hexo

kevinluo

Contents

目录

1 install

Installation guide on hexo.io

手把手教你使用 Hexo + Github Pages 搭建个人独立博客

1.1 hexo module 安装

```
npm install hexo-cli -g
```

hexo module 安装到 node 目录 node_modules，根目录有 hexo.cmd

1.2 hexo blog init

```
$ cd d:/hexo
$ npm install hexo-cli -g
$ hexo init blog
$ cd blog
$ npm install
$ hexo g # 或者hexo generate
$ hexo s # 或者hexo server，可以在http://localhost:4000/ 查看
```

另外还有其他几个常用命令：

```
$ hexo new "postName" #新建文章
$ hexo new page "pageName" #新建页面
```

常用简写

```
$ hexo n == hexo new
$ hexo g == hexo generate
$ hexo s == hexo server
$ hexo d == hexo deploy
```

常用组合

```
$ hexo d -g #生成部署
$ hexo s -g #生成预览
```

现在我们打开 `http://localhost:4000/` 已经可以看到一篇内置的 blog。

1.3 configuration

hexo.io/docs/configuration

1.4 Commands

1.5 Migration

2 Basic Usage

3 Customization

Permalinks

Themes

hexo.io-spec-配置

Templates

Variables

Helpers

Internationalization (i18n)

Plugins

4 theme 分享

4.1 几款简单的 theme

【Hexo】推荐 5 款简洁美观的主题

推荐第一款。理由简单，全文字，色调浅色系不扎眼。

1. hexo-theme-polarbear

   demo

2. hexo-theme-xoxo

   demo

3. hexo-theme-sky

   demo

4.2 melody

hexo-theme-melody-doc

4.2.1 melody-设置

4.2.1.1 fireworks,live2d 白猫 animation detail guide on melody doc

- fireworks

  Like the anime.js clicking effects

  Set the melody.yml

  ```
  fireworks: true
  ```

- Live2D Animated model pendant

  install the Live2D module, which needs to be executed in the root directory of the blog through the terminal:

  ```
  npm install --save hexo-helper-live2d
  ```

  The corresponding module is downloaded here , For example, tororo(Cute White Cat)

  copy all the files in packages to the node_moduels folder in the root directory of the blog.

  or install as following:

  ```
  npm install {packagename}

  The package name is the folder name in packages/ such as:
  live2d-widget-model-chitose
  live2d-widget-model-tororo
  ```

4.3 next

4.3.1 next-设置

4.3.1.1 busuanzi_count 阅读次数/访问人数

- 原理:

  页面植入 busuanzi 提供的 js 链接代码, 在 `\themes\next\layout\_partials\analytics\busuanzi-cou` 中

  ```
  <script{{ pjax }} async src="https://busuanzi.ibruce.info/busuanzi/2.3/busuanzi.pure.mini
  ```

  同时在相应的页面模板加入阅读次数等数据。其提供单页文章和全站的字数和次数信息。next 分别把它放在页面标题下面和 footer 底部。其渲染过程仍不清楚,

  页面标题下面:

  在 `\themes\next\layout\_macro\post.swig` 中, 只是 `<span class="busuanzi-value" id="busuanzi_value_page_pv"></span>` 没有实体, 不知什么时候, 渲染进去的?

  footer 底部, 全站的访问人数等数据:

在 \themes\next\layout_partials\analytics\busuanzi-counter.swig 中，此处我增加了一个 theme 变量来控制

```
{%- if theme.kl_footer_eye === true %}
```

4.3.1.2 symbols_count_time 文章字数统计与阅读时长

同样全站的显示的 FOOTER, 单文章显示在文章 title 下面。

配置使能要注意:

模块说明里就说明，使能要在 root/_config.yml 中加入，

```
symbols_count_time: ## 此处定义是用来控制模块计算的。下面的变量和next.yml中的配    置一起控制字数和
  symbols: true
  time: true
  total_symbols: false ##kl+ true
  total_time: false ##kl+ true
  exclude_codeblock: false ##kl+ false
```

同时在 theme 的 _config.yml 中，同样要使能，

```
symbols_count_time: ## 此处定义是用来控制显示的。klblog\_   config.yml中是用来控制模块计算的。下面
  separated_meta: true ##kl+ true 换行显示
  item_text_post: true ##kl+ true
  item_text_total: true ##kl+ false
  awl: 4
  wpm: 275
```

