GitHub Gist

JosefJezek / **how-to-use-pelican.md**
Last active a month ago

How to use Pelican on GitHub Pages

how-to-use-pelican.md

How to use Pelican on GitHub Pages 

Author: Josef Jezek

Install on Ubuntu

Installing Python tools

```
sudo apt-get install python-setuptools
sudo easy_install pip
sudo pip install virtualenv virtualenvwrapper
```

`vi ~/.bashrc` (maybe `virtualenvwrapper.sh` is in other location, change if necessary)

```
# Virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

Load file `.bashrc`

```
source ~/.bashrc
```

Create virtualenv with virtualenvwrapper

```
mkvirtualenv blog
workon blog
```

And to leave

```
deactivate
```

Installing Pelican

```
sudo pip install pelican markdown ghp-import shovel
```

Upgrading Pelican

```
sudo pip install --upgrade pelican markdown ghp-import shovel
```

Creating a blog on GitHub Pages

We need to make a respository. For example blog, so our blog url will be `user.github.com/blog` . If you want to set a custom domain see link: Setting up a custom domain with Pages

Prepare git

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
git config --global core.editor vi
```

Clone git repository

```
git clone git@github.com:user/blog.git
cd blog
```

Create in the root folder a file `.gitignore`

```
vi .gitignore
```

```
#Custom
output
pelicanconf_local.py

#Python
*.py[cod]

# C extensions
*.so

# Packages
*.egg
*.egg-info
dist
build
eggs
parts
bin
var
sdist
develop-eggs
.installed.cfg
lib
lib64

# Installer logs
pip-log.txt

# Unit test / coverage reports
.coverage
.tox
nosetests.xml

# Translations
*.mo
```

```
# Mr Developer
.mr.developer.cfg
.project
.pydevproject
```

Download Themes

Clone `pelican-themes` repository

```
git clone --recursive https://github.com/getpelican/pelican-themes themes
```

Create Pelican settings file

Create settings file called `pelicanconf.py` from sample `pelican.conf.py`

```
wget -O pelicanconf.py https://raw.github.com/getpelican/pelican/master/samples/pelican.conf.py
```

Add this variables

```
THEME = 'themes/bootstrap2'
OUTPUT_PATH = 'output'
PATH = 'content'
```

Period archives work best when the final path segment is `index.html` , see URL Settings

```
ARTICLE_URL = 'posts/{date:%Y}/{date:%m}/{slug}/'
ARTICLE_SAVE_AS = 'posts/{date:%Y}/{date:%m}/{slug}/index.html'
```

Custom Home page

Add this variables to `pelicanconf.py`

```
# Custom Home page
DIRECT_TEMPLATES = (('index', 'blog', 'tags', 'categories', 'archives'))
PAGINATED_DIRECT_TEMPLATES = (('blog',))
TEMPLATE_PAGES = {'home.html': 'index.html',}
```

Duplicated the `index.html` to `blog.html` in your template folder and add this lines:

```
{% set active_page = "blog" %}
{% block title %}{{ SITENAME }} - Blog{% endblock %}
```

Create `home.html` or use page override feature to use a Markdown page as your home page.

```
vi home.html
```

```
{% extends "base.html" %}
{% block content %}
<div class='page'>
  <div class="page-header"><h1>Home</h1></div>
  <div class="page-content">Content</div>
```

```
</div>
{% endblock %}
```

Creating post

```
mkdir -p content/posts/2013/07
vi content/posts/2013/07/welcome-all.md
```

```
Title: Welcome All
Slug: welcome-all
Date: 2013-07-22 19:19
Category: Python
Tags: pelican, publishing
Author: Josef Jezek
Summary: Short version for index and feeds

This is the content of my blog post.
```

Generate blog

To generate the static html we use:

```
pelican -s pelicanconf.py
```

This will take all the settings and apply, but if we want to override some settings we could do. For example to specify the ouput we use `-o` :

```
pelican -s pelicanconf.py -o /tmp/blog
```

Issues

- `WARNING: LOCALE option doesn't contain a correct value` see Ubuntu Locale

Then we have a new directory named `output` , our blog is there, so, to test it , insert there and run a simple server:

```
cd output
python -m SimpleHTTPServer 8000
# For Python 3
python -m http.server 8000
```

Point your browser to `localhost:8000` and you will see the blog.

Deploying the blog

Point your browser to you.github.com/blog , Awesome!!!

Resources

- http://blog.xlarrakoetxea.org/posts/2012/10/creating-a-blog-with-pelican
- http://docs.getpelican.com/en/latest/getting_started.html
- http://docs.getpelican.com/en/latest/tips.html#publishing-to-github
- http://raichev.net/blohg-to-pelican.html

- https://github.com/getpelican/pelican/issues/735#issuecomment-17674540

yasoob commented on Nov 24, 2013

That was purely awsome

michaelaye commented on Nov 29, 2013

I don't get it. The html files are produced in the output folder, but the output folder is in .gitignore, so I guess they should not be pushed to github, so how is the html transported to github?

ninly commented on Jan 18, 2014

@**michaelaye**: He doesn't go into detail here, but he installs ghp-import, which takes care of that by putting the output in a branch of the git repo. This part of the pelican docs explains how it works.

Kentoseth commented on May 5, 2014

@**michaelaye**: What you should normally do is to keep the output in a separate git repo. When you want to push to github pages, this becomes very easy. You just copy whatever is in the generated output to your specified repo for hosting the static files (eg. yourusername.github.io).

This is what I've read about others using the technique I mentioned.