Alternative blogging platform

John HR Schuster

Version 2.1b, 04/03/2017

Table of Contents

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 discoverfed 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 beacuse 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 begining of the document to help process the document.
- Static generators require considerable knowledge to implement, maimtain 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 becuase of the lack of a database

Jekell 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 feayure to move a WordPress site into it.
- Plugins to extend the feature and functions
- Reference YouTube Jon doesn't like Jekylll, June 2014, 22 minutes
  - Wanted an indivdual CMS
  - They didn't use Markdown
  - No database lessens the hacking probability
  - Just "text" is cool??
  - Collabaration 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[ress 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 bkox
- Supports Markdown natively but no support for ASCIIDoctor
- Websockets and FastCGI support
- Call it as a command from any directory
- Digital Ocean spopnsered 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 unknow 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 respository 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 mechanisum that detects when a file change or new file has occured. 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 Liniux 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 donje in PowerSell.

Test bed starting points

To get started quickly I'm going to install the WoprdPress 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 fomr withing GitHub.

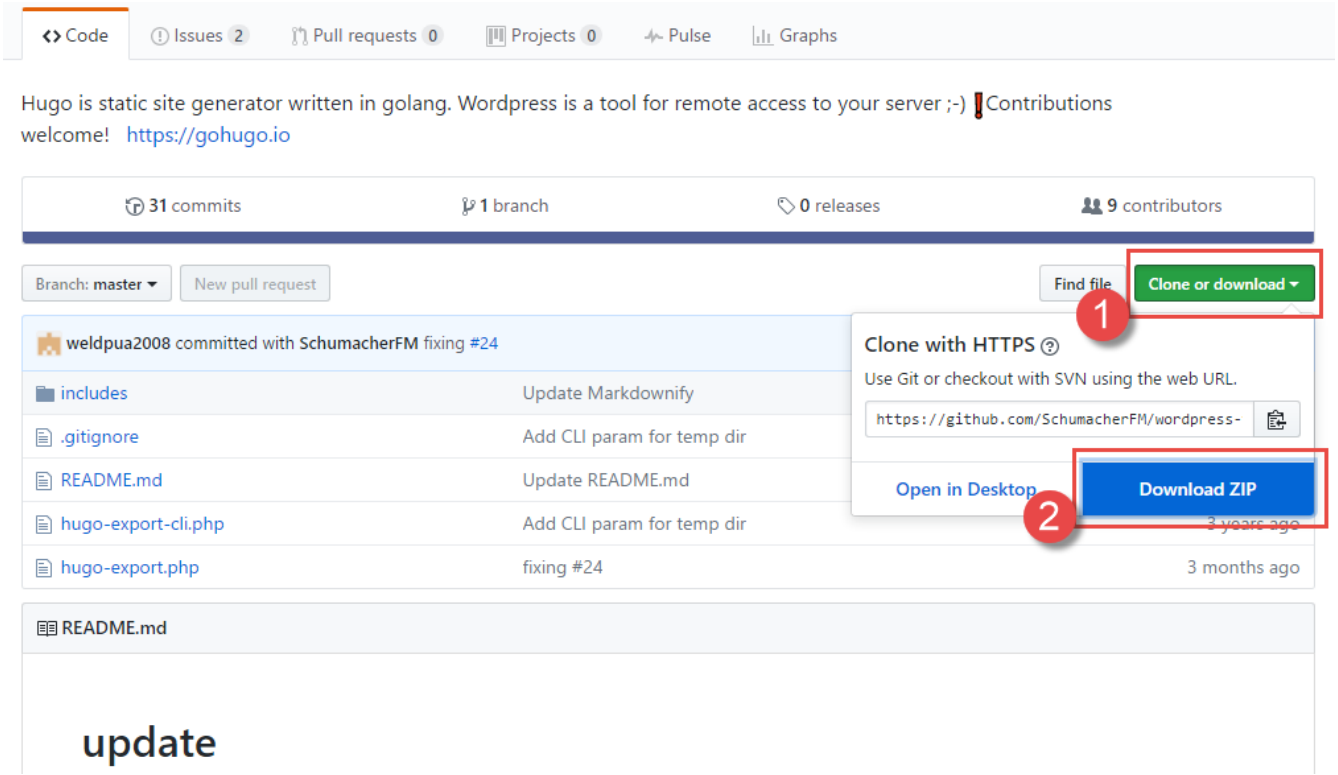

Figure 1. GitHub WordPress to Hugo convertor

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

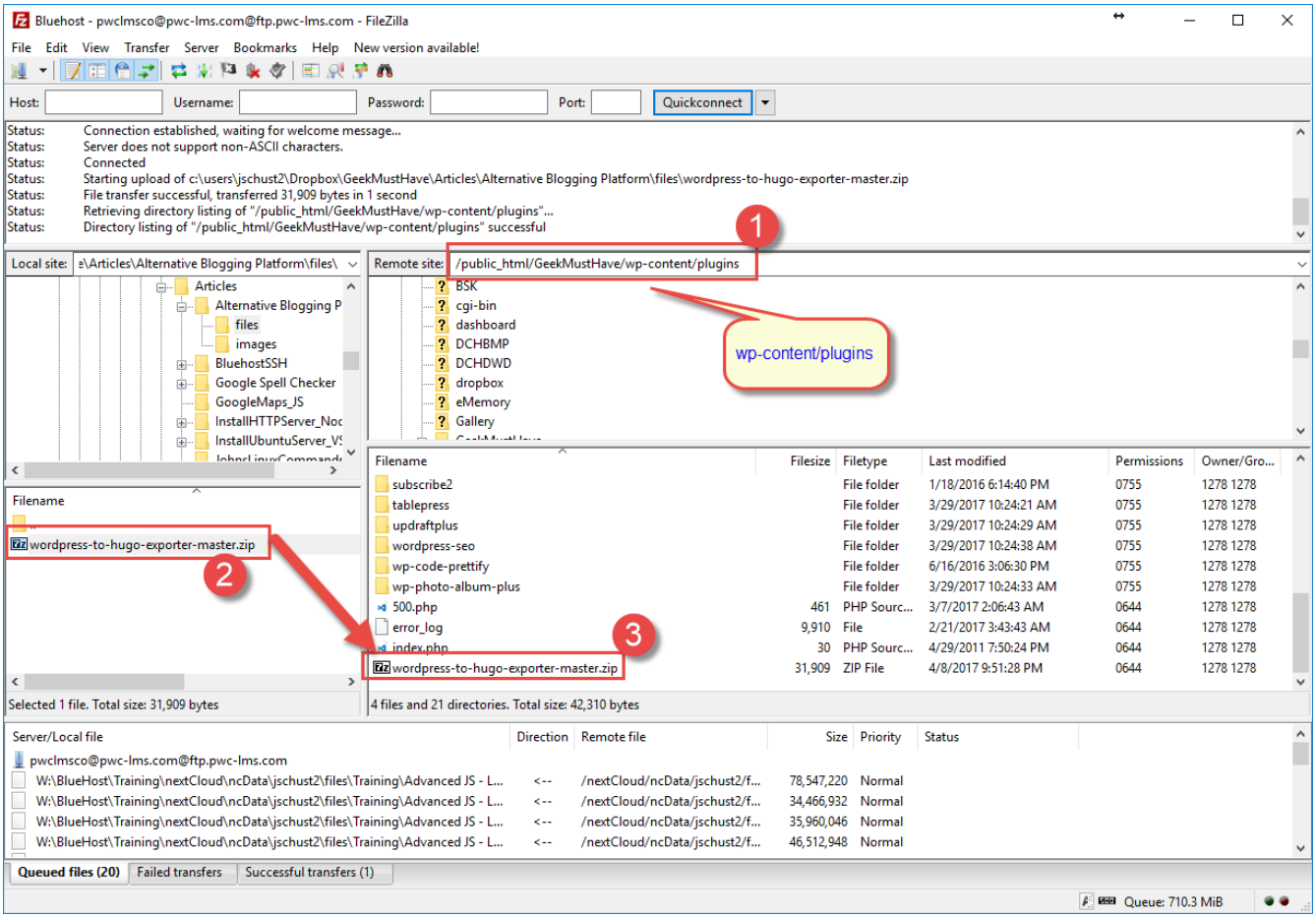

Figure 2. Transfer ZIP to WP server

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

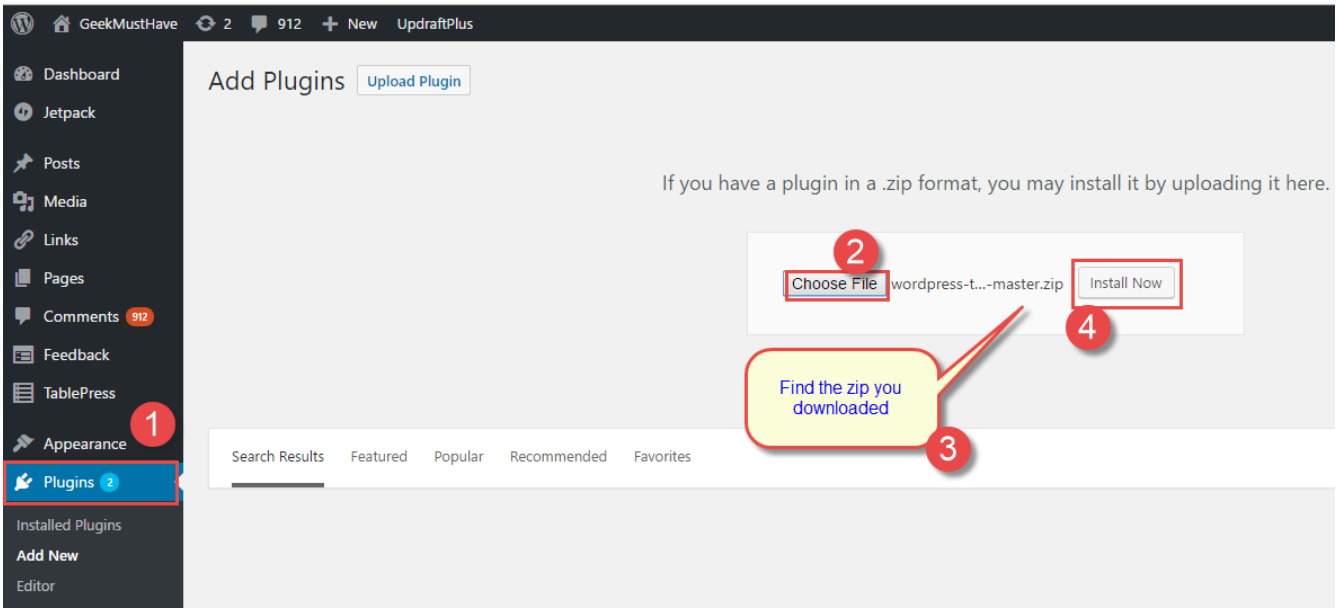

Figure 3. Install from Admin in WP

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

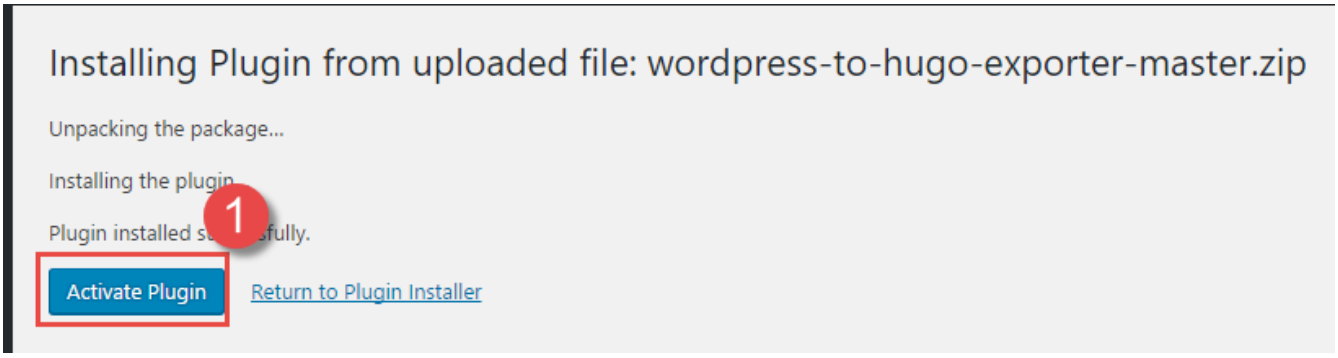

Figure 4. Activate Hugo plugin

Locate the place where it was installed and run it.

