

ETRD博客

Electronic Technology Research&Development

hexo中完美插入本地图片

📅 2017-01-23 | 📁 [hexo使用](#) | 👁 5584

引言

Hexo一个很好的博客框架，然而插入本地图片的路径问题让人十分头疼，因为在最后生成public的html文件时，每篇博文都是生成在用年月日做文件目录的文件下，例如

```
1 \blog\public\2017\01\22\Windows下远程桌面连接树莓派3代
```

而主页的html文件又是这个文件下，与博文的目录肯定是不一样的

```
1 \blog\public
```

这与我们在

```
1 \blog\source\_posts
```

目录下编写博客md源文件时的目录结构是完全不同的。

在hexo的官方文档[资源文件夹章节](#)上，其提供了一种插入图片的方面，但是非常遗憾，它竟然是使用标签插件的方式，而不是markdown语法。据了解这实际上是一个Hexo 3之前网友为图片路径问题开发的一个插件，这个插件被集成在了Hexo 3里面。这个插件的使用方法官方给出是这样的：

```
1 {% asset_path slug %}  
2 {% asset_img slug [title] %}  
3 {% asset_link slug [title] %}
```

这种奇怪的方式我不明白为什么会被官方采纳，因为它最大的弱点在于，在编写文章时，你看到的是一行标签代码，而不能显示图片，也就是使用markdown编辑器进行文章编写时，不能可见即所得。

Hexo环境

Hexo环境的搭建看过不少文章，但对我而言是极其繁琐和陌生的，而我的目的主要在于利用它来写博客，我在意的是我博客的内容而不是搭建博客自身的过程，因此我是直接使用了[Hexo懒人携带版 / PortableHexo](#)，它的优势是，纯傻瓜式操作，无法安装hexo git nodejs等环境，压缩包解压直接用，直接发布，十分感谢这个懒人包团队的工作。该懒人包内相关软件的版本如下：

```
1  版本 / Version
2  Git: 2.7.4
3  Nodejs: 4.4.0
4  Npm: 3.8.3
5  Hexo: 3.2
```

md编辑器

我使用的是md编辑器为[Typora](#)，这个软件的方便之处在于直接将markdown语法呈现出其效果，并且已经集成了很多方便的功能，你无需记住插入代码的语法格式，无需记住插入图片的语法格式，这使得我可以专心写博客而无法关心学习markdown语法。有很多md编辑器，他们分成两个界面，一个界面是md源代码，另外一个界面实时更新显示其效果，而Typora是浑然一体的，真正所见即所得，就跟在使用word一样方便。

并且，它可以通过插件免费的将md文件生成pdf（自动生成标签），word，便于笔记的存档和离线交互，完美。

Hexo配置文件的设置

确保你的Hexo的配置文件_config.yml里面有个这个选项配置，并将其置为true

```
1  post_asset_folder: true
```

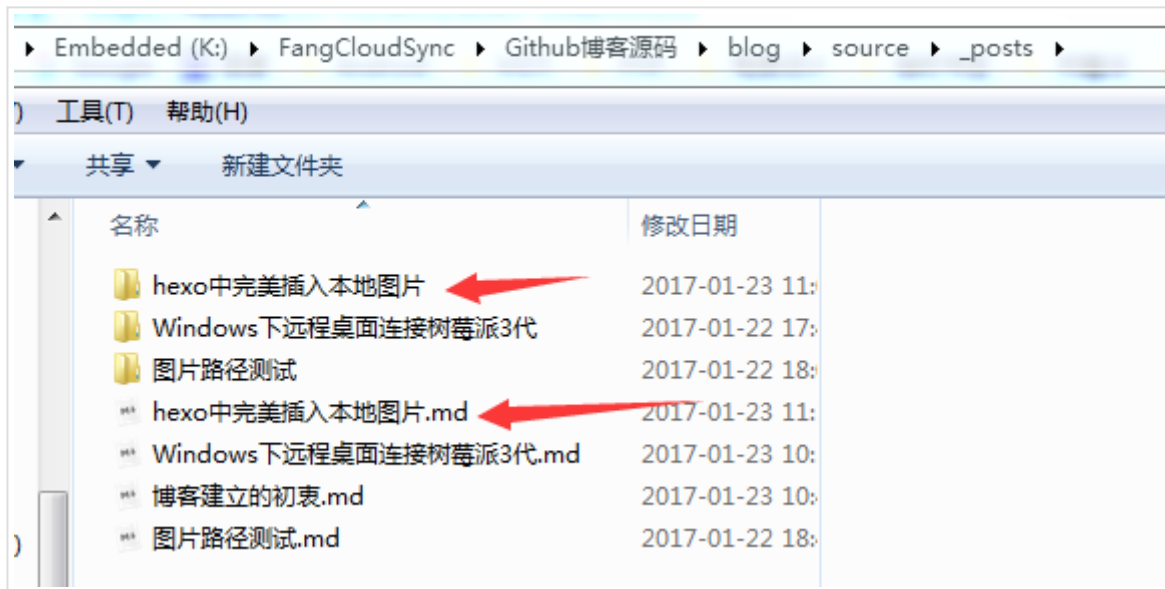
这个功能实际上是Hexo官方文档中提到的资源文件夹功能，它的作用在于当你使用

```
1  hexo new
```

