

Alternative blogging platform

John HR Schuster

Version 2.1b, 04/03/2017

Table of Contents

Document revision history	1
ASCIIDoctor	1
GitHub Research on ASCIIDoctor	1
Static vs Dynamic websites	1
Dynamic websites	2
Static web generators	2
Jekell web site generation	2
Jekyll Amin a GUI	3
Hugo web site generation	3
Awestruct	3
Base64 encoding of images	3
Caddy web server	4
Travis CI (Continuous Integration)	4
GIT	4
Alternative design concept	4
Web server platform	5
Automatic file change detection	5
Web server application	5
Test Bed for alternative design	5
Test bed starting points	5
WordPress to Hugo convertor	5
WordPress to Jekyll convertor	9
Standard WordPress export to WXR format	11
Writing on GitHub for WordPress	12
Use of Front Matter	13
Use of neDB for database if required	13
References	13

GeekMustHave is using WordPress for its blogging platform. It has been a good platform and with the introduction of Jetpack the maintenance and post maintenance has become easier. I use ASCIIDoctor for most of my documentation and was considering using it for blogging. This is when I discovered there is no love for ADCIIDoctor.

Document revision history

Document Revision History

[cols="2,1,1,5" options="header"]

Date	Version	Author	Description
04/08/2017	2.1b	JHRS	Initial version of document created

ASCIIDoctor

There doesn't seem to be any love for ASCIIDoctor. Everyone uses the Markdown format for doing web work. Markdown is fine but it was designed for HTML pages and doesn't deliver the ability to have real "Documents" with table of contents, tables and some of the extended features of ASCIIDoctor.

There are some disappointments with ASCIIDoctor

- Requires a Ruby installation and GEM installation, which is not possible in some hosting environments (InMotion)
- ASCiiDoctor is not as mature as Markdown
- Jekyll and Hugo web site generators both support Markdown natively and ASCIIDoctor with consider amount of pain
- There are quite a few useful extensions to ASCIIDoctor that improve it as a documentation and blogging tool.

GitHub Research on ASCIIDoctor

There are a number of projects on GitHub for ASCIIDcotor.

Static vs Dynamic websites

WordPress is a dynamic database driven CMS/Blogging platform. The database drives the pages and post with the automatic layout. The WordPress editor is web based and is the typical WYSIWYG model.

Static web sites just deliver HTML files to the user. They are simple in concept but difficult to

generate.

Dynamic websites

- Almost always have some overhead due to a database doing the **dynamic** part.
- Allow for quick continuous deployment because as you change the database the web site changes.
- Can be difficult to maintain due to the admin module and the database requirements.
- Applications like WordPress don't handle ASCIIDoctor formatted documents "At all"

Static web generators

- Have less overhead because there is no database involved, it just HTML text being delivered by the web server.
- Static websites have to be generated from some form of source document, Markdown or ASCIIDoctor (Poorly)
- Static websites use **Front Matter** which is a block of text at the beginning of the document to help process the document.
- Static generators require considerable knowledge to implement, maintain and use.
- Have many more files than a dynamic site where much of the content and images are in the database
- It is **not** possible to have Comments to web posts because of the lack of a database

Jekyll web site generation

- Very stable and established platform for generating **static** websites.
- Ruby based application and ASCIIDoctor is a Ruby based application.
- Deploy to GitHub very easily.
- Importer feature to move a WordPress site into it.
- Plugins to extend the feature and functions
- Reference YouTube [Jon doesn't like Jekyll, June 2014, 22 minutes](#)
 - Wanted an individual CMS
 - They didn't use Markdown
 - No database lessens the hacking probability
 - Just "text" is cool??
 - Collaboration not built in, GIT comes to rescue
- Reference: Blog post [Jekyll Issues aren't as Bad as You Think, 9minute read](#)
 - Some of the themes are out of date and will generate errors

- Lanyon is a great theme with side bar menus

Jekyll Amin a GUI

- Word[press like GUI to manage content
- Work in progress
- Use Grammarly to checj spelling and grammar.
- Reference: [Jekyll Admin](#)
- Meant for local usage, server version from [Siteleaf](#)

Hugo web site generation

- Newcomer to static generation but has a number of advatages over Jekyll.
- Very quick, about 5-10x quicker than Jekyll.
- Written in Go and there is very little infrastructure to stand up.
- Built in **liverload** automatically refresh browser on change to web website
- WordPress plugin to convert posts, pages, taxonomies, metadata into markdown and YAML
- Plenty of other importers for other CMS or Blogging platforms

Awestruct

- Ruby based site generator with specific support for ASCIIDoctor
- Various extensions done in Ruby are available
- Posts extension - Posts extension scans pages within a particular subtree of your site, and if they match the format of YYYY-MM-DD-post-title, they are registered as blog posts, and slightly manipulated.
- Can use RSync to keep content uptodate
- last update Feb 2016
- Super complex set-up, HAML files everywhere
- Hardly any YouTube traiuning content

Base64 encoding of images

At first I thought this might be a good way to elminate the **images** directory in the ASCIIDoctor document I have. This is not the case for a number of reasons.

- Difficult to add endcoding automatically to a ASCIIDoctor document in the HTML generation phase
- Method to do the encoding after HTML generation are clugy and difficult on Windows

platforms.

