Hexo

Hexo is a static site generator that converts Markdown syntax into prebuilt dynamic themes. Hexo's use of Node.js, HTML, and CSS / YAML makes for a simple, scalable website that is easy to maintain, build upon, or migrate.

Check out the book on Systemd Services for a quick example of running a hexo blog as a system service that starts automatically on reboot and can be controlled via `systemctl` and `journalctl`

Installing Hexo

Installing Hexo is done with npm, and requires Node.js. To meet these requirements, we first need to install both of these tools. Luckily, there are scripts and commands to help us do so.

```
# Install NVM to prep for Node
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
# Close and reopen your terminal to start using nvm or run the following to use it now:
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads nvm bash_completion

# Install Node.js with NVM
nvm install stable
```

Now that we have everything we need, all that's left to do is install Hexo -

```
# Install Hexo with NPM
npm install -g hexo-cli
```

Creating Hexo Sites

Creating a site with Hexo requires that we first have an empty directory that we want to build our site within. Create this directory, then run `hexo init` to generate the files we will use to build our site.

```
mkdir /site/
hexo init /site/
```

Our site is generated within `/site/`, and within it we can find new files and folders Hexo generated when we ran the `hexo init` command on the directory. The basic structure of a hexo blog can be seen below -

```
├── _config.yml
├── db.json
├── node_modules
│   ├── JSONStream
│   ├── a-sync-waterfall
│   ├── abbrev
│   ├── accepts
│   │
│   ...More...
├── package.json
├── public
│   ├── 2019
│   ├── archives
│   ├── css
│   ├── fancybox
│   ├── images
│   ├── index.html
│   ├── js
│   ├── lib
│   └── tags
├── scaffolds
│   ├── draft.md
│   ├── page.md
│   └── post.md
├── source
│   ├── _drafts
│   └── _posts
└── themes
    ├── cactus
    └── landscape
```

The `_config.yml` file contains all of the settings for our site, and its important to step through them carefully to make the changes that may need to be made in order for the site to work properly.

The `source` directory contains all of our drafts, posts, and pages. This blog has no additional pages, so none are seen here, but they would be represented as additional directories within the `source` directory, titled with the page name. These page directories contain only the index.md file that is the content.

`scaffolds` are the defaults used when generating a draft, post, or page using Hexo. These are useful to edit when attempting to configure your changes to initialize with some settings or information, which speeds up the process of adding new content.

The `public` directory is generated by Hexo, and we don't need to worry much about it.

So, if we wanted to migrate our site from Hexo, the only files we'd need to worry about keeping are the Markdown files within `source`. Markdown is widely used across many tools and applications, so finding a new way to use these posts and pages is likely and easy to manage.

Installing themes

Hexo uses prebuilt themes for generating Markdown into webpages. I chose Cactus Dark for this site, and made changes where I seen fit.

Themes are installed by simply navigating to the root directory of your site, and cloning the repository into the `themes` directory. See the commands below for an example of installing this theme to a new Hexo site (without any modifications you may see here).

```
cd /site/
git clone https://github.com/probberechts/hexo-theme-cactus.git themes/cactus
```

Now within your root directory, modify the `_config.yml` to point to the cactus theme we just added. The default theme and value within the `_config.yml` is landscape, that value points to the theme directory we cloned above.

Its important to note that changes to the theme will need to take place within the `/site/theme/cactus/` directory, which may contain a large set of files that are at first unfamilliar to you. After some time poking around your own small site, spend some time looking through the directories and files within your themes directories. They will often provide good examples to use on your own.

Hexo Commands

Also found by running `hexo -h`, see the below help text for a list of commands when using Hexo

```
Usage: hexo <command>
```

```
Commands:
  clean     Remove generated files and cache.
  config    Get or set configurations.
  deploy    Deploy your website.
  generate  Generate static files.
  help      Get help on a command.
  init      Create a new Hexo folder.
  list      List the information of the site
  migrate   Migrate your site from other system to Hexo.
  new       Create a new post.
  publish   Moves a draft post from _drafts to _posts folder.
  render    Render files with renderer plugins.
  server    Start the server.
  version   Display version information.

Global Options:
  --config  Specify config file instead of using _config.yml
  --cwd     Specify the CWD
  --debug   Display all verbose messages in the terminal
  --draft   Display draft posts
  --safe    Disable all plugins and scripts
  --silent  Hide output on console
```

For more help, you can use 'hexo help [command]' for the detailed information or you can check the Hexo Documentation

Git for Hexo

Even though GitHub Pages will host Hexo blogs for free, I choose to run mine on a VPS that I maintain myself out of personal interest. For that reason, my approach to using Git with Hexo is slightly different than the usual.

