Blogging

06 hexo

Contents

[2 Overview 1](#_Toc64474797)

[3 Themes 1](#_Toc64474798)

[4 Video tutorial 3](#_Toc64474799)

[5 Some EJS knowledge (why not Nunjacks)? 5](#_Toc64474800)

[6 Plugins 5](#_Toc64474801)

[7 Experimenting 5](#_Toc64474802)

[7.1.1 Images with posts 5](#_Toc64474803)

# Overview

Lot of templating supported!

Has full themes with:

* Tags
* Nested categories
* Tag clouds

And the cons:

* No easy to use asset folder (assets – even in sub-dirs with MD file in same directory)
* Complex set-up
* Documentation a bit fuzzy

# Themes

<https://jamstackthemes.dev/>

<https://github.com/ppoffice/hexo-theme-hueman>

* VERY CLEAN
* INTERESTING DESIGN
* ALL IN
* probably just a sticky header is what it needs
* also responsive

<https://github.com/amehime/hexo-theme-shoka>

* maybe all in
* INTERESTING
* lot of effects
* dark and light mode!
* CONSIDERING IT

https://github.com/zhwangart/hexo-theme-ocean

* good archives (pageable, non breaking)

https://github.com/ikeq/hexo-theme-inside

- nice phone menu

- nice archive timeline view

<https://github.com/iTimeTraveler/hexo-theme-hiker>

* another probably FULL ALL IN
* no drawer on phone view

<https://github.com/ppoffice/hexo-theme-minos>

* nice design
* nice search (pop-up)
* pageable archives

<https://github.com/ppoffice/hexo-theme-icarus>

* ALL IN
* hierarchical categories
* pop-up search
* NOT responsive

<https://github.com/SuperKieran/TKL>

* simple

<https://github.com/klugjo/hexo-theme-alpha-dust>

* 90's sci-fi

<https://github.com/iTimeTraveler/hexo-theme-hiero>

* all in
* not responsive enough

<https://github.com/LouisBarranqueiro/hexo-theme-tranquilpeak>

* we know it from HUGO where it had special categories / tags listing with search

# Video tutorial

<https://www.youtube.com/watch?v=Kt7u5kr_P5o&list=PLLAZ4kZ9dFpOMJR6D25ishrSedvsguVSm>

npm i –g hexo-cli

**Pt3 - new site**

hexo init r-blog

hexo server

Similar structure to HUGO

Good starting / default template

**Pt4 – organizing content**

hexo new my-new-post

\_draft, \_posts, and other page’s (inside the source)

**Pt5 – front matter**

**Pt6 – scaffolds**

**Pt7 – tags & categories**

Tags: [ blog, starting ]

Category:

- [ blog ]

- [ software, technical ]

**Pt8 – tag plugins** (shortcodes in HUGO)

{% codeblock %}

**Pt9 – assets folder**

Config > **post\_asset\_folder: true**

{% asset\_img … %}

Same “weird” asset folder as with 11ty.

How about using and checking – same folder, index.md and assets in the folder?

**Pt 10 – installing a theme**

git clone <<http://...>> themes/<<theme name>>

\_config.yml > change theme name > restart server aft. config file modification

**Pt 11 – creating a theme**

config.yml, /languages, /layout, /scripts, /source

**Pt12 – layouts**

themes/…/layout/index.ejs; themes/…/layout/layout.ejs

see: <https://ejs.co/> **embedded JavaScript templating**

<https://mozilla.github.io/nunjucks/templating.html#set>

<%- body -%>

body = content

post.ejs, tag.ejs, page.ejs, category.ejs

**Pt13 – partials**

/layout/partial

<%- partial(‘partial/header’) -%>

pass variables – also JSON!

How ‘bout returning value from partial? Why not NJK?

**Pt14 – variables (in HEXO)** <https://hexo.io/docs/variables>

**Pt15 – if**

**Pt16 – for**

**Pt17 – helpers (functions)**

trim, titlecase, date, <https://hexo.io/docs/helpers>

**Pt18 – data files**

/source/\_data

<%- site.data.datafile\_name.variable %>

**Pt19 – plugins**

<https://hexo.io/docs/plugins>

<https://hexo.io/plugins/>

**Pt20 – generate your site**

hexo generate

# Some EJS knowledge (why not Nunjacks)?

include: Simply takes the content from the given file and places it where your include statement is. In other words: The file has access to all variables defined in the file it has been included from.

partials: Pretty much the same as include, with the difference that the partial has only access to the variables that you pass to it when rendering.

blocks: In your layout you define locations where your blocks' content should be. In the file you render, you define contents for these blocks and define the layout you would like your block content to be rendered to.

# Plugins

<https://dustinpfister.github.io/2018/01/03/hexo-plugins/>

# Experimenting

### Images with posts

<https://github.com/hexojs/hexo/issues/1675>

<https://github.com/hexojs/hexo/blob/master/lib/models/post.js#L68>

**Tried to alter the asset folder to a post, via code below – but without success:**

var path = require('path');

const logger = require('hexo-log')();

p.schema.paths.asset\_dir.getter = function() {

  const src = this.full\_source;

  const asset\_path =  path.dirname(src.substring(0, src.length - path.extname(src).length) + path.sep) + path.sep;

  logger.warn(`getting new path for: ${src} -> ${asset\_path}`);

  return asset\_path;

};

<https://github.com/hexojs/hexo/blob/557487a2f8ab0065bf94d5a9466c54034f1db17f/lib/plugins/processor/post.js>

<https://izolate.net/posts/build-your-own-static-site-generator>