- The encoded text result is larger than the Binary, bad for mobile use
- The encoding process is additional overhead and possible place for sync issues

Caddy web server

- This is a new light web server available on all platforms
- It is HTTPS right out of the box
- Supports Markdown natively but no support for ASCIIDoctor
- Websockets and FastCGI support
- Call it as a command from any directory
- Digital Ocean sponsored and interconnect
- Written in Go
- GIT is a caddy extension

Travis CI (Continuous Integration)

- Sync GitHub pages with a deployment process.
- Creates a VM Heroku does some processes
- Build phase tools set is unknown to me at this time.



Additional research is needed in this CI area.

GIT

More and more so I see heavy integration with Static Generation packages with GitHub. These applications can use GitHub as you Blog platform no other server required.

When I'm writing documents using ASCIIDoctor that I know will go through many revision I will add a GIT repository to the directory.

It would be nice if there was a way to use GIT to publish to the web site.

Alternative design concept

The GeekMustHave blogging site would become a single directory on Dropbox. This would make the content available from any of my systems with that directory sync'd. There would be a GIT repository for the top level directory.

Web server platform

The web server platform will be a linux based server or service. It will need to have ASCIIDoctor installed on it.

Automatic file change detection

The server will need to have a mechanism that detects when a file change or new file has occurred. This process will then copy the staged files to the production area. This process would also translate ADOC files into HTML files

Web server application

While using the Linux existing Apache it might be better to use Caddy which can be tested across platforms.

Test Bed for alternative design

The testbed for the alternative design has to work on Windows as well as on Linux.

I tend to use command line interface for doing work so most of the scripting will be done in PowerShell.

Test bed starting points

To get started quickly I'm going to install the WordPress Plug in to generate my entire GeekMustHave website into Hugo ready files. This step will give me a large set of posts in which to do proper scale testing.

WordPress to Hugo convertor

This is a [package](#) written by Schumacher.FM

Generate the ZIP file from within GitHub.

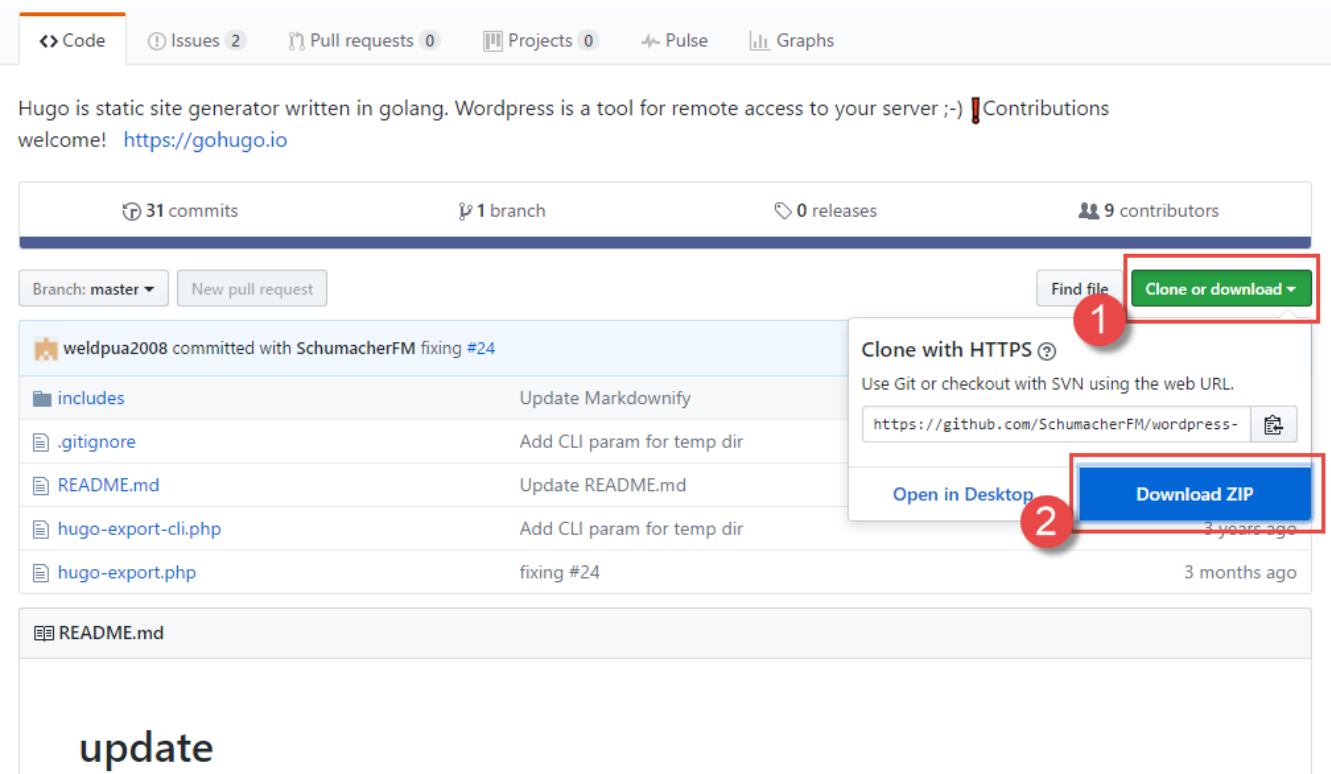


Figure 1. GitHub WordPress to Hugo convertor

Use Filezilla to move ZIP to the `wp-content/plugins` directory of GeekMustHave. This step isn't necessary but I did it anyway.

