

# 用 Travis-CI 搭建 Github Pages 网站

目录

- [用 Travis-CI](#)
- [用 Travis-CI 部署网站](#)
- [用 Travis-CI 搭建 Github](#)
- [用 Web 部署网站](#)

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 部署网站，需要用到 [Jekyll](#) 这个工具。Github Pages 会自动把 [Jekyll](#) 的模板渲染成 HTML，当你 push 到 Github Pages 仓库时，Github Pages 会自动部署到网上。

用 [Pelican](#) 部署网站，需要用到 [pelican](#) 这个工具。Pelican 会自动把 [Jekyll](#) 的模板渲染成 HTML，当你 push 到 Github Pages 仓库时，Github Pages 会自动部署到网上。

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時に pip をインストールする before\_install と install の間で build errored となり build fail となり script が build fail した

less.js を ppa でインストールする nodejs をインストールする less をインストールする opencv 1.0.2 を Ubuntu でインストールする opencv (0.4) を doxygen を opencv を pelican を 4 locale を 4 locale-gen を Ubuntu を Linux を Ubuntu

.travis.yml を push して github で travis を clone して travis を build して passing して build する

## Travis-CI と Github

travis-ci と Github Pages を使えば Github で ssh key を github repo に travis の key を

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 を取得する

2. Personal Access Token Token を `public_repo` 権限を持つ Token として

3. travis へ

2015年2月21日

1. travis encrypt して ruby 上で

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 <github-id/repo> | openssl rsa -pubkey -in travis.pem -out travis.pubkey

2 openssl rsa -pubkey -in travis.pubkey -out travis.pubkey.pem

```
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
```

3 echo -n 'token' | openssl sdiff -in travis.pubkey.pem -out travis.token

4 travis token archlinux aur/ruby-travis

```
1  $ gem install travis
```

5 travis status

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

6 .travis.yml

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

7 travis

```

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

部署流程如下：



1. 在 Travis-CI 上配置好项目后，每次提交到 `main` 分支时，Travis-CI 会自动触发构建。

2. 构建成功后，Travis-CI 会自动推送到 GitHub，并触发 GitHub Actions 的部署流程。

