Nicholas Tan Jerome

§

Software Developer

Mountain Biking

§

How to setup Github User Page with Pelican

Posted: September 28, 2013 at 11:45 am

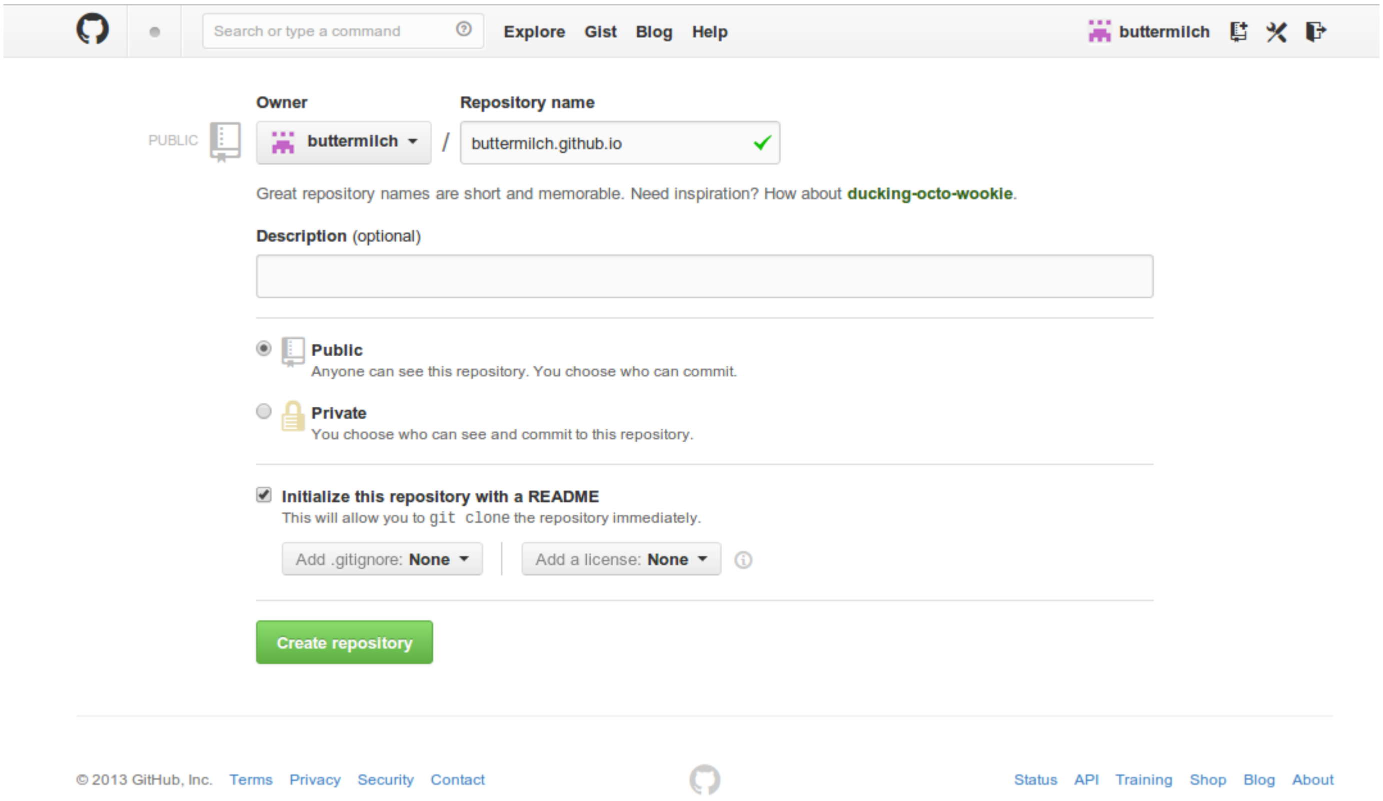
Github Pages has been introduced for a while and it's a good tool for static websites or just as blogs. Here, I will show how to set up a working static blog on github pages using Pelican. Pelican is a static site generator, written in Python. It strikes as a very interesting method as you can write your content purely in your favourite editor (in Markdown Flavour) and commit your post or style changes using Git. Each and every revision of your changes will be saved in this revision control system. If you are not into GUI Editor system, you might just find yourself your favourite tool.

This is a guide that just works without needing to clone anything.