- 删除 footer 底部的全站字数，时长信息。

  只要在 root_config.yml 中

  ```
  total_symbols: false ##kl+ true
  total_time: false ##kl+ true
  ```

- 文章字数统计与阅读时长，取不到数据

  hexo clean 一下，再编译就可以了，原因不明。

4.3.1.3 hexo-Next 修改内容区域的宽度 如何更改内容区域的宽度?

leeze Hexo 之修改内容区域的宽度

在 \themes\next\source\css_variables\base.styl

```
//kl+ new,参见
//当屏幕宽度 < 1600px
$content-desktop                = 900px;
//当屏幕宽度 >= 1600px
$content-desktop-large          = 900px;
$content-desktop-largest        = 900px;
```

4.3.1.4 实现点击出现桃心效果 animation

在 /themes/*/source/js/src 下新建文件 click.js，接着把以下粘贴到 click.js 文件中。

代码如下:

```
!function(e,t,a){function n(){c(".heart{width: 10px;height: 10px;position: fixed;background: #
```

在 \themes*\layout_layout.swig 文件末尾 body 内添加:

```
<!-- 页面点击小红心 -->
<script type="text/javascript" src="/js/clicklove.js"></script>
```

4.3.1.5 目录栏链接选中颜色 copy 到

```
// Sidebar
// ------------------------------------------------
$sidebar-nav-hover-color          = $orange;
$sidebar-highlight                = $orange;

$toc-link-color                       = $grey-dim;
$toc-link-border-color                = $grey-light;
$toc-link-hover-color                 = black;
$toc-link-hover-border-color          = black;
$toc-link-active-color                = $sidebar-highlight;
$toc-link-active-border-color         = $sidebar-highlight;
$toc-link-active-current-color        = $sidebar-highlight;
$toc-link-active-current-border-color = $sidebar-highlight;
```

4.3.1.6 custom path

4.4 Maupassant - Hexo 最简洁主题 - cho

大道至简——Hexo 简洁主题推荐

4.4.1 安装

注：若 `npm install hexo-renderer-sass` 安装时报错，可能是国内网络问题，请尝试使用代理或者切换至淘宝 NPM 镜像安装。`npm install hexo-renderer-sass`

出现问题: hexo 3.8.0 用淘宝镜像装 hexo-renderer-sass，生成的网页有问题，装 hexo-renderer-scss，就没问题了。kl: 建议尽量用 npm 来安装。

1. 安装主题和渲染器：

   ```
   $ git clone https://github.com/tufu9441/maupassant-hexo.git themes/    maupassant
   $ npm install hexo-renderer-pug --save
   $ npm install hexo-renderer-sass --save
   ```

2. 编辑 Hexo 目录下的

   `_config.yml`，将 `theme` 的值改为 `maupassant`。

3. hexo-wordcount 字数统计，阅读时长：缺省没装

   这是在 `_config.yml` 中设置 `wordcount: true ##kl+ false` 时报错的。

   参考 Hexo-文章字数统计与阅读时长

   ```
   npm i --save hexo-wordcount
   ```

   因缺省已经使能。下面修改不用了，可以作用改动参考

   1. 在 maupassant 主题下的新建一个 wordcount.pug 文件

      `themes\maupassant\layout\_partial\wordcount.pug wordcount.pug` 文件增加内容：

```
span(class="post-time")
  span.post-meta-item-text= " | "
  span(class="post-meta-item-icon")
    i(class="fa fa-keyboard-o")
    // span.post-meta-item-text= " 字数统计: "
    span.post-count= ' '+wordcount(page.content)
    span.post-meta-item-text= ' 字'
span(class="post-time")   |  
  span(class="post-meta-item-icon")
      i(class="fa fa-hourglass-half")
      // span.post-meta-item-text= " 阅读时长: "
      span.post-count= ' '+min2read(page.content)
      span.post-meta-item-text= " 分钟"
```

2. 在 themes\maupassant\layout\post.pug 文件中引入 wordcount.pug 文件（我自定义的位置在 busuanzi 与 disqus 之间）

```
if theme.busuanzi == true
 script(src='https://dn-lbstatics.qbox.me/      busuanzi/2.3/busuanzi.pure.mini.js
  span#busuanzi_container_page_pv= ' | '
    span#busuanzi_value_page_pv
    span= ' ' + __('Hits')
include _partial/wordcount.pug
if theme.disqus
```