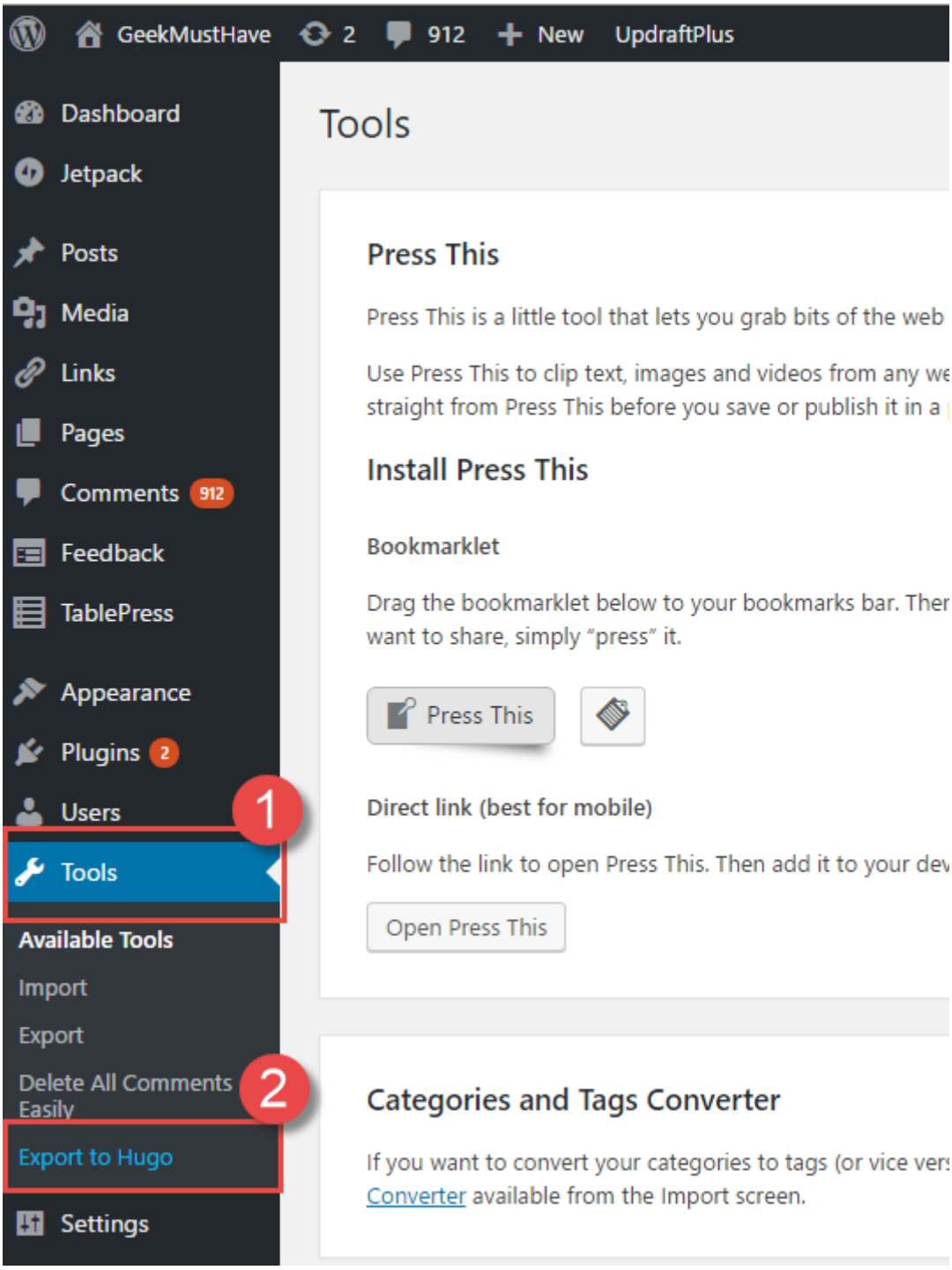

Figure 5. Run tool

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

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

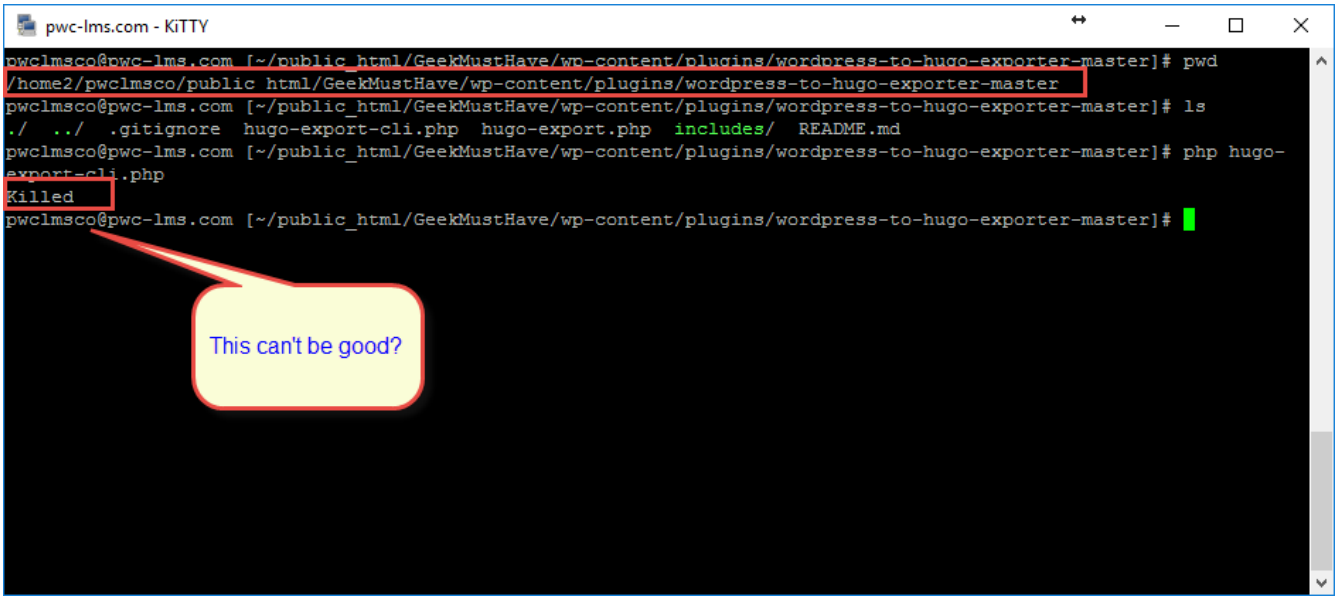

Figure 6. CLI Version of tool failed

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

Figure 7. wp2hugokilled

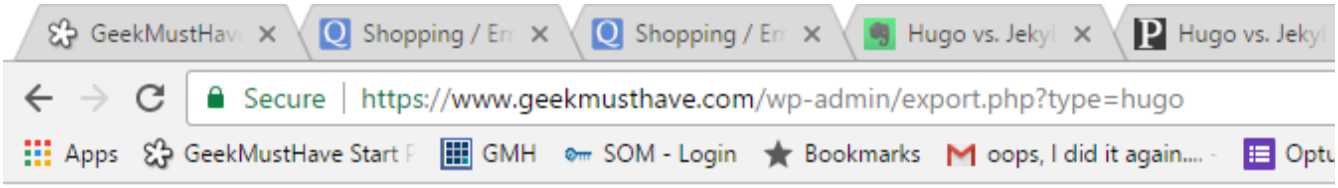

Figure 8. Results of Hugo conversion

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.

Figure 9. Details on WP 2 Jekyll

It can be installed directly from the Admin panel.