Create a github page

Firstly, create a github page by creating a github repository and name it USERNAME.github.io
USERNAME refers to your github account username, it makes sense as to make your github user page unique. For example, I have a username called buttermilch and I

create a repository for this exercise and named it buttermilch.github.io



Owner: buttermilch / Repository name: buttermilch.github.io ✓

Great repository names are short and memorable. Need inspiration? How about **ducking-octo-wookie**.

Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

☑ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately.

Add .gitignore: **None** | Add a license: **None**

Create your working environment

Now, navigate to where you want your folder to be and clone the repository you just created. Here, I will clone the buttermilch repository. You can find your clone destination by going to the main page of your newly created repository. At right down of the page you will find a HTTP clone URL. Copy that and clone it to your desired folder.

```
$ git clone https://github.com/buttermilch/buttermilch.github.io.g
```

Next I will use virtualenv. Virtualenv is a great tool to create an isolate environment for your Python application. I prefer to have an isolated environment for each and everyone of my python project, you can proceed without using it, but I strongly suggest you start using it. If you don’t have it, you can get it as follow:

```
$ pip install virtualenv
```

Isolate this folder or application by creating an extra folder called ENV to store the virtualenv files.

```
$ cd buttermilch.github.io.git
$ mkdir ENV
$ virtualenv ENV
$ source ENV/bin/activate
```

This ENV folder is only interesting for us in our local environment, it does not need to be committed to the Github server. You will need to ignore this whole folder from your Git. You can do this by editing .gitignore file to exclude ENV. This file is at your highest hierachy of your application structure.

```
(ENV)$ vi .gitignore
```

Just type ENV/ on the first or last row of the file then commit it.

```
(ENV)$ git add .gitignore
(ENV)$ git commit -m "added ENV folder to ignore list"
```

Install and Setup Pelican

Github User page will display everything on master branch. So, I prefer to work in a new branch with my files and push only the output to master branch. Let’s create a new branch to work on, i.e. source

```
(ENV)$ git checkout -b source
```

Let’s install Pelican now. Base on the documentation, you can install pelican using “pip install pelican” or “easy_install pelican”. Try these two methods first, but if it does not work, as is happening to me, just install it directly from its github project page using the command below:

```
(ENV)$ pip install -e git://github.com/getpelican/pelican#egg=peli
```

Next, get the markdown as well, it is useful. I think you can choose between Markdown or reST, but I use Markdown.

```
(ENV)$ pip install Markdown
```

We now have Pelican, we can now kickstart our site by using the pelican-quickstart. This will launch an assisted installation process, if you are not seeing anything or it’s not working. There should be some problem with your pelican installation. Head back on top and see where you did it wrongly.

```
(ENV)$ pelican-quickstart
```

It will ask you many questions to set up your site. I use the following settings, mostly I just use the default, below are the answers I provided to launch this test application. You can refer to mine if you are stuck anywhere. The questions are largely straight-forward.

```
ere do you want to create your new web site? [.]
at will be the title of this web site? Buttermilch
o will be the author of this web site? Tony Stark
at will be the default language of this web site? [en]
 you want to specify a URL prefix? e.g., http://example.com   (Y/n)
 you want to enable article pagination? (Y/n)
w many articles per page do you want? [10]
 you want to generate a Fabfile/Makefile ... and publishing? (Y/n)
 you want an auto-reload & simpleHTTP ... and site development? (Y/n
 you want to upload your website using FTP? (y/N)
 you want to upload your website using SSH? (y/N)
 you want to upload your website using Dropbox? (y/N)
 you want to upload your website using S3? (y/N)
 you want to upload your website using Rackspace Cloud Files? (y/N)
```

Create a Markdown Post

Your path will be populated with many files now. What you will need is just to create a Post, so that you will see your post later. Navigate to content and create a test.md file.

The .md is the file type extension for a markdown file. The Markdown syntax can be found here. Markdown syntax is suppose to enlighten the work of writing an article and let the writer to focus mainly on the content, that being said, the syntax for formatting used in Markdown is really simple. I'll provide a sample markdown post for you to get on to speed.

```
(ENV)$ vi content/test.md
```

Populate your test.md as follow:

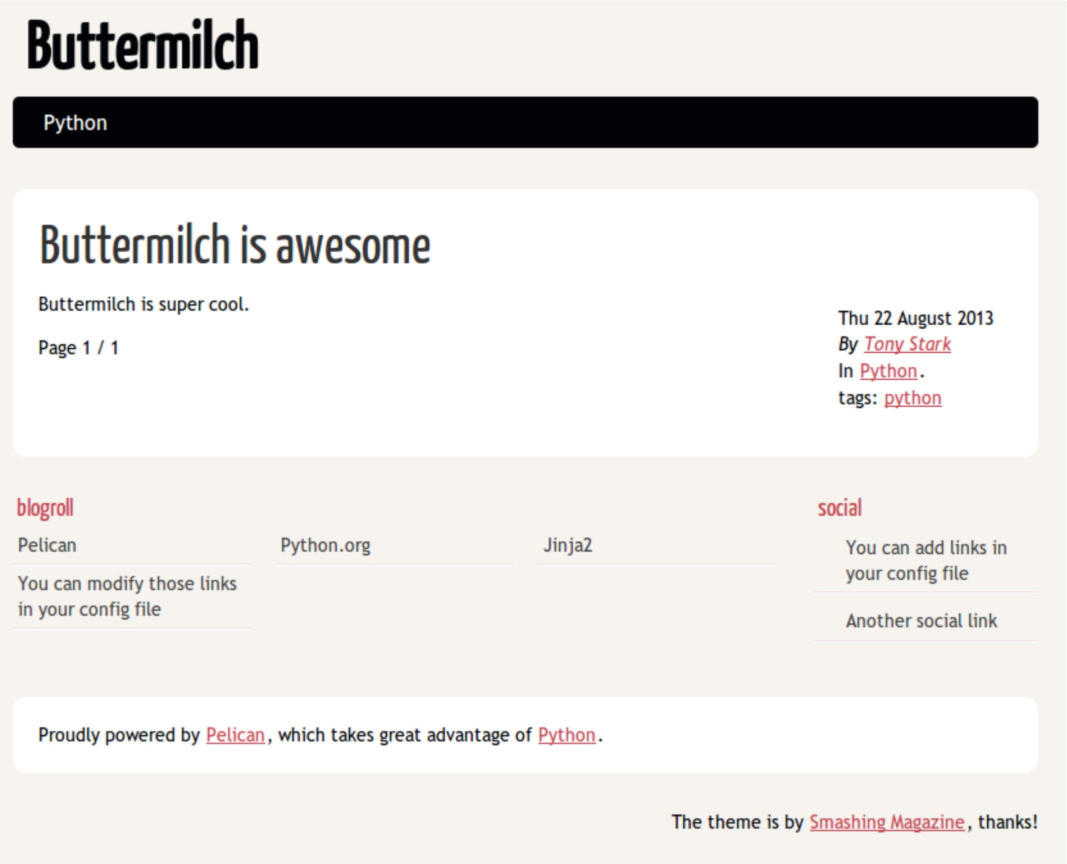
```
Title: Buttermilch is awesome
Date: 2013-08-22 16:08
Category: Python
Tags: python
Author: Tony Stark
Summary: Buttermilch is super cool.

Buttermilch is super cool.
```

Test your localhost site

So, we have a markdown post, we have installed Pelican and its friends correctly. Now we need to process or interpret the markdown file that we had just created. This can be done by simply typing the two commands below. First "make html" will processed the markdown file in your CONTENT folder. The processed files will be generated in OUTPUT folder. When there's no warnings nor error (good sign), we proceed with the next command "make serve", this launches your application from your localhost.

```
(ENV)$ make html
(ENV)$ make serve
```

Try launching your localhost:8000 from your web browser and your site should be there. I got my buttermilch. That’s great, everything is working fine. Give yourself a big clap.



Buttermilch

Python

Buttermilch is awesome

Buttermilch is super cool.

Page 1 / 1

Thu 22 August 2013
By Tony Stark
In Python.
tags: python

blogroll

Pelican

Python.org

Jinja2

You can modify those links in your config file

social

You can add links in your config file

Another social link

Proudly powered by Pelican, which takes great advantage of Python.

The theme is by Smashing Magazine, thanks!

Since everything is fine, I have my desired output, I want to save what I have at the moment. I do this by using Git to commit all my changes locally. This is a good practice to have revision control on your hard work. Do that by using the two commands below. Firstly, add your files to staging and secondly commit your staged files.

```
(ENV)$ git add .
(ENV)$ git commit -m "initial commit"
```

Launch your site to github

Remember that we are in our ‘source’ branch. We need to copy the necessary files to ‘master’ branch to get it on our github user page. You can directly push the output folder to master folder, but I prefer to use a script to help me do my work. I found out that I can use a script called ghp-import. This script export the content of the output folder to the ‘gh-pages’ branch. So, I created a ‘gh-pages’ branch, run the script, and then merge the content of the gh-pages to my ‘master’ branch. These steps are shown below in the commands.

if you don’t have ghp-import, you can get it simply by:

```
(ENV)$ pip install ghp-import
```

Create gh-pages branch and run the ghp-import script.

```
(ENV)$ git branch gh-pages
(ENV)$ ghp-import output
(ENV)$ git checkout master
(ENV)$ git merge gh-pages
(ENV)$ git push --all
```

Launch your github user page and wait patiently, your site should be live soon. I pushed all the branches to Git repository, but you can just push only master branch and source branch. We never need the gh-pages branch anyway.