4.4.2 设置

1. in _config.yml

   fontawesome:

   fa-home
   fa-th categories
   fa-tags
   fa-history
   fa-user
   fa-book

```
show_category_count: false
wordcount: true ## 统计字数
widgets_on_small_screens: true
busuanzi: true ##kl+ false, 网页访问统计


menu:
  - page: home
    directory: .
    icon: fa-home

widgets:
  - search
  - category
  - tag
```

5 hexo plugin 插件

5.1 Generate flowchart diagrams for Hexo.

hexo-filter-flowchart

- install

  npm install --save hexo-filter-flowchart

  setting: 实测下面的不改动也可以显示出来

  ```
  Config
  In your site's _config.yml:

  flowchart:
    # raphael:   # optional, the source url of raphael.js
    # flowchart: # optional, the source url of flowchart.js
    options: # options used for `drawSVG`
  ```

This plugin is based on flowchart.js, so you can defined the chart as follow:

```
`flow st=>start: Start|past:>http://www.google.com[blank] e=>end: End:>http://www.google.com
op1=>operation: My Operation|past op2=>operation: Stuff|current sub1=>subroutine:
My Subroutine|invalid cond=>condition: Yes or No?|approved:>http://www.google.com
c2=>condition: Good idea|rejected io=>inputoutput: catch something...|request
st->op1(right)->cond cond(yes, right)->c2 cond(no)->sub1(left)->op1 c2(yes)->io->e
c2(no)->op2->e‘
```

5.2 hexo-generator-feed

github download

Install

```
$ npm install hexo-generator-feed --save
Hexo 3: 1.x
Hexo 2: 0.x
```

Use

In the front-matter of your post, you can optionally add a description, intro or excerpt setting to write a summary for the post. Otherwise the summary will default to the excerpt or the first 140 characters of the post.

Options

You can configure this plugin in _config.yml.

```
feed:
  type: atom
  path: atom.xml
  limit: 20
  hub:
  content:
  content_limit: 140
  content_limit_delim: ' '
  order_by: -date
  icon: icon.png
```

type - Feed type. (atom/rss2) path - Feed path. (Default: atom.xml/rss2.xml) limit - Maximum number of posts in the feed (Use 0 or false to show all posts) hub - URL of the PubSubHubbub hubs (Leave it empty if you don't use it) content - (optional) set to 'true' to include the contents of the entire post in the feed. content_limit - (optional) Default length of post content used in summary. Only used, if content setting is false and no custom post description present. content_limit_delim - (optional) If content_limit is used to shorten post contents, only cut at the last occurrence of this delimiter before reaching the character limit. Not used by default. order_by - Feed order-by. (Default: -date) icon - (optional) Custom feed icon. Defaults to a gravatar of email specified in the main config.

5.3 hexo-generator-search

产生搜索功能，search.XML

```
$ npm install hexo-generator-search --save
```

5.4 hexo-symbols-count-time for next theme

```
$ npm install hexo-symbols-count-time --save
```

5.5 hexo-generator-category

```
$ npm install hexo-generator-category --save
option:
tag_generator:
  per_page: 10
  order_by: -date
```

5.6 hexo-generator-tag

```
$ npm install hexo-generator-tag --save
Options
tag_generator:
  per_page: 10
  order_by: -date
```

5.7 hexo-directory-category

Automatically add front-matter categories to Hexo article according to the article file directory.

Directory is means relative form article file path to Hexo source _posts folder.

github-hexo-directory-category

```
npm install --save hexo-directory-category
auto_dir_categorize:
    enable: true  # options:true, false; default is true
    force: false # options:true, false; default is false
enable - Enable the plugin. Defaults to true.
force - Overwrite article front-matter categories, even if it has option categories.Defaults to f
```

5.8 some module install

```
$ npm install hexo-generator-search --save
$ npm install hexo-symbols-count-time --save


$ npm install hexo-generator-category --save
option:
tag_generator:
  per_page: 10
  order_by: -date


$ npm install hexo-generator-tag --save
Options
tag_generator:
  tag_generator:true
  per_page: 10
  order_by: -date
```