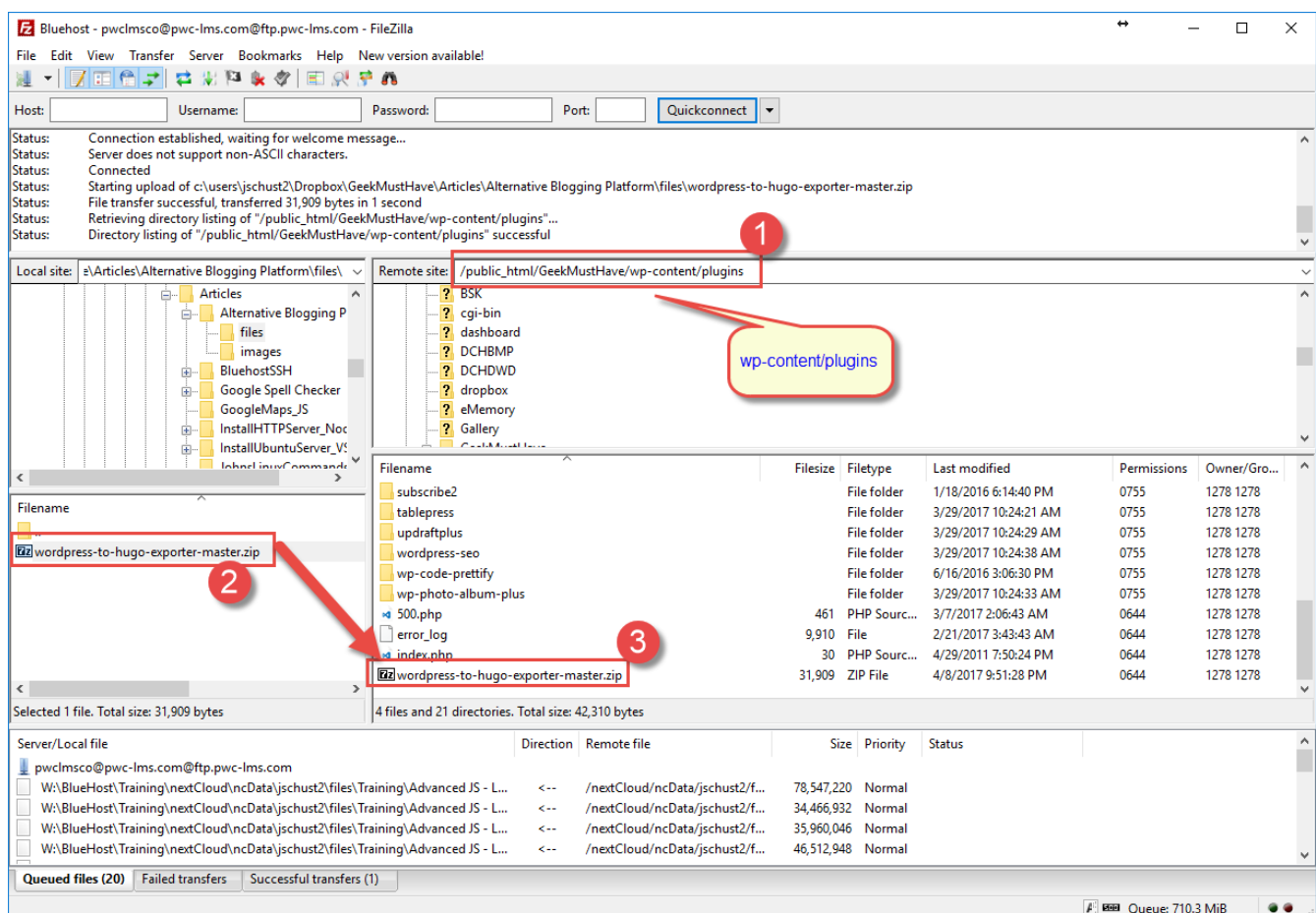


Figure 2. Transfer ZIP to WP server

You can upload the zip directly from the WP admin page under plugins.