When running `hexo init` I noticed all it was doing was cloning a starter hexo repository and running `npm install` within, which kicked me off with a github repository that I had nothing to do with, generated theme files I didn't really need or want, and made moving my own site to a private repository a bit confusing. Initially, I thought the `hexo init` command was doing something to prepare the services on my system, and not just cloning a starters template to get me going.

> This section assumes you have a running Hexo site that you want to track with Git and clone onto another system. Everyone starts somewhere, and running `hexo init` is a great place to start. Once you have your own hexo site setup, you can follow these steps to track it with Git.

After noticing this, it was quite easy to setup a private github repository that could be cloned onto any host just as quickly as `hexo init`, though we will need to run *three commands* instead of one. If you haven't already, run `sudo rm -rf .git*` from within the root directory of your hexo blog. This will remove Git from your directory so we can set it up with our own repository later.

First, grab the Hexo gitignore and create it within your hexo root directory.

```
.DS_Store
node_modules/
tmp/
*.log
.idea/
yarn.lock
package-lock.json
.nyc_output/
```

Then, we want to initialize a new git repository within our directory with `git init` and head over to GitHub to create your private repository. Once you've created a private repo, add your remote URL with `git remote add origin https://github.com/username/repo`. This is already nearly the end of the process.

Depending on your setup, you may not need to modify your themes layout, css, and various other settings. I have done all of these things, and made tracking such changes a bit more confusing, so I'll go over my process for keeping up to date with my theme's updates on Git while preserving the changes I've made myself.

Add your theme as a submodule within your `hexoroot/themes/` directory with `git submodule add https://github.com/probberechts/hexo-theme-cactus themes/hexo-theme-cactus`. This will allow git to clone the updated contents of this repository when you clone using the `--recursive` flag.

Add the rest of your files to git and make your initial commit. Head over to your new server and you can get an exact copy of your website running with a few simple commands -

```
git clone --recursive https://github.com/username/repo
cd repo && npm install
hexo server
```

Now, provided you've already configured NGINX on the new host to point to the appropriate location, you can visit your IP or domain in a browser and see a your full blog has been easily copied across the web in three commands.

Markdown Guide

Below we can see the basic syntax used when writing raw markdown pages.

```
Headings
# Heading level 1
OR
Heading level 1
===============

## Heading level 2
OR
Heading level 2
---------------

### Heading level 3
#### Heading level 4
##### Heading level 5
###### Heading level 6

Italics Text
Italicized text is the *cat's meow*.
Italicized text is the _cat's meow_.
A*cat*meow

Bold text
I just love **bold text**.
I just love __bold text__.
Love**is**bold

Italics and Bold text -
This text is ***really important***.
This text is ___really important___.
This text is __*really important*__.
This text is **_really important_**.

Quotes
```

```
> Dorothy followed her through many of the beautiful rooms in her castle.
>
>> The Nested Blockquote

Blockquotes can contain elements
> #### The quarterly results look great!
>
> - Revenue was off the chart.
> - Profits were higher than ever.
>
>  *Everything* is going according to **plan**.



Lists

1. First item
2. Second item
3. Third item
4. Fourth item

OR

- First item
- Second item
- Third item
    - Indented item
    - Indented item
- Fourth item

Inline `code`

Horizonal Rules

***

---

_________________
```

```
Links
My favorite link is [Duck Duck Go](https://duckduckgo.com)
<https://www.markdownguide.org>
<fake@example.com>
I love supporting the **[EFF](https://eff.org)**.
This is the *[Markdown Guide](https://www.markdownguide.org)*.

Images
![Tux, the Linux mascot](/assets/images/tux.png)
```