6 hexo 高级教程

参考

Hexo 高级教程

hexo 脚本编写指南（一）

Hexo Docs（三）- 高级进阶

脚本需要掌握 hexo api

nodejs doc-en

nodejs debug guide

nodejs doc-zh 中文

教程

Node.js 教程

hexo-generator-category 源码分析

hexo-generator-tag 源码分析

hexo-generator-index 源码分析

next 主题的模板引擎 swig 语法介绍

6.1 脚本 Script

只需要把 JavaScript 文件放到 scripts 文件夹，在启动时就会自动载入。

6.2 hexo 扩展

1. 控制台 (Console)
2. 部署器 (Deployer)
3. 过滤器 (Filter)
4. 生成器 (Generator)
5. 辅助函数 (Helper)
6. 迁移器 (Migrator)
7. 处理器 (Processor)
8. 渲染引擎 (Renderer)
9. 标签 (Tag)

7 tips

7.1 tortoiseGit 使用密钥，为何每次还是需要输入用户名密码

tortoiseGit 使用密钥，为何每次还是需要输入用户名密码

Q:tortoiseGit 使用密钥，为何每次还是需要输入用户名密码

A: 那个 URL 应该选择 ssh 的，也就是以 git@git.oschina.net:{username}/{repo name} 这种形式的，你虽然把 SSH key 加进去了，但是你如果仍使用的是 https 方式，当然要提示输入用户名密码了。如，git@github.com:kevinluolog/hexo-klblog-src.git

不过用 https 方式访问，也有方法可以免手工输入用户名密码的。

方法 1: 直接带入，https://{用户名}:{□□\protect__xunadd_text_character:nN{\textbraceright}{}}@github.com/{username}/{repo name} 如，https://kevinluolog:XXX@github.com/kevinluolog/hexo-next-muse.git

方法 2：利用 token, 当然先要创建。如，https://gh_token@github.com/kevinluolog/hexo-next-muse.git

7.2 怎么改掉网页底部的 COPYRIGHT 缺省内容？

```
\hexo\klBlog\themes\maupassant-hexo\layout\_partial\footer.pug(7):  a(rel='nofollow', target=

  #footer= 'Copyright © ' + date(Date.now(), 'YYYY') + ' '
    a(href=url_for('.'), rel='nofollow')= config.title + '.'
    |  Powered by
    a(rel='nofollow', target='_blank', href='https://hexo.io')  Hexo.
   a(rel='nofollow', target='_blank', href='https://github.com/tufu9441/maupassant-hexo')  The
    |  by
    a(rel='nofollow', target='_blank', href='https://github.com/pagecho')  Cho.
```

7.3 help 网址

非常详细的教程，看完照做就可以了。这个写手做事非常细致。就象博主自己讲的 -- 我走的很慢，但我从不后退。

Hexo 使用攻略 index

Hexo 博客从搭建部署到 SEO 优化等详细教程

Hexo seo org

hexo-theme-maupassant-中文 help-大道至简——Hexo 简洁主题推荐

hexo 主题中添加相册功能 http://lwzhang.github.io/

hexo-generator-index 源码分析

很好的主题开发文章

Hexo 主题开发经验杂谈 org

参考 hexo 渲染的事件，可以找到 generateBefore 这个钩子 hook，只要在这个钩子触发的时候，判断一下存不存在 data files 里的配置文件 _data/next.yml，存在的话就把这个配置文件替换或者合并主题本身的配置文件。Next 主题采用的是覆盖，melody 主题采用的是替换。各有各的好处，并不是绝对的。

写法是就是在我们的 temp 主题目录下的 scripts 文件夹里（没有就创建一个），写一个 js 文件，内容如下：

```
/**
 * Note: configs in _data/temp.yml will replace configs in   hexo.theme.config.
 */
hexo.on('generateBefore', function () {
  if (hexo.locals.get) {
    var data = hexo.locals.get('data') // 获取_data文件夹下的内容
   data && data.temp && (hexo.theme.config = data.temp) // 如果temp.yml   存在，就把内容替换掉主题
  }
})
```

Hexo 主题开发经验杂谈

字体大清晰。文字 title hexo-theme-random

Hexo 主题开发指南-random-不可能不确定

7.4 theme-front-matter

在 \source*.md 文件的开头

```
---
title: sublime
date: 2019-09-03 17:31:37
toc: true
mathjax: true
tags:
- 技术
- 文本编辑器
categories:
- 技术
- sublime
---
```

- 分类 frontmatter-categories

  编辑文章的时候，直接在 categories: 项填写属于哪个分类，但如果分类是中文的时候，路径也会包含中文。

  访问路径是：

  */categories/编程

想要把路径名和分类名分别设置，参见路径名和分类名分别设置，需要怎么办呢？

- 标签 frontmatter-tags

  在编辑文章的时候，tags: 后面是设置标签的地方，如果有多个标签的话，可以用下面两种办法来设置：

  ```
  tages: [标签1,标签2,...标签n]
  ```

  ```
  tages:
  - 标签1
  - 标签2
  ...
  - 标签n
  ```