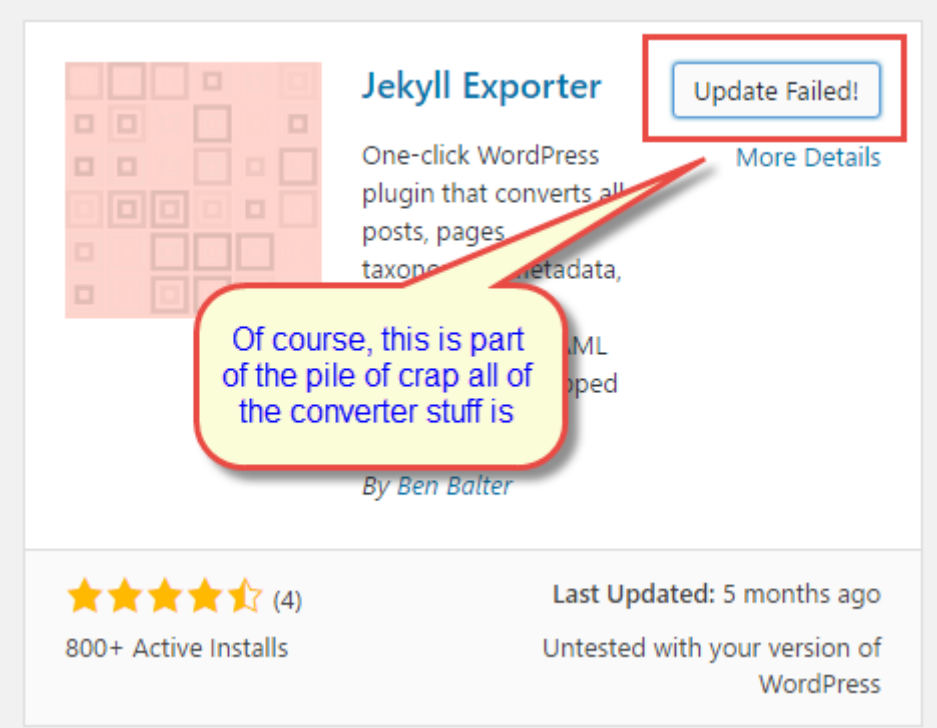

Figure 10. Just Another failed

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.