生成一篇新文章时，会在

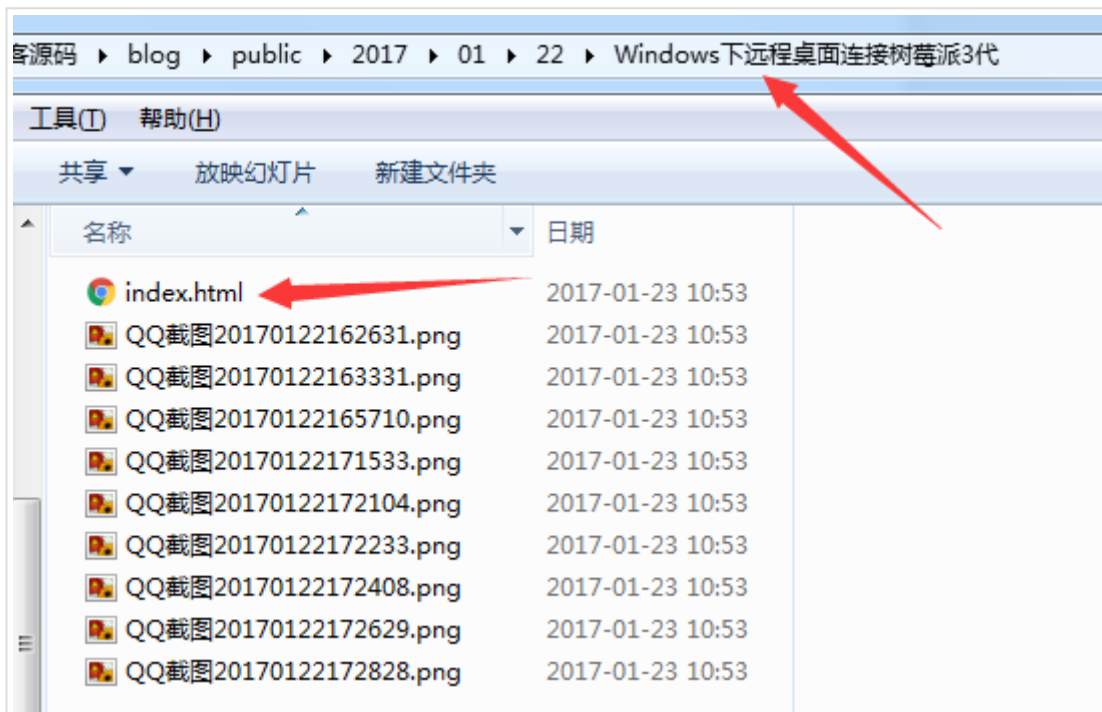
```
1  \source\_posts
```

目录下生成一个 文章名.md 文件外，附带生成一个与 文章名 同名的文件夹，可以用它来存放这篇文章的所有资源，比如图片，附件等



实际上我通常喜欢自己右键手动去建立一个md文件来写博客，而不使用hexo new命令，此时要达到同样的效果，我同样也需要手动去建立一个同名的文件夹去存放资源，我试过这样做是可行的，并且这样做虽然有点麻烦，但我似乎发现了它的一个优点，那就是当我的文章中没有图片时，我并不需要这个空的资源文件夹。

有了上面的配置，Hexo在public文件下生成html文件时，会在相应的目录中把那文章对应的资源文件夹中的图片等附件复制过去，例如，这是我生成的一个例子：



安装hexo-asset-image插件

你需要安装一个图片路径转换的插件，这个插件名字是[hexo-asset-image](#)，我目前安装的是最新的0.03版本，使用如下命令安装插件

```
1 npm install https://github.com/CodeFalling/hexo-asset-image --save
```

插件安装完以后，实际上是存放在下面的目录文件夹下

```
1 \blog\node_modules\hexo-asset-image
```

想要卸载这个插件怎么处理，我听说直接删掉这个文件夹，反正我这么试过了。

md文件的时间标签

hexo-asset-image的原理应该是根据Hexo生成html后以时间的为目录的规则，将我们图片的相对地址转换成了生成后与时间对应目录的绝对地址，因此我认为md文件中应该要有时间标签，一来是为了让Hexo根据md的时间标签来确定文章的新旧排序，如果没有时间标签，Hexo会把md文件的修改时间作为发布时间，这显然是不合理的，再者这个时间标签，我想hexo-asset-image也会引用到，在一篇实际的博文中，我的md文件完整的开头是这样的：

```
1 ---
2 title: hexo中完美插入本地图片
3 date: 2017-01-23 11:00:55
4 categories: hexo使用
5 tags:
6 - hexo
7 - 本地图片
8 - hexo-asset-image
9 ---
```