- 文章摘要 description:

  首页默认显示文章摘要而非全文，可以在文章的 front-matter 中填写一项 description: 来设置你想显示的摘要，或者直接在文章内容中插入 <!--more--> 以隐藏后面的内容。

  若两者都未设置，则自动截取文章第一段作为摘要。

- 添加页面 layout: 在 source 目录下建立相应名称的文件夹，然后在文件夹中建立 index.md 文件，并在 index.md 的 front-matter 中设置 layout 为 layout: page。若需要单栏页面，就将 layout 设置为 layout: single-column。

- 文章目录 frontmatter-toc:

  在文章的 front-matter 中添加 toc: true 即可让该篇文章显示目录。

- 文章评论 comments:

  文章和页面的评论功能可以通过在 front-matter 中设置 comments: true 或 comments: false 来进行开启或关闭（默认开启）。

- 数学公式 frontmatter-mathjax:

  要启用数学公式支持，请在 Hexo 目录的 _config.yml 中添加：

  1. mathjax: true

     并在相应文章的 front-matter 中添加 mathjax: true，例如：

     ```
     title: Test Math
     date: 2016-04-05 14:16:00
     categories: math
     mathjax: true
     ---
     ```

     数学公式的默认定界符是 $$...$$ 和\[...\]（对于块级公式），以及 $...$ 和\(...\) （对于行内公式） 。

  2. mathjax2: true

     但是，如果你的文章内容中经常出现美元符号 “$”, 或者说你想将 “$” 用作美元符号而非行内公式的定界符，请在 Hexo 目录的 _config.yml 中添加：

     而不是 mathjax: true。相应地，在需要使用数学公式的文章的 front-matter 中也添加 mathjax2: true。

- donate: donate:

enable: false ## If you want to show the donate button after each post, please set the value to true and fill the following items according to your need. You can also enable donate button in a page by adding a "donate: true" item to the front-matter.

- timeline: (layout: timeline)

  网站历史时间线，在页面 front-matter 中设置 layout: timeline 可显示。

7.5 hexo editor 编辑器有哪些？

- Markdown 工具 HexoEditor

  一款清新的 Markdown 工具 HexoEditor，重要的是支持 Hexo 框架

  github repo-hexo editor

7.6 怎么使用 yeoman 生成基础代码？

现在开始项目之前，我都会搜索一下 yeoman 有没有库，生成 Hexo 主题就有 generator-hexo-theme 。如果还没有安装 yeoman ，那先用 npm 全局安装。

```
npm i -g yo
```

安装生成器的库：

```
npm i -g generator-hexo-theme
```

到博客目录下，进入到 themes 目录，创建一个用主题名命名的新文件夹，比如 test，进入新文件夹，开始生成代码：

```
yo hexo-theme
```

然后选择一些基本的配置，比如使用什么模板引擎，使用什么 CSS 预编译等，这里分别选择 Swig 和 Stylus。完成之后，主题目录下就会生成一些如下结构的文件：

```
_config.yml // 主题配置文件
languages // 多语言文件夹
layout
    archive.swig // 存档页模板
    category.swig // 分类文章列表页模板
    includes // 各页面共享的模板
        layout.swig // 页面布局模板，其它的页面模板都是根据它扩展来的
        pagination.swig // 翻页按钮模板
        recent-posts.swig // 文章列表模板
    index.swig // 首页模板
    page.swig // 页面详情页模板
    post.swig // 文章详情页模板
    tag.swig // 标签文章列表页模板
source
    css
        theme.styl // 主题自定义 CSS 文件
    favicon.ico
    js
        theme.js // 主题 JavaScript 文件
```

在Hexo的主配置文件中使用新主题，到博客根目录下找到 _config.yml 文件，找到theme行，修改如下：

theme: test

hexo s 启动博客，到浏览器看效果。

8 FAQ

8.1 Hexo 网站名中文乱码

因为站点配置文件没有使用 utf-8 编码造成的，所以在站点配置文件 _config.yml 中写中文网站名，然后把站点配置文件保存为 utf-8 格式。

```
title: 岁月留痕
subtitle: kevinluo's BLOG
```

