.2 && make && sudo make install && cd
..
27   - sudo locale-gen zh_CN.UTF-8
28   - sudo locale-gen zh_HK.UTF-8
29   - sudo locale-gen en_US.UTF-8
30   - sudo locale-gen ja_JP.UTF-8
31
32 script:
33   - git clone --depth 1 https://github.com/farseerfc/pelican-
plugins plugins
34   - git clone --depth 1 https://github.com/farseerfc/pelican-
bootstrap3 theme
```

bootstrap theme

35 - mkdir output

36 - env SITEURL="farseerfc.me" make publish

Travis-CI 環境で Ubuntu 12.04 LTS で python をインストールしようとしたら、python 2.7 をインストールする際に pip をインストールする before\_install ではなく install が必要で build errored ではなく build fail になる script を修正して build fail を回避する

less.js を ppa を使わずに nodejs でインストールする less を opencv 1.0.2 を Ubuntu で opencv をインストール(0.4)して doxygen を opencv をインストールして pelican をインストールして locale を 4 個 locale-gen を Ubuntu で Linux をインストールして Ubuntu をインストールする

.travis.yml を push して github で travis を clone して travis を build して passing になる build を確認する

## Travis-CI と Github

travis-ci を Github Pages で公開して Github を ssh key を github repo を travis に key をインストールする

Github Personal Access Token

Applications / **New personal access token**

Token description

travis blog push

What's this token for?

Select scopes

Scopes *limit* access for personal tokens. [Read more about OAuth scopes.](#)

☐ repo ⓘ
 ☐ repo:status ⓘ
 ☐ repo\_deployment ⓘ
 ☒ public\_repo ⓘ
 ☐ delete\_repo ⓘ
 ☐ user ⓘ
 ☐ user:email ⓘ
 ☐ user:follow ⓘ
 ☐ admin:org ⓘ
 ☐ write:org ⓘ
 ☐ read:org ⓘ
 ☐ admin:public\_key ⓘ
 ☐ write:public\_key ⓘ
 ☐ read:public\_key ⓘ
 ☐ admin:repo\_hook ⓘ
 ☐ write:repo\_hook ⓘ
 ☐ read:repo\_hook ⓘ
 ☐ admin:org\_hook ⓘ
 ☐ gist ⓘ
 ☐ notifications ⓘ

Generate token

? Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

1. Github 上 Personal Access Token App Token 生成 Travis-CI 使用

2. Personal Access Token Token 生成 public\_repo 使用 Token

3. travis 上

2015-2-21

1. travis encrypt 使用 ruby 生成 travis api

2. repo pubkey

```

1 curl -H "Accept: application/vnd.travis-ci.2+json" https://api.travis-ci.org/repos/<github-id/repo>/key | python2 -m json.tool | grep key | sed 's/.*"key": "(.*)"/\1/' | xargs -0 echo -en | sed 's/ RSA/' > travis.pem
  
```

1 echo -n <github-id/repo> | openssl genrsa -out key.pem 2048  
2 openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365  
3 openssl xargs -0 echo -en "-----BEGIN RSA PRIVATE KEY-----\n" <key.pem> | openssl rsa -outform -out travis.pem  
4 mv travis.pem .travis.yml

5 openssl pubkey -in travis.pem -out travis.pub

```
1  echo -n 'GIT_NAME="Jiachen Yang" GIT_EMAIL=farseerf  
c@gmail.com GH_TOKEN=<Personal Access Token>' | ope  
nssl rsautl -encrypt -pubin -inkey travis.pem | base64 -w0
```

6 echo -n "-----BEGIN RSA PRIVATE KEY-----\n" <travis.pub> | openssl rsa -outform -out travis.pem

7 travis encrypt 'GIT\_NAME="Jiachen Yang" GIT\_EMAIL=farseerf  
c@gmail.com GH\_TOKEN=<Personal Access Token>'

```
1  $ gem install travis
```

8 travis status --repo <github-id/repo> --build <github-id/repo>

```
1  $ travis encrypt 'GIT_NAME="Jiachen Yang" GIT_EMAIL=farse  
erfc@gmail.com GH_TOKEN=<Personal Access Token>'
```

9 echo -n "-----BEGIN RSA PRIVATE KEY-----\n" <travis.pub> | openssl rsa -outform -out travis.pem

```
1  env:  
2    - secure: "long secure base64 string"
```

10 travis status --repo <github-id/repo> --build <github-id/repo>

```

1  script:
2    - git config --global user.email "$GIT_EMAIL"
3    - git config --global user.name "$GIT_NAME"
4    - git config --global push.default simple
5    - git clone --depth 1 https://github.com/farseerfc/pelican-plugins
6    - git clone --depth 1 https://github.com/farseerfc/pelican-bootstrap3 theme
7    - git clone --depth 1 https://$GH_TOKEN@github.com/farseerfc/farseerfc.github.io output
8    - env SITEURL="farseerfc.me" make publish
9
10  after_success:
11    - cd output
12    - git add -A .
13    - git commit -m "update from travis"
14    - git push --quiet

```

1. 在 `script` 部分添加 `git push --quiet` 命令，确保每次构建成功后都能自动推送到 GitHub。

2. 在 `after_success` 部分添加以下命令，用于构建并推送到 GitHub：

```

cd output
git add -A .
git commit -m "update from travis"
git push --quiet

```

## Web 部署

1. 在 GitHub 仓库的 `Travis-CI` 配置文件中，添加 `build passing` 状态，并指定 `Readme.md` 文件。

2. 在 GitHub 仓库的 `Web` 配置文件中，添加 `Github` 配置，并指定 `commit` 命令。