In this post, the username I'm using is buttermilch, so I'll be expecting to see my static blog page on buttermilch.github.io

16 Comments Nicholas TJ 1 Login

Recommend 3 Share Sort by Best

Join the discussion…

Andrey Kaygorodov • a year ago

Thanks, it helped me.

1 • Reply • Share ›

Poisoner1024 • 4 months ago

Hello Nicholas, after I done "git push -- all", my local content and output directory is disappear, does it mean if I want to post another .md file to github repository, I have to run "pelican-quickstart" again?

• Reply • Share ›

Poisoner1024 → Poisoner1024 • 4 months ago

Sorry for this fool question, all of the files which generated by pelican is under git 'source' branch.

• Reply • Share ›

Nicholas Tan Jerome Mod → Poisoner1024 • 4 months ago

No worry, glad you succeeded :)

• Reply • Share ›

Hansun Lam • 6 months ago

Hi, nice post. Would you mind to tell me what the themes' name you use if it's from pelican themes

• Reply • Share ›

Nicholas Tan Jerome Mod → Hansun Lam • 6 months ago

This theme is called "notmyidea" by Smashing Magazines. I used this as it's the default theme at the time of this post.

^ | v • Reply • Share ›

Luis Jose Salazar • 7 months ago

Hello Nicholas, nice post. It help me but I used the command "pelican content" instead of "make html" and "make serve" since the last two instructions give errors that I could not resolve.

^ | v • Reply • Share ›

Adinan Paiva • a year ago

Following this tutorial, the end result is not expected. : /

By posting site on github pages, I believe that the styles/themes are not being sent correctly, the page that was generated me was the following.

http://adinanp.github.io/blog/

Someone is having the same problem?

^ | v • Reply • Share ›

Nicholas Tan Jerome Mod → Adinan Paiva • a year ago

Your user page is online, it's here http://adinanp.github.io/

seems ok to me.

^ | v • Reply • Share ›

Adinan Paiva → Nicholas Tan Jerome • a year ago

Yes this OK, after I comment here I reviewed the tutorial and I noticed there was created the repository without the suffix, such as "adinanp.github.io" before I had created only "adinanp" after search and fix this, everything worked as expected.

Sorry have not returned before and edited my comment. = (

Thank you for your attention, and congratulations on the great article,

^ | v • Reply • Share ›

Nicholas Tan Jerome Mod → Adinan Paiva • a year ago

The URL that has suffix is normally a project page, but for user page, there's no suffix at the end.

I'm happy to help you out. Glad it works with you. Happy blogging ^^

^ | v • Reply • Share ›

Joe Lotz • a year ago

Big questions: after deploying to github, I write more articles and make the html, do I then do the full??
git branch
ghp-import output
git checkout
git merge
git push

Or is there subset of commands?

^ | ˅ • Reply • Share ›

Nicholas Tan Jerome Mod ➦ Joe Lotz • a year ago

Hi Joe,

yes, after you "make html", you should run the rest of the commands. There might be some automated script that does this, but I did this only as a small test :)

Regards,
Nicholas

^ | ˅ • Reply • Share ›

This comment is awaiting moderation. Show comment.

Nicholas Tan Jerome Mod ➦ Akshay Pratap Singh • 2 years ago

Nice.

^ | ˅ • Reply • Share ›

El Diego Efe • 2 years ago

Thanks a lot for this tutorial. It helped me a lot to understand how to set up my blog on GitHub.

I think there's a little typo, where it says "source ENV/bin/active" should say "source ENV/bin/activate"

^ | ˅ • Reply • Share ›

Nicholas Tan Jerome Mod ➦ El Diego Efe • 2 years ago

Hi El Diego Efe,

thanks for reporting the typo. I just updated it and glad this tutorial helps you.

Cheers,
Nicholas

^ | ˅ • Reply • Share ›