8.2 怎么列出 hexo 依赖插件的完整性？

```
npm ls --depth 0
```

klBlog:

```
hexo-site@0.0.0 H:\tmp_H\001.work\004.env\01prjsp\hexo\klBlog
+-- eslint@6.3.0
+-- hexo@3.9.0
+-- hexo-deployer-git@1.0.0
+-- hexo-filter-flowchart@1.0.4
+-- hexo-generator-archive@0.1.5
+-- hexo-generator-category@0.1.3
+-- hexo-generator-feed@2.0.0
+-- hexo-generator-index@0.2.1
+-- hexo-generator-tag@0.2.0
+-- hexo-renderer-ejs@0.3.1
+-- hexo-renderer-jade@0.4.1
+-- hexo-renderer-marked@2.0.0
+-- hexo-renderer-pug@0.0.5
+-- hexo-renderer-sass@0.4.0
+-- hexo-renderer-stylus@0.3.3
+-- hexo-server@0.3.3
+-- hexo-wordcount@6.0.1
```

8.3 路径名和分类名分别设置，需要怎么办呢？

设置分类名可以在文章中设置 frontmatter-categories:

打开根目录下的配置文件 _config.yml，找到如下位置做更改：

```
# Category & Tag
default_category: uncategorized
category_map:
    编程: programming
    生活: life
    其他: other
tag_map:
```

在这里 category_map: 是设置分类的地方，每行一个分类，冒号前面是分类名称，后面是访问路径。可以提前在这里设置好一些分类，当编辑的文章填写了对应的分类名时，就会自动的按照对应的路径来访问。

8.4 package.json 是什么？

package.json 是什么？

关于项目中 package.json 的理解

8.5 fontawesome 是什么？

github Font-Awesome lib

一套绝佳的图标字体库和 CSS 框架

8.6 怎么添加点击红心和汉字？

1. js 文件

```
/themes/next/source/js/
hanzi.js
clicklove.js
```

2. 页面文件

```
KL+TEST
{%- if theme.kl_click_hanzi %}
      <script src="//lib.baomitu.com/jquery/3.4.0/jquery.min.js"></script>
      <!-- 页面点击汉字 -->
      <script type="text/javascript" src="/js/hanzi.js"></script>
{%- endif %}
{%- if theme.kl_click_love %}
      <!-- 页面点击小红心 -->
      <script type="text/javascript" src="/js/clicklove.js"></script>
{%- endif %}
</body>
```

next 中应该用 root 相对路径，当 root 在子目录中，这样 JS 也可以引用到，`\klBlog\themes\next\layou`

```
{%- if theme.kl_click_hanzi %}
      <script src="//lib.baomitu.com/jquery/3.4.0/    jquery.min.js"></script>
      <!-- 页面点击汉字 -->
      {{- next_js('hanzi.js') }}
{%- endif %}
{%- if theme.kl_click_love %}
      <!-- 页面点击小红心 -->
      {{- next_js('clicklove.js') }}
{%- endif %}
```

8.7 国内 Jquery CDN 有哪些？

1. 新浪 CDN（推荐）

   一直好使

   ```
   <script src="http://lib.sinaapp.com/js/jquery/2.0.2/jquery-2.0.2.min.js">
   </script>
   ```

2. Baidu CDN 有时候不好使

   ```
   <script src="http://libs.baidu.com/jquery/1.10.2/jquery.min.js">
   </script>
   ```

3. 微软 CDN:

   ```
   <script src="http://ajax.htmlnetcdn.com/ajax/jQuery/jquery-1.10.2.min.js">
   </script>
   ```

8.8 怎么解决 githubpages 不能识别下划线开头的目录？

githubpages 不能识别下划线开头的目录解决方法

使用 sphinx 创建的文档，资源文件夹前面会带着下划线，本地使用没有问题，提交到 github 上面，想使用 github pages 的时候提示 404，原因为 github pages 的 jekyll 模版会忽略下划线开头的文件，自动忽略下划线开头的目录，从而导致引用不到 CSS,JAVASCRIPT,ETC.，所以要禁用 jekyll

禁用方法就是在文件在项目目录下添加.nojekyll 文件

CDN

CDN 前端静态资源库 baomitu-used in maupassant-hexo

75CDN 奇舞团

9 hexo deploy 网站部署

参见 Hexo 博客从搭建部署到 SEO 优化等详细教程

9.1 hexo d 法

hexo.io/docs/deployment

安装扩展：

```
$ npm install hexo-deployer-git --save
```

需要在配置文件 _config.xml 中作如下修改：

```
deploy:
  type: git
  repo: git@github.com:jiji262/jiji262.github.io.git
  branch: master
```