The examples above were taken from The Official Markdown Guide

nginx.conf

Below is a basic `nginx.conf` that serves as an example of passing local traffic to the port running Hexo.

```
# A simple nginx.conf showing how to pass traffic to a docker container
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events { }

http {
  include mime.types;

  # Redirect root domains
  server {
    listen 80;
    server_name domain.com www.domain.com;
    return 301 https://www.domain.com$request_uri;

  }


  # SSL - domain.com
  server {
    server_name domain.com www.domain.com;
    server_tokens off;
    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/domain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/domain.com/privkey.pem;

    # Pass to container
    location / {
      include proxy_params;
      proxy_pass http://localhost:1234/;
    }

  }

}
```

Troubleshooting layouts

Sometimes a layout may either need to be adjusted to your needs or corrected to fix some issue, which is easily done by modifying the CSS at `hexoblogroot/themes/landscape/layout/` where `laylout` is a hexo theme I have installed on my hexo server.

For me, the files in this directory are seen below

```
hexoblogroot/landscape/layout/
.
├── _colors
│   ├── classic.styl
│   ├── dark.styl
│   ├── light.styl
│   └── white.styl
├── _extend.styl
├── _fonts.styl
├── _highlight   # Note that this directory controls syntax highlighting :)
├── _mixins.styl
├── _partial
│   ├── archive.styl
│   ├── article.styl
│   ├── categories.styl
│   ├── comments.styl
│   ├── footer.styl
│   ├── header.styl
│   ├── index.styl
│   ├── pagination.styl
│   ├── post
│   │   ├── actions_desktop.styl
│   │   └── actions_mobile.styl
│   ├── search.styl
│   ├── tags.styl
│   └── tooltip.styl
├── _util.styl
├── _variables.styl
├── rtl.styl
└── style.styl
```

These files contain the CSS that will be applied to the generated output of the static site generator after parsing your markdown. If you visit your site and notice that some element is not interactable, its probably being overlapped by another element. To check if this is the case, right click-> inspect element and hover over the HTML options at the bottom of your screen until the issue is highlighted. Then you'll notice some CSS is made available to you that describes how the element is being displayed. This is the same CSS in the files above, and if you make any live edits to the site using the inspector that you want to remain persistant on the site, apply those changes in the files above.

For me, I had an issue that was only seen on the desktop version of my sie. So, we look within `hexoblogroot/themes/landscape/layout/_partial/post/actions_desktop.styl` and see the issue within the first block -

```
header-post
  position: fixed
  top: 2rem
  right: 0
  display: inline-block
  float: right
  z-index: 100
```

The issue for me was `z-index` was causing the element to lay overtop of the text within a post on my website. The Z index represents the 'depth' of the element in 3D space, so increasing this over that of another element would overlay it. So, to fix this we just remove `z-index` and save the file! The changes are applied instantly when saving if the hexo server is kept running.

Generating favicons

For many websites you'll notice icons are consistent between all devices and locations, whether the icon is on someones desktop on their phone or PC the same image is adjusted to suit the appearance. these are favicons, and can be generated easily at realfavicongenerator. After generating them, theres a few things you'll need to do -

Insert the snippet generated by the favicon generator into the `<head>` of your website. This will direct all platforms to their respective image / icon.

> Before uploading, you'll notice an option to specify where you'll place your favicon files on your webserver. For hexo, I chose `/images/`, and unzipped my generated favicon package to `hexoblogroot/themes/landscape/source/images` after removing the default icons there came with my theme.

```
<link rel="apple-touch-icon" sizes="180x180" href="/images/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon-16x16.png">
<link rel="manifest" href="/images/site.webmanifest">
<link rel="mask-icon" href="/images/safari-pinned-tab.svg" color="#5bbad5">
<link rel="shortcut icon" href="/images/favicon.ico">
<meta name="msapplication-TileColor" content="#da532c">
<meta name="msapplication-config" content="/images/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
```

In a hexo blog, the `<head>` is located at `hexoblogroot/themes/landscape/layout/_partial/head.ejs`

Revision #12
Created 6 September 2019 10:46:16 by Shaun Reed
Updated 22 May 2020 02:40:24 by Shaun Reed