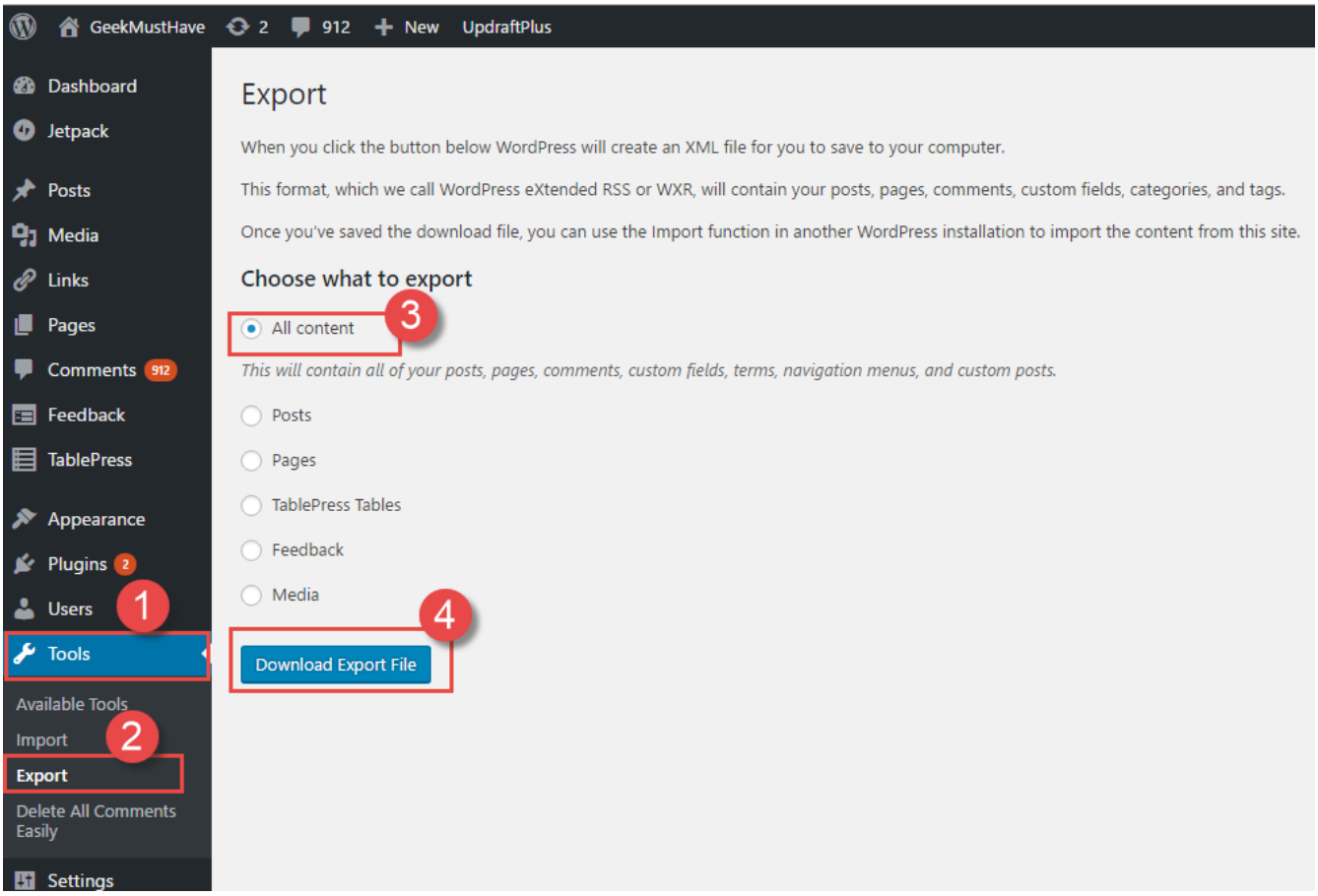

Figure 11. Standard WXR export from Tools→ export

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 confortable.

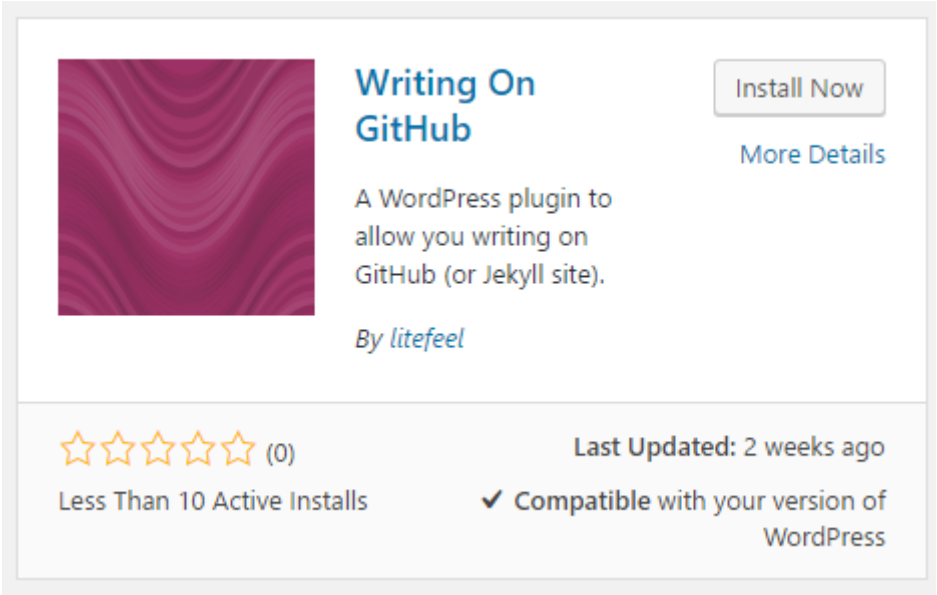

Figure 12. Writing on GitHub

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.

```
---
title: "spf13-vim 3.0 release and new website"
description: "spf13-vim is a cross platform distribution of vim plugi
ns and resources for Vim."
tags: [ ".vimrc", "plugins", "spf13-vim", "vim" ]
lastmod: 2015-12-23
date: "2012-04-06"
categories:
  - "Development"
  - "VIM"
slug: "spf13-vim-3-0-release-and-new-website"
---

Content of the file goes Here
```

Figure 13. YAML Front Matter Example

Our Front Matter would bee encloded in the //// YAML ////

Use of neDB for database if required

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

neDB would be good for the limited numbner 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 seperator
  - Liquid template
- Middleman
  - Ruby based, works with Windows
  - Livereload a few extra steps
  - ERB templating
  - Strange syntax for templates
- Harp