然后在命令行中执行

```
hexo d
```

密码输入形式

```
deploy:
  type: git
 repo: https://kevinluolog:xxxxxx[密码]@github.com/kevinluolog/  kevinluolog.github.io.git
  branch: master

#  repo: git@github.com:kevinluolog/kevinluolog.github.io.git

#例如你的账号为:crown3,密码为 BBB;
#那你的repo填写为下面这样即可
#github: https://crown3:BBB@github.com/crown3/crown3.github.io.git
#coding: https://crown3:BBB@git.coding.net/crown3/仓库名.git
```

9.2 直接 **git clone** 法

9.3 **CI** 法，

hexo_klblog

这个网址里面提到了常用的持续集成 CI 工具:

好代码是管出来的——使用 GitHub 实现简单的 CI/CD

Hexo 遇上 Travis-CI：可能是最通俗易懂的自动发布博客图文教程

GitHub 的 CI/CD 与 Travis 配置小记

9.3.1 十大 **CI** 工具**:**

```
Travis CI
Circle CI
Jenkins
AppVeyor
CodeShip
Drone
Semaphore CI
Buildkite
Wercker
TeamCity
```

9.3.2 **travis CI**:

Travis 可以执行多种语言的测试及构建，官方文档

Build Lifecycle documentation

9.3.2.1 travis CI 配置步骤**:** 那既然需要使用 travis 自动化更新你的博客，travis 自然需要读写你的 github 上的 repo。github 提供了 token 机制来供外部访问你的仓库。

https://github.com/settings/tokens

1. 安装 travis, 并授权管理 repo

   marketplace 搜索 travis, 并安装。

   github/{user name}/personal settings/application/ travis CI configure 选择要 travis 管理的 repo.

2. 配置 github token

   分三步，

   ```
   step 1: 在github中生成token，
        github/{user name}/-> setting->Developer settings-> Personal access tokens -> generat
   step 2: 再在travis网页项目中配置环境变量$GH_TOKEN，填入token
   step 3: 在blog root _config.yml deploy 中，设置gh_token访问标记
   step 4: 在.travis.yml中，在编译完后，deploy之前用sed替换，step 3 中在 _confi.yml 中的gh_toke
   ```

3. 配置 travis

   在 travis 进入仓库同步管理, here

   主要是前面的 gh-token 环境变量.

4. 在源码仓库根目录增加.travis.yml 修改 _config.yml

   - 增加.travis.yml

     注意，如果源码是在分支上要修改 branches 为相应的分支名，缺省是 master:

     ```
     ### for branch of hexo-next-muse

     # 指定语言环境
     language: node_js
     # 指定需要sudo权限
     sudo: required
     # 指定node_js版本
     node_js:
       - 10.16.3
     # 指定缓存模块，可选。缓存可加快编译速度。
     cache:
       directories:
         - node_modules

     # 指定博客源码分支，因人而异。hexo博客源码托管在独立repo则不用设置此项
     branches:
       only:
         - hexo-next-muse

     before_install:
       - npm install -g hexo-cli

     # Start: Build Lifecycle
     install:
       - npm install
       - npm install hexo-deployer-git --save

     # 执行清缓存，生成网页操作
     script:
       - hexo clean
       - hexo generate
     ```

```
  # 设置git提交名，邮箱；替换真实token到_config.yml文件，最后depoy部署
  after_script:
    - git config user.name "kevinluolog"
    - git config user.email "kevinluolog_72@163.com"
   # 替换同目录下的_config.yml文件中gh_token字符串为travis后台刚才配置的变量，注意此处sed
    - sed -i "s/gh_token/${GH_TOKEN}/g" ./_config.yml
    - hexo deploy
  # End: Build LifeCycle
```

- 修改 _config.yml

  1. 主要是修改 deploy 部分，决定 gh-token 和推送部署到什么 repo 的什么分支。如果是 xxxx.github.io 就推到 master, 如果是子目录 repo, 则推送到 gh-pages 分支。