插入图片

由于每个md文件都有对应的一个资源文件夹，因此，你首先要将需要插入的图片放到这个资源文件夹下，然后进行路径引用，这是上面插入图片时的实际引用代码

```
1 ![QQ截图20170123114005](/hexo中完美插入本地图片/QQ截图20170123114005.png)
```

使用Typora又有一个十分方便的地方在于，只需要将图片往这里一拖就插入了，但其实际拖入图片后的插入代码是绝对路径的引用，是这样的

```
1 ![QQ截图20170123114005](K:\FangCloudSync\Github博客源码\blog\source\_posts\hexo中完美
```

对比上面的两个代码，我只需要将前面的那段绝对路径删掉~~K:\FangCloudSync\Github博客源码\blog\source_posts~~，然后再将剩余的两个~~\~~分别换成/就可以了

这样子，Typora编辑时同时能够看到插入的图片。

生成测试

在很多网友的文章中，这么做就可以了，你可以尝试一下生成并本地测试，看看是否博客的首页及博文的正文页都能够正确显示图片，或者打开博文生成的html源文件，找到图片插入的地方看看src路径是否是对应public后的绝对路径，如下示例这样

```
1 <p></p>
```

看起来hexo-asset-image的确起了点作用，但似乎有点问题，它生成的绝对路径不完整，仿佛只获取了日，年月没获取到，因为这个md文件的时间是2017-01-22，所以"/2017/01" 似乎没有生成啊，对比上面正确正常的绝对路径，我不明白这是是什么原因导致的，也许有可能是我使用的hexo懒人包导致的，Google了一晚上好像就我一个人有这种情况，最后我也在Github上向该插件的作者递交了我遇到的麻烦，但估计等到作者回复不知道是何时了。

今天早上，我受到[自定义修改插件，更便捷的使用七牛空间](#) 这篇文章的启发，决定自己也尝试修改插件代码试试能否解决。虽然完全不懂js代码，但还是尝试阅读了那段极其简练的代码，这是其中一段：

```
1 var link = data.permalink;
2 var beginPos = getPosition(link, '/', 3) + 1;
3 // In hexo 3.1.1, the permalink of "about" page is like ".../about/index.html".
4 var endPos = link.lastIndexOf('/') + 1;
5 link = link.substring(beginPos, endPos);
```

可以猜测出来这段点大概是在获取博文生成的绝对路径，然后最后与md文件中的相对路径做一个拼接，但这个函数中

```
1 var beginPos = getPosition(link, '/', 3) + 1;
```

为什么是寻找第3个' / ' ? 虽然我不知道, 但这样做肯定有作者的意图, 只是我不懂其中的原理不理解, 不可能这个关键的数字是错的。但我假设如果完整的路径应该是/2017/01/22, 那么第3个/似乎正好是日期里面的日, 跟我的html生成的不完整的路径点似乎有点关联, 因此如果这个数字修改为1, 那么是不是就能将/2017/01补上了, 试试看吧, 然而, 幸运的是, 成功了!

```
1 var beginPos = getPosition(link, '/', 1) + 1; //just by launcher
```

小结

需要插入本地图片, 并让Hexo启用资源文件的功能, 每个md文件对应一个资源文件夹, 然而这只是方便了资源的管理, 插入图片时不推荐使用hexo官方说的不伦不类的标签插入方法, 因为编辑md文件时无法预览图片, 而需要安装hexo-asset-image插件, 这样正常使用markdown语法对图片的相对路径引用, 即可实现编辑时可预览图片, 且发布时插件自动将相对路径转为绝对路径。如果碰到与我同样的生成的绝对路径不完全的情况, 尝试修改插件的js代码, 但修改器建议先备份它, 我认为我所解决我遇到问题的方法可能是一种巧合, 但我暂时先这么使用了, 最终的原因和处理方法可能还需要等待Github上该插件作者的官方回应。

参考文献

<https://github.com/Bitmoe/PortableHexo>

<https://typora.io/>

<https://hexo.io/zh-cn/docs/asset-folders.html>

<http://www.tuicool.com/articles/umEBVfi>

<https://github.com/CodeFalling/hexo-asset-image>

<http://error408.com/2016/08/02/Hexo%E4%B8%83%E7%89%9B%E5%9B%BE%E5%BA%8A%E4%BD%BF%E7%94%A8/>

[# hexo](#) [# 本地图片](#) [# 路径转换](#) [# hexo-asset-image](#)

[◀ Windows下远程桌面连接树莓派3代](#)

[Windows下建立第一个树莓派应用程序 ▶](#)


撰写评论

发布

账号 (邮件地址)

评论 2

时间正序 时间倒序 同感正序



无风引漩 2019.02.15 08:54

感谢博主


另：卸载npm包的方法是

...

npm uninstall hexo-asset-image

...

000



NO NICKNAME 2018.11.01 04:05

感谢博主，我只安装了这个插件并把图片插入以如下格式即可正常预览：

![eps](./dbscan/eps.png)

![图片别名](./文章名/图片.png)

000

© LiveRe.