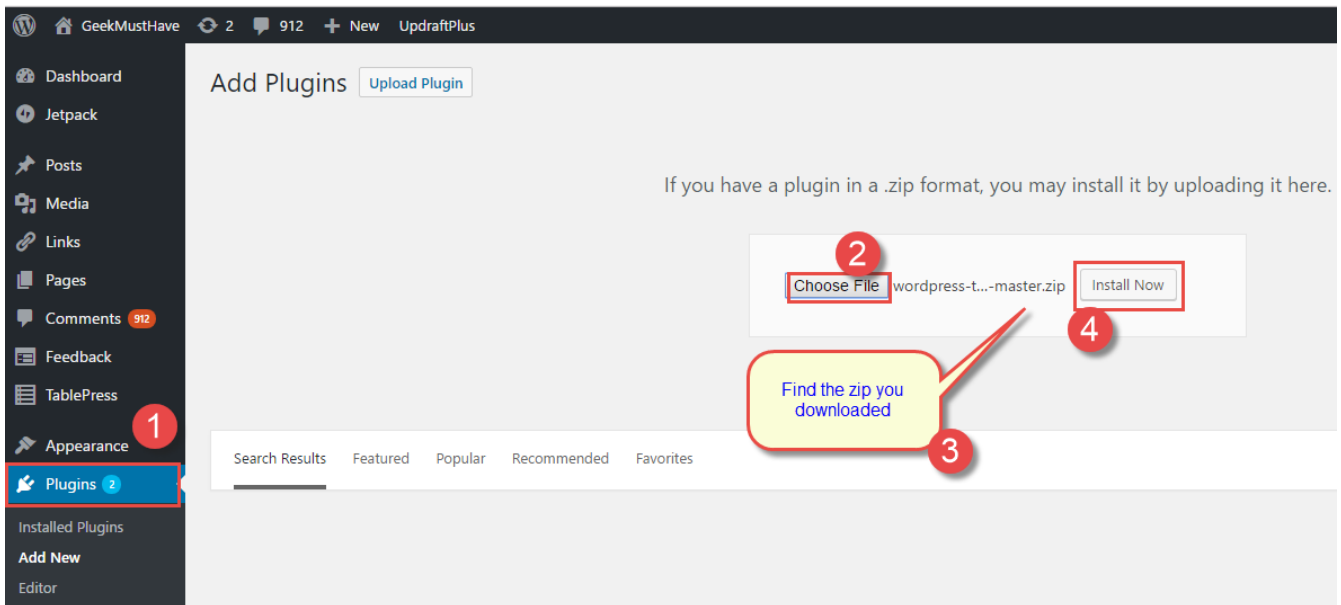


Figure 3. Install from Admin in WP

After it has installed you need to **activate** it.

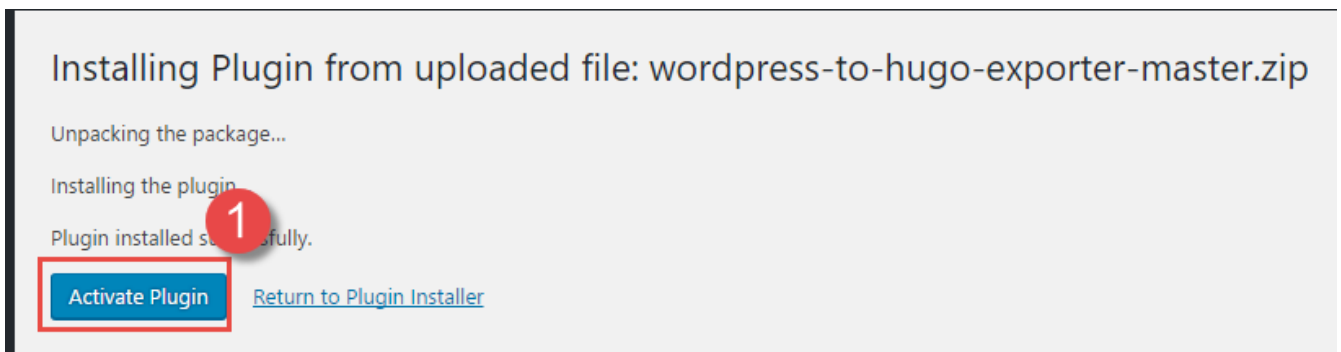


Figure 4. Activate Hugo plugin

Locate the place where it was installed and run it.

