

Pelican

Christian Külker

2024-07-14

Contents

1 Introduction	2
2 TL;TR Debian 10 Buster	2
3 Features	3
4 Installation Debian 10 Buster	3
5 Quickstart Debian Buster 10	4
6 Writing Content Debian 10 Buster	5
6.1 article-1.rst	5
6.2 about.rst	6
7 Compile	6
8 Testing	6
9 Upload	7
10 Create HTML	7
11 Publish (create HTML and mix with theme)	7
12 Themes	7
12.1 Enable A Theme	8
13 Differences Between Upstream (git) and Debian 10 Buster	9
14 Review	9
15 Urls	9

16 Interesting Themes	9
16.1 Tier 1 - tech collaboration	9
16.2 Tier 2 - blog	10
17 Examples	10
18 Taxonomy	10
19 History	10
20 Disclaimer of Warranty	10
21 Limitation of Liability	11

1 Introduction

This document serves as a guide for installing and configuring Pelican, a static site generator, specifically tailored as Free Open sources also to be used under Debian. This guide was tested on Debian 10 Buster. Pelican facilitates the creation of lightweight websites by allowing users to write content in reStructuredText or Markdown, which is then transformed into static HTML. The utility of Pelican extends to its ability to support multiple themes via Jinja2 templates, its compatibility with various content types and languages, and its provision for easy hosting due to the static nature of its output. Additionally, the guide details the process of setting up Pelican, including the installation of necessary dependencies, the initial setup of a new site using the `pelican-quickstart` utility, and the subsequent steps required to write, compile, and publish content. The document also discusses various operational themes and plugins that enhance the functionality of Pelican, alongside methods for content importation from different sources. The aim is to provide a structured approach to using Pelican for content management and website development under a Debian environment, focusing on technical descriptions and step-by-step procedures.

2 TL;TR Debian 10 Buster

```
# install
aptitude install pelican
/usr/bin/pelican --version
4.0.1 # Upstream version 4.2.0 as of 2020-02-05
# add content (reStructuredText)
...
```

```
# compile
pelican PATH_TO_SITE
```

3 Features

- Write content in reStructuredText or Markdown markup
- Completely static output is easy to host anywhere
- Themes that can be customized via Jinja2 templates
- Publish content in multiple languages
- Atom/RSS feeds
- Code syntax highlighting: Pygments
- Import from WordPress, RSS feeds, and other services
- Modular plugin system and corresponding plugin repository

4 Installation Debian 10 Buster

```
aptitude install pelican
# This will install pelican and its dependencies:
docutils-common pelican python3-blinker python3-docutils
python3-feedgenerator python3-jinja2 python3-markdown python3-markupsafe
python3-pygments python3-roman python3-tz python3-unidecode python3-yaml
xml-core
```

This gives

```
1 /usr/bin/pelican
2 /usr/bin/pelican-import
3 /usr/bin/pelican-quickstart
4 /usr/bin/pelican-themes
```

In the vanilla installation only the reStructuredText format including meta data:

```
1 My super title
2 #####
3
4 :date: 2020-02-05 19:57
5 :modified: 2020-02-05 20:02
6 :tags: Linux; documentation; pelican, blog software
```

```
7 :category: Computer
8 :slug: my-pelican-post
9 :authors: Christian Külker; Külker, Christian
10 :summary: Describes the pelican blog static site generator
11
12 This will be turned into :abbr:`HTML (HyperText Markup Language)`.
```

5 Quickstart Debian Buster 10

```
cd
mkdir -p scratch/pelican-new-site
mkdir -p scratch/pelican-new-site-upload
pelican-quickstart
Welcome to pelican-quickstart v4.0.1.

This script will help you create a new Pelican-based website.

Please answer the following questions so this script can generate the files
needed by Pelican.

> Where do you want to create your new web site? [.]
> What will be the title of this web site? pelican-new-site
> Who will be the author of this web site? Christian Külker
> What will be the default language of this web site? [en]
> Do you want to specify a URL prefix? e.g., https://example.com (Y/n)
http://localhost:8000
> Do you want to enable article pagination? (Y/n)
> How many articles per page do you want? [10]
> What is your time zone? [Europe/Paris]
> Do you want to generate a tasks.py/Makefile to automate generation and
publishing? (Y/n)
> Do you want to upload your website using FTP? (y/N)
> Do you want to upload your website using SSH? (y/N) y
> What is the hostname of your SSH server? [localhost]
> What is the port of your SSH server? [22]
> What is your username on that server? [root] $USER
> Where do you want to put your web site on that server? [/var/www]
/home/$USER/scratch/pelican-new-site-upload
> Do you want to upload your website using Dropbox? (y/N)
> Do you want to upload your website using S3? (y/N)
> Do you want to upload your website using Rackspace Cloud Files? (y/N)
```

```
> Do you want to upload your website using GitHub Pages? (y/N)
Done. Your new project is available at
↳ /home/$USER/scratch/pelican-new-site
```

This will create the structure (except `pages`):

```
1 |—
2 | content|
3 |   |— pages|—
4 | Makefile|—
5 | output|—
6 | pelicanconf.py|—
7 | publishconf.py|—
8 | tasks.py
```

The `pages` directory can be used for static non blog content and has to be created manually.