  2. 设置 root 变量，如果是子目录 repo，则需要设置相应的子目录 repo 名字。这样在网页引用 css 等资源时可以直接引用到。因为 hexo 引用 CSS 等资源时用的是绝对目录，如/{子目录 repo 名}/css/xxx.css, sphinx 用的是相对目录，如 _static/css/xxx.css。此处因为 sphinx 资源目录前面带了下划线，_, 因 hexo 和 jekyll 会自动忽略下划线开头的目录，从而导致引用不到 CSS,JAVASCRIPT,ETC.，所以在根目录要添加.nojekyll 文件, 详细请参考怎么解决 githubpages 不能识别下划线开头的目录？

```
root: /hexo-next-muse/
# Deployment
deploy:
  type: git
  repo: https://gh_token@github.com/kevinluolog/hexo-next-muse.git
  branch: gh-pages
```

9.3.3 travis CI 配置实例

网站规划结构参见my deploy: kevinluolog.github.io

简单讲就是，kevinluolog/hexo-klblog-src repo 作为源码仓库，master 分支对应主网站 repo 的 master 分支，其余分支各对应主网站的子目录网站 repo 的 gp-pages 分支。源码 repo 分支名称和子目录网站的仓库名要取得一样，以方便对应。每次源码有推送时，触发对应分支的 travis CI 启动，源码拉取-> 环境搭建-> 编译-> 部署，部署时是部署到对应的网站 repo 的 gp-pages 分支上的。

最终的效果是，写或修改文章时只和 source 目录中的 _post 目录相关 `\hexo\klBlog\source\_posts` 改动完成提交到对应的分支后，过 2 分钟左右，对应的网站即要自动更新。非常方便和快捷，不用占用本人的时间也不用占用本机的 CPU 去编译和部署。同时可以的任何可以上网的地方写文章，提交。和写程序一模一样，一个 github 搞定一切。

9.3.3.1 创建源码新分支需要改动的文件

1. 文件.travis.yml

   启动分支名。

```
# 指定博客源码分支，因人而异。hexo博客源码托管在独立repo则不用设置此项
branches:
  only:
    - hexo-next-xxx
```

2. 文件 _config.yml

   - theme

```
theme: next
```

- root, 子网站根目录，要和网站 repo 名字相同

```
root: /hexo-next-xxx/
```

- deloy, 推送目标仓库名

```
deploy:
  type: git
  repo: https://gh_token@github.com/kevinluolog/hexo-next-xxx.git
  branch: gh-pages
```

3. 其它博客定制相关文件

   如 theme 下配置文件 _config.yml，next 和 melody 支持独立配置文件在 _data/next.yml 和 melody.yml

   i. _data/next.yml
   - 风格

```
# Schemes
#scheme: Muse
#scheme: Mist ##kl+
#scheme: Pisces
scheme: Gemini
```

9.4 需求：在 github pages 子目录建立 hexo 博客

在 githubpages 子目录建立 hexo 博客

实现：

1. 首先建立 xxx.github.io 的 repo，xxx 是你的用户名，之后开启 github pages 服务
2. 再建立一个 bbbb 的 repo，bbbb 是你想要的子目录
3. 设置 hexo 的 deploy 配置文件 _config.yml

```
- type: git
  repo: https://github.com/xxx/bbbb.git
  branch: gh-pages
```

4. 修改 _config.yml 中的 root 选项，由”/” 改为”/bbbb”

github page 就大概两种，一种 user page 必须 master 分支，另一种 project page 需要给对应的 project 设置一个 gh-pages 分支，上传好网页资源文件之后，就可以在 username.github.io/projectname 这样的域名访问了。

网上挺多教程都不太对，自己解决了之后记录一下。

9.5 my deploy: kevinluolog.github.io

9.5.1 Repo of sites:

1. kevinluolog.github.io: branch:master

   generated by travis Ci from repo of hexo source - branch of master

2. hexo-XXX: branch:gh-pages

   can be accessed by `kevinluolog.github.io/hexo-xxx`

generated by travis Ci from repo of hexo source - branch of hexo-XXX

3. gp: branch:gh-pages

   can be accessed by `kevinluolog.github.io/gp/xxx`

   generated by sphinx etc. directly

9.5.2 Repo of hexo source: private

Travis can access with private repo.

- branch: master

  hexo source, and deploy to kevinluolog.github.io - master automatically

- branch: hexo-xxx

  deploy to REPO： hexo-xxx - gh-pages, automatically.

  branch _config.yml for set differrent theme

  travis-ci triggered by source file merged in. then compile and deploy.

  books, created by sphinx or docutils or pandoc.