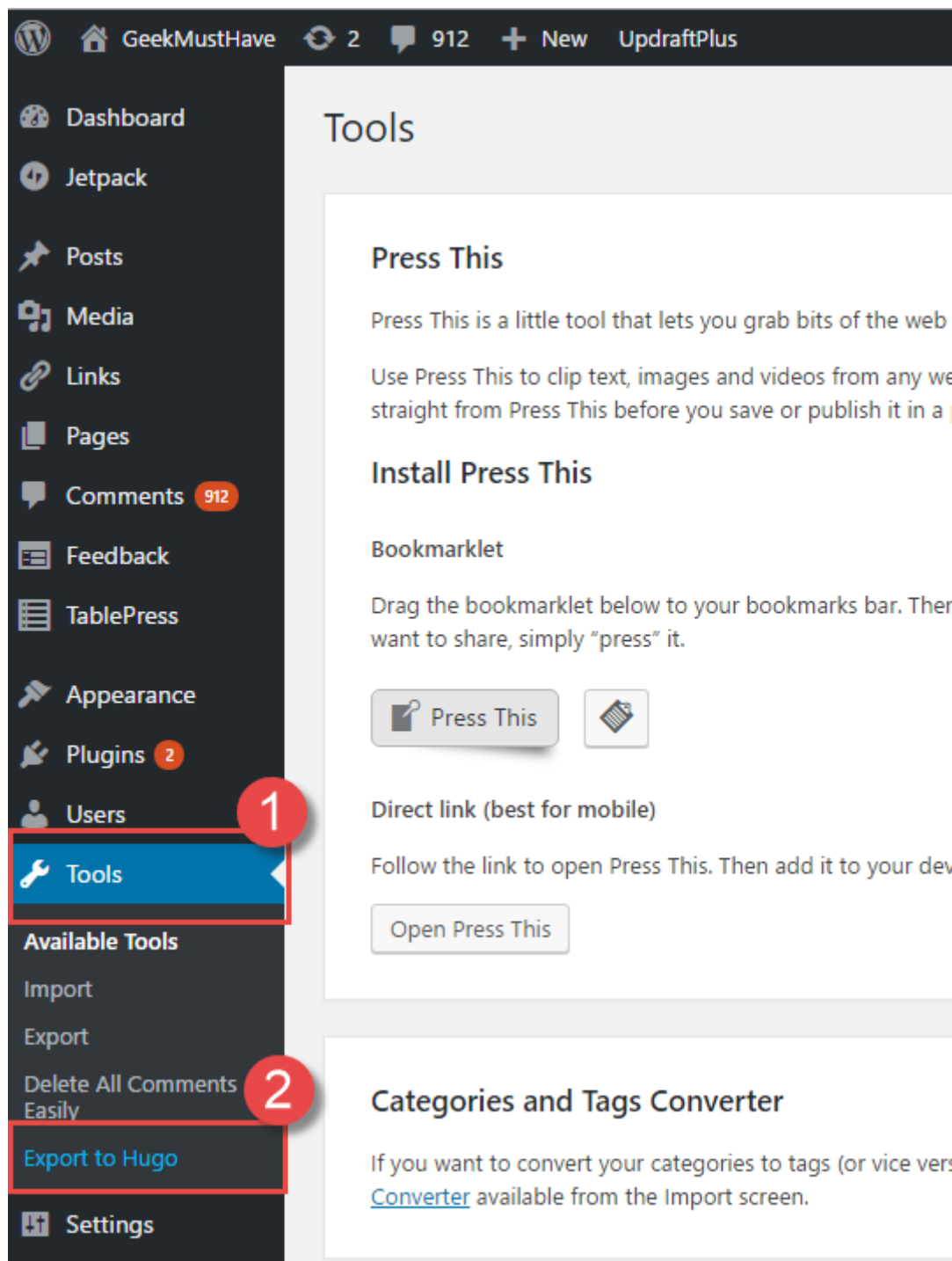
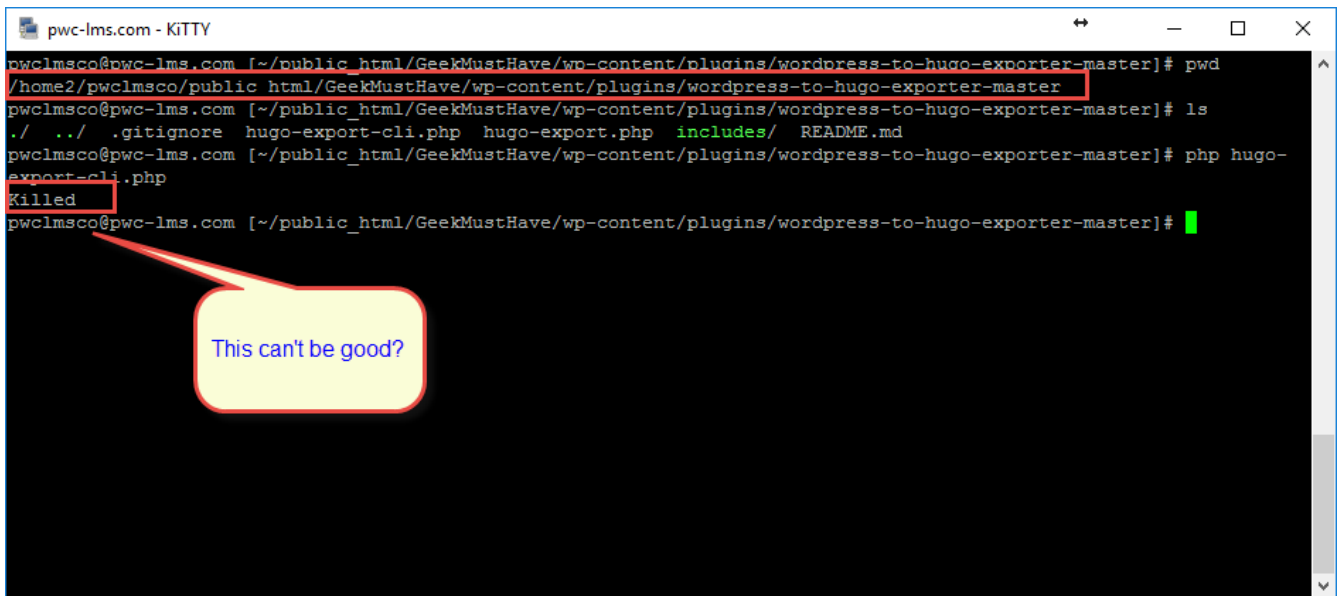


Figure 5. Run tool

After I ran it came back with a blank screen. Did it actually run or did it just wimp out? Where did the output go? What is it named?

Attempted to run the tool from the Terminal, it failed as well.



A terminal window titled 'pwc-lms.com - KITTY' showing a series of commands and their outputs. The user is in the directory `~/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master`. They run `pwd`, which shows the full path. Then they run `ls`, listing files including `hugo-export-cli.php`, `hugo-export.php`, `includes/`, and `README.md`. Finally, they run `php hugo-export-cli.php`, which results in a `Killed` message. A red box highlights the `Killed` message, and a yellow speech bubble points to it with the text 'This can't be good?'.

```
pwc-lms.com [~/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master]# pwd
/home2/pwclmsco/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master
pwc-lms.com [~/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master]# ls
./ ../ .gitignore hugo-export-cli.php hugo-export.php includes/ README.md
pwc-lms.com [~/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master]# php hugo-export-cli.php
Killed
pwc-lms.com [~/public_html/GeekMustHave/wp-content/plugins/wordpress-to-hugo-exporter-master]#
```

Figure 6. CLI Version of tool failed

Seems to have failed at this point in the PHP. I am way out of my league here.