6 Writing Content Debian 10 Buster

```
1 |—
2 | content
3 | |— article-1.rst|
4 |   |— pages|
5 |     |— about.rst|—
6 | Makefile|—
7 | output|—
8 | pelicanconf.py|—
9 | publishconf.py|—
10 | tasks.py
```

6.1 article-1.rst

```
1 This is the first article
2 #####
3
4 :date: 2020-02-05 20:05
5
6 This is an example article for the pelican static site generator.
```

6.2 about.rst

```
1 About
2 #####
3
4 :date: 2020-02-05 20:10
5
6 This site is a test site for the pelican static site generator.
```

7 Compile

```
pelican /home/c/scratch/pelican-new-site
WARNING: Feeds generated without SITEURL set properly may not be valid
WARNING: No timezone information specified in the settings. Assuming your\
  timezone is UTC for feed generation. Check\
  http://docs.getpelican.com/en/latest/settings.html#timezone for more\
  information
CRITICAL: TypeError: not all arguments converted during string formatting
```

8 Testing

To actually see CSS (and a theme) your `SITEURL` needs to match the development server. Set `SITEURL` in `pelicanconf.py`:

```
1 SITEURL = 'http://localhost:8000'
```

Either `pelican --autoreload --listen`, `make serve` or `make devserver` to fire up a test server.

```
pelican --autoreload --listen

-> Modified: content, theme, settings. re-generating...
Done: Processed 2 articles, 0 drafts, 0 pages, 0 hidden pages and 0 draft\
  pages in 0.19 seconds.

127.0.0.1 - - [05/Feb/2020 20:21:11] "GET / HTTP/1.1" 200 -
WARNING: Unable to find `/favicon.ico` or variations:
| /favicon.ico.html
| /favicon.ico/index.html
```

```
| /favicon.ico/  
| /favicon.ico
```

9 Upload

```
cd  
cd scratch/pelican-new-site  
make help  
TODO
```

10 Create HTML

```
1 cd ~/scratch/pelican-new-site  
2 make html
```

To actually serve this, the theme needs to be in a sub directory `CSS` is served from `/css`. This creates HTML in the `~/scratch/pelican-new-site/output` directory.

Defining:

```
1 THEME = '/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea'
```

will not work. Use `make publish` for merging the theme and the HTML in the `~/scratch/pelican-new-site/output`.

11 Publish (create HTML and mix with theme)

```
cd ~/scratch/pelican-new-site  
make publish
```

A `img` directory in the `content` directory does not get copied.

12 Themes

```
/usr/bin/pelican-themes --help
/usr/bin/pelican-themes --list
simple
cd
cd scratch
git clone https://github.com/nasskach/pelican-blueidea.git
cd pelican-new-site
pelican-themes -v --install ../pelican-blueidea
Installing themes...
Copying '../pelican-blueidea' to
'/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea' ...
Cannot copy '../pelican-blueidea' to
`/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea':
[Errno 13] Permission denied:
'/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea'
# as root
/usr/bin/pelican-themes -v -i /home/c/scratch/pelican-blueidea
Installing themes...
Copying '/home/c/scratch/pelican-blueidea' to
'/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea' ...
```

This is a 1 to 1 copy including the `.git` directory !

12.1 Enable A Theme

The default theme is enabled (by default). This can be seen with the following command:

```
pelican --print-settings|grep -i theme
'THEME': '/usr/lib/python3/dist-packages/pelican/themes/simple',
'THEME_STATIC_DIR': 'theme',
'THEME_STATIC_PATHS': ['static'],
'THEME_TEMPLATES_OVERRIDES': [],
```

Add the line to `pelicanconf.py` :

```
1 THEME = '/usr/lib/python3/dist-packages/pelican/themes/pelican-blueidea'
```


13 Differences Between Upstream (git) and Debian 10 Buster

The “notmyidea” theme was removed from the Debian package due to the presence of non-free files. See #868047 [1] and #868049 [2] for further details. The Debian package defaults to the upstream “simple” theme instead.

Alternatively, you can download a copy of the “notmyidea” theme directly from upstream’s git repository [3], or other themes not shipped by default [4].

See: `/usr/share/doc/pelican/README.Debian`

14 Review

- no database
- static site generator
- python
- markdown
- code syntax highlighting
- current (updates 17 days ago, 2020-02-05)
- installation of Debian packages and testing is straight forward
- installed 3 years old theme seems to work out of the box
- content seem to be blog centric

15 Urls

1. Home: <https://blog.getpelican.com/>
2. Github: <https://github.com/getpelican/pelican>
3. Documentation: <https://docs.getpelican.com/en/stable/> on ReadTheDocs
4. Theme show room: <http://www.pelicanthemes.com/>
5. Elegant theme: <https://elegant.oncrashreboot.com/>

16 Interesting Themes

16.1 Tier 1 - tech collaboration

- elegant [99] comments
- Blueidea [4] See <https://github.com/nasskach/pelican-blueidea/tree/58fb13112a2707baa7d65075517c40>

16.2 Tier 2 - blog

- bootlex [4]
- bootstrap2 [4]
- pelican-blue [4] phone friendly

17 Examples

- 99 Pelican + Elegant Theme: <https://martin-thoma.com/configuration-files-in-python/>

Elegant is the only Pelican theme that search, live filter tags, zero clutter categories, lazy load comments, Mailchimp, FreeLists support and a customizable landing page. That's not all. Go through the documentation and you'll find several unique features and nuances that you'll not find in any other theme. Elegant is the only theme with comprehensive documentation and an automated build, deploy and release process.

18 Taxonomy

While level 2 taxonomy works fine (1.st level are displayed in header bar). When switching to 2 level taxonomy, only the 2nd level gets in the bar and the 1st level is not seen anywhere.

19 History

Version	Date	Notes
0.1.1	2024-07-14	Add history, formatting
0.1.0	2020-02-05	Initial release

20 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

21 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.