

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 Travis-CI 搭建 Github Pages 网站的过程，主要涉及到 [Jekyll](#) 和 [Github Pages](#) 两个工具。Github Pages 是一个托管静态网站的平台，它支持使用 [Jekyll](#) 作为静态网站生成器。Jekyll 是一个开源的静态网站生成器，它可以将 Markdown 文件转换成 HTML 文件。当你在 Github 上 push 代码时，Github Pages 会自动为你生成并部署网站。

本文档主要介绍了如何使用 [Pelican](#) 搭建静态网站。Pelican 是一个开源的静态网站生成器，它支持使用 Markdown 文件来编写内容。Pelican 的官方网站是 [http://docs.getpelican.com/](#)。Pelican 支持使用 [python](#) 语言来编写主题和插件。

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

```
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ところ、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 が必要だったため build errored となり build fail となった script を修正して build fail を回避した

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

.travis.yml を push して github にアップロードして travis に clone して build して passing するまで build する

Travis-CI と Github

travis-ci を Github Pages で公開して Github を使えば Github の ssh key を github repo にアップロードして travis に key をインストールする

Github の 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 を生成する

2. 生成した 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 `<github-id/repo>` 2 `github` 3 `repo/repo` 4 `farseerfc/farseer` 5 `travis api` 6 `json` 7 `python` 8 `key` 9 `grep` 10 `sed` 11 `key` 12 `xargs -0 echo -en` 13 `"< >RSA"` 14 `openssl` 15 `travis.pem`

1 `pubkey` 2 `openssl` 3 `base64`

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

1 `token` 2 `secure`

1 `travis` 2 `token` 3 `archlinux` 4 `aur/ruby-travis` 5 `gems`

```
1 $ gem install travis
```

1 `Travis-CI` 2 `repo` 3 `travis status` 4 `Travis-CI` 5 `repo` 6 `build` 7 `repo`

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

1 `.travis.yml`

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

1 `Travis-CI` 2 `build` 3 `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

```

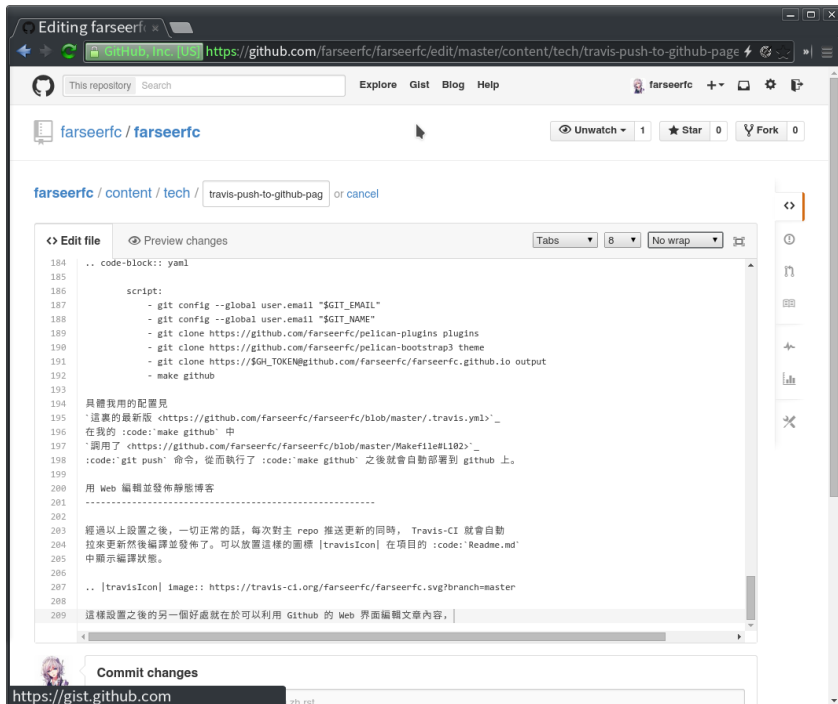
1. 在 `script` 部分添加 `git push --quiet` 命令，并设置 `$GH_TOKEN` 环境变量，URL 替换为 `https://$GH_TOKEN@github.com`。

2. 在 `after_success` 部分添加 `git push --quiet` 命令，并设置 `$GH_TOKEN` 环境变量，URL 替换为 `https://$GH_TOKEN@github.com`。

Web 部署

1. 在 `Travis-CI` 中配置 `build` 命令，并设置 `passing` 状态，Readme.md 文件。

2. 在 `Github` 中配置 `Web` 部署，并设置 `Github` 仓库，commit 命令，Travis-CI 仓库。



📄 Github 📄 Web 編輯器

編輯器 Github 編輯器 Android/iPhone 編輯器
github 編輯器

That is all, happy blogging ~