A snippet of PHP code from a file. The code defines a `Hugo_Export_Command` class that extends `WP_CLI_Command`. It has a `__invoke()` method that calls `$je->export();`. A red box highlights this line, and a yellow speech bubble points to it with the text 'Blowing up at this line'.

```
517     }
518 }
519
520 $je = new Hugo_Export();
521
522 if (defined('WP_CLI') && WP_CLI) {
523     class Hugo_Export_Command extends WP_CLI_Command
524     {
525         function __invoke()
526         {
527             global $je;
528             $je->export();
529         }
530     }
531     WP_CLI::add_command('hugo-export', 'Hugo_Export_Command');
532 }
533
534
535
536
537
```

Figure 7. wp2hugokilled

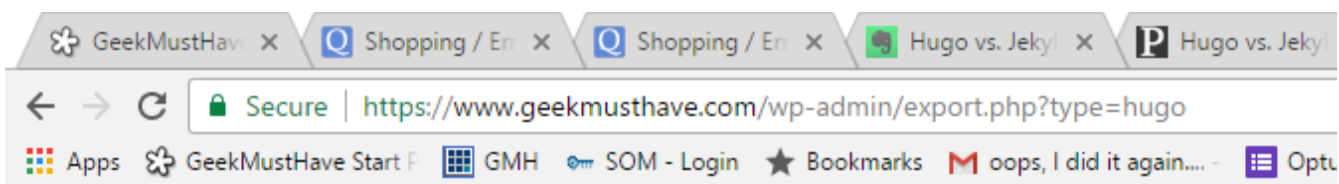


Figure 8. Results of Hugo conversion

WordPress to Jekyll convertor

This convertor look much more mature and established, the one above was a GitHub versions. The WP 2 Hugo is an external ZIP because it probably hasn't been tested on the current version of WP. Version 4.7.3

This plugin is available [here](#).

The screenshot shows the WordPress plugin page for 'Jekyll Exporter'. At the top, the WordPress logo is next to the plugin title. A red arrow points from the logo to a yellow callout box that says 'Of course, everything down this path to use a Static Generator is just crap'. Below the title are tabs for 'Description', 'Changelog', and 'Reviews'. A yellow warning box states: 'Warning: This plugin has not been tested with your current version of WordPress.' The main description lists features like converting posts, pages, and settings to Jekyll format, using the `the_content` filter, and generating a `_config.yml` file. A second yellow callout box points to the description, saying 'Have no idea what version of PHP being used'. On the right, metadata includes Version 2.2.0, Author Ben Balter, Last Updated 5 months ago, Requires WordPress Version 4.4 or higher, Compatible up to 4.6.0, and Active Installs 800+. The average rating is 4 stars (based on 4 ratings). A star rating breakdown shows 3 4-star ratings, 0 3-star ratings, 1 2-star rating, 0 1-star ratings, and 0 0-star ratings. A 'CONTRIBUTORS' section lists Ben Balter. A note at the bottom states: 'Many shared hosts may use an outdated version of PHP by default. WordPress to Jekyll Export requires PHP 5.5 or greater.' Below this, it says 'If you get an error message that looks like `unexpected T_STRING`,'. An 'Install Now' button is at the bottom right.

WordPress Jekyll Exporter

Of course, everything down this path to use a Static Generator is just crap

Description Changelog Reviews

Warning: This plugin has not been tested with your current version of WordPress.

- Converts all posts, pages, and settings from WordPress for use in Jekyll
- Export what your users see, not what the database stores (runs post content through `the_content` filter prior to export, allowing third-party plugins to modify the output)
- Converts all `post_content` to Markdown Extra (using Markdownify)
- Converts all `post_meta` and fields within the `wp_posts` table to YAML front matter for parsing by Jekyll
- Generates a `_config.yml` with all settings in the `wp_options` table
- Outputs a single zip file with `_config.yml`, pages, and `_posts` folder containing `.md` files for each post in the proper Jekyll convention
- No settings. Just a single click.

A Note

Many shared hosts may use an outdated version of PHP by default. WordPress to Jekyll Export requires PHP 5.5 or greater.

If you get an error message that looks like `unexpected T_STRING`,

Version: 2.2.0
Author: [Ben Balter](#)
Last Updated: 5 months ago
Requires WordPress Version: 4.4 or higher
Compatible up to: 4.6.0
Active Installs: 800+
[WordPress.org Plugin Page »](#)
[Plugin Homepage »](#)

AVERAGE RATING
★★★★★
(based on 4 ratings)

4 stars 3
3 stars 0
2 stars 1
1 star 0

CONTRIBUTORS
[benbalter](#)

Install Now

Figure 9. Details on WP 2 Jekyll

It can be installed directly from the Admin panel.

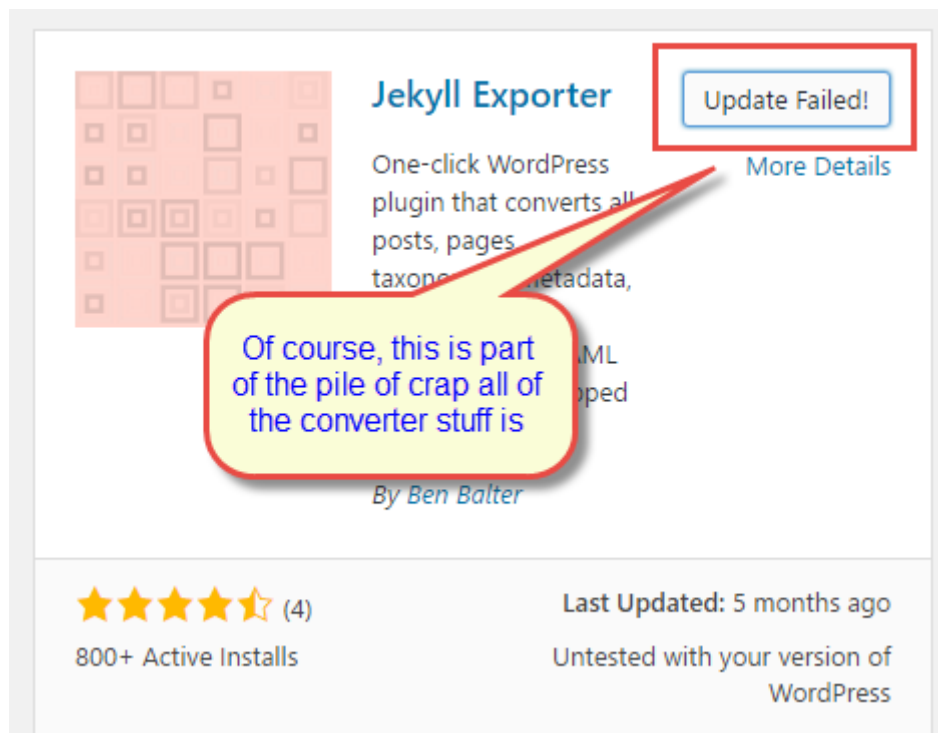


Figure 10. Just Another failed

Standard WordPress export to WXR format

When you click the button below WordPress will create an XML file for you to save to your computer.

This format, which we call WordPress eXtended RSS or WXR, will contain your posts, pages, comments, custom fields, categories, and tags.

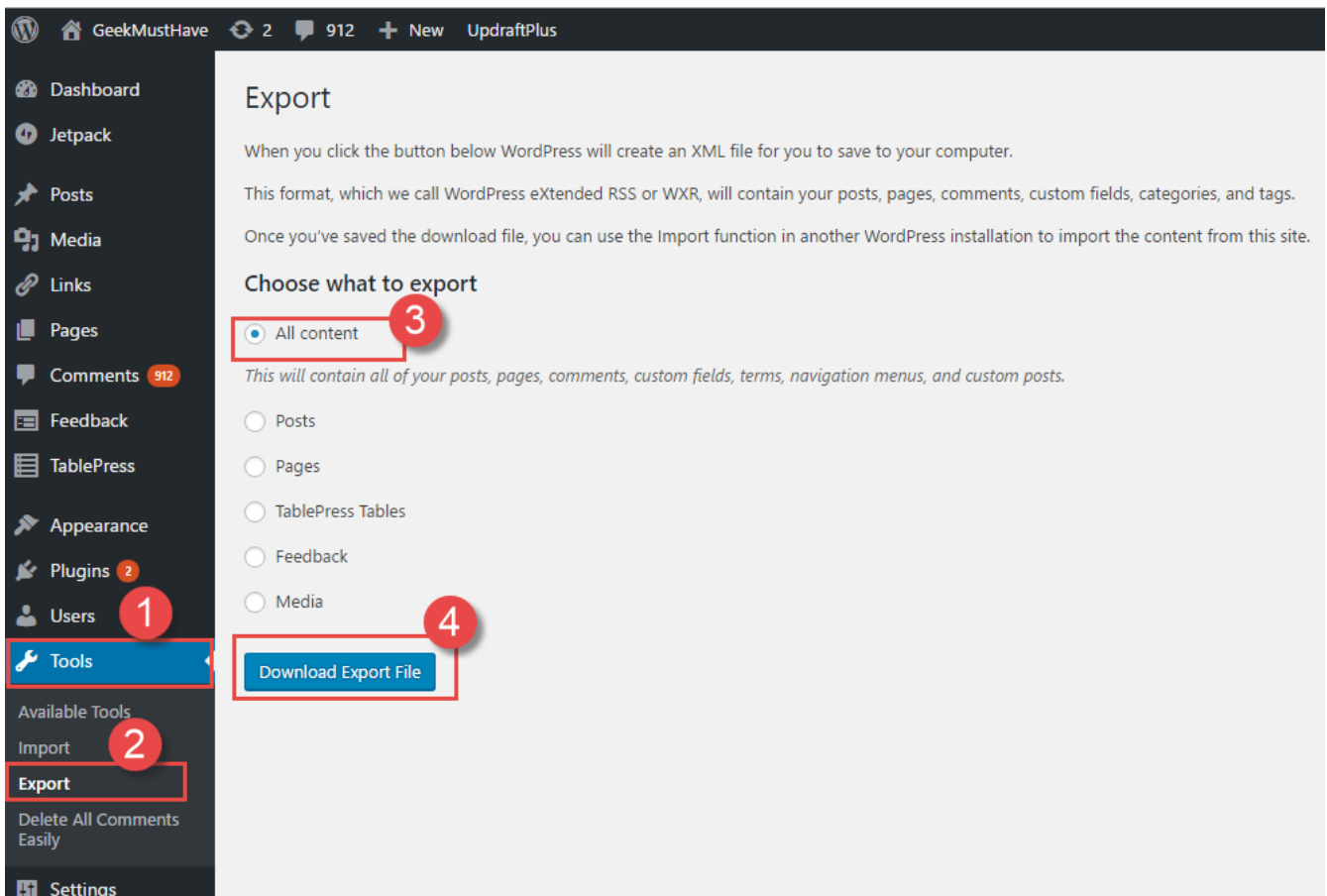


Figure 11. Standard WXR export from Tools → export

This also poops out with a blank page.

Writing on GitHub for WordPress

Is the a possible solution to the posting problem. I am sure that GitHub not accepts ASCIIDoctor documents. The link on the install page failed to connect to GitHub, no making me feel comfortable.

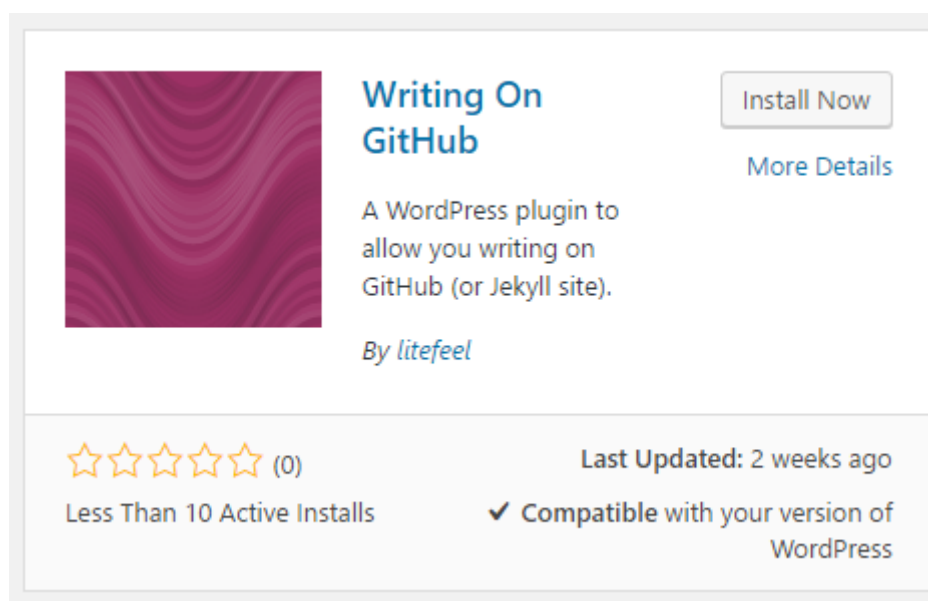


Figure 12. Writing on GitHub

Use of Front Matter

Front matter defined how the post is to be processed. Both Jekyll and Hugo use Front Matter for this. Each has it own keywords used for loading and tagging.

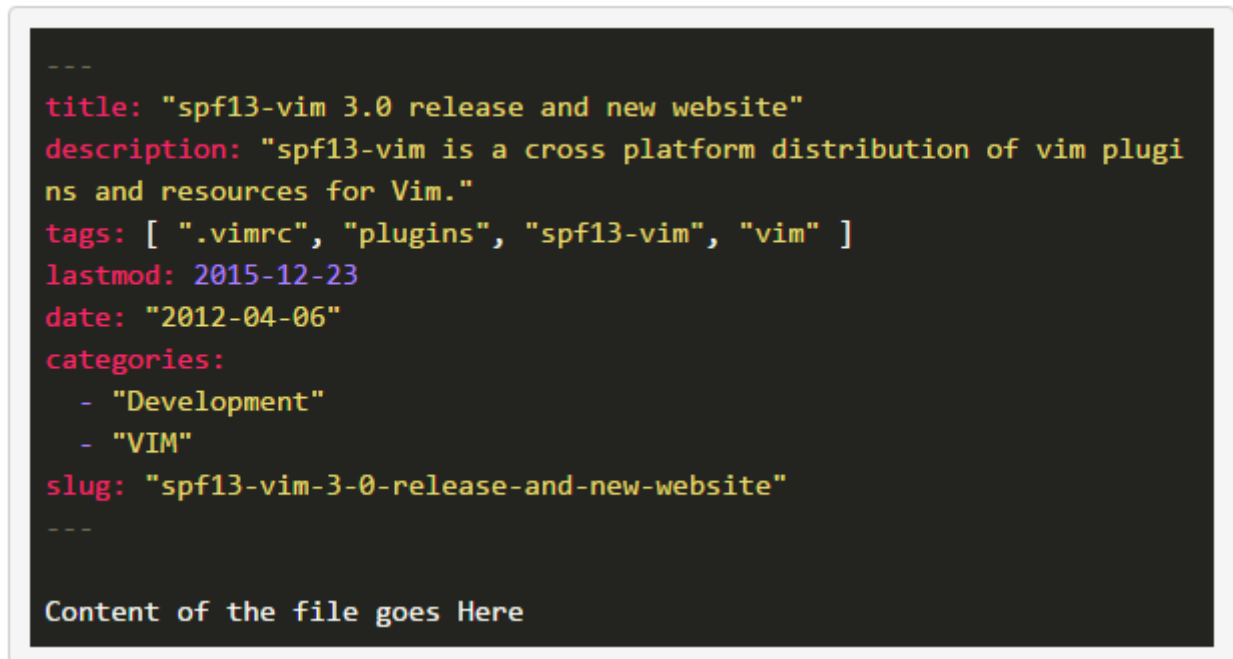


Figure 13. YAML Front Matter Example

Our Front Matter would be enclosed in the `--- YAML ---`

Use of neDB for database if required

Should the solution require a DB we would initially use neDB which emulates Mongo but is a file system based solution.

neDB would be good for the limited number of hits GMH gets

References

[Comparing Static Site Engines with Brian Rinaldi Feb 2015 54:26](#)

- Customize - Templates
- Dynamic content
 - Comments - Disqus
 - Calendar - Google
 - Forms - Wufoo Google
- Process is intended for developers - That's me
- 384 static generation engines???
- Engines

- Jekyll (Github)
 - 02/15 No Windows support
 - Automatic browser refresh
 - YAML Front Matter
 - Summary separator
 - Liquid template
- Middleman
 - Ruby based, works with Windows
 - Livereload a few extra steps
 - ERB templating
 - Strange syntax for templates
- Harp