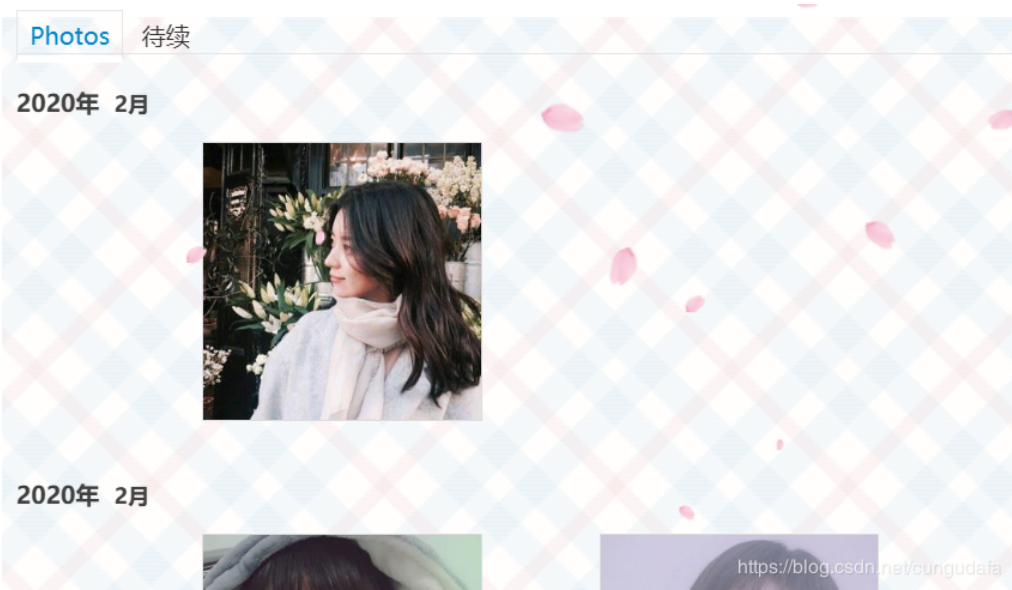


Github+Hexo博客增添相册功能

前言：hexo的sakura主题默认是不含有相册页面的,如果想添加此页面需自行编写
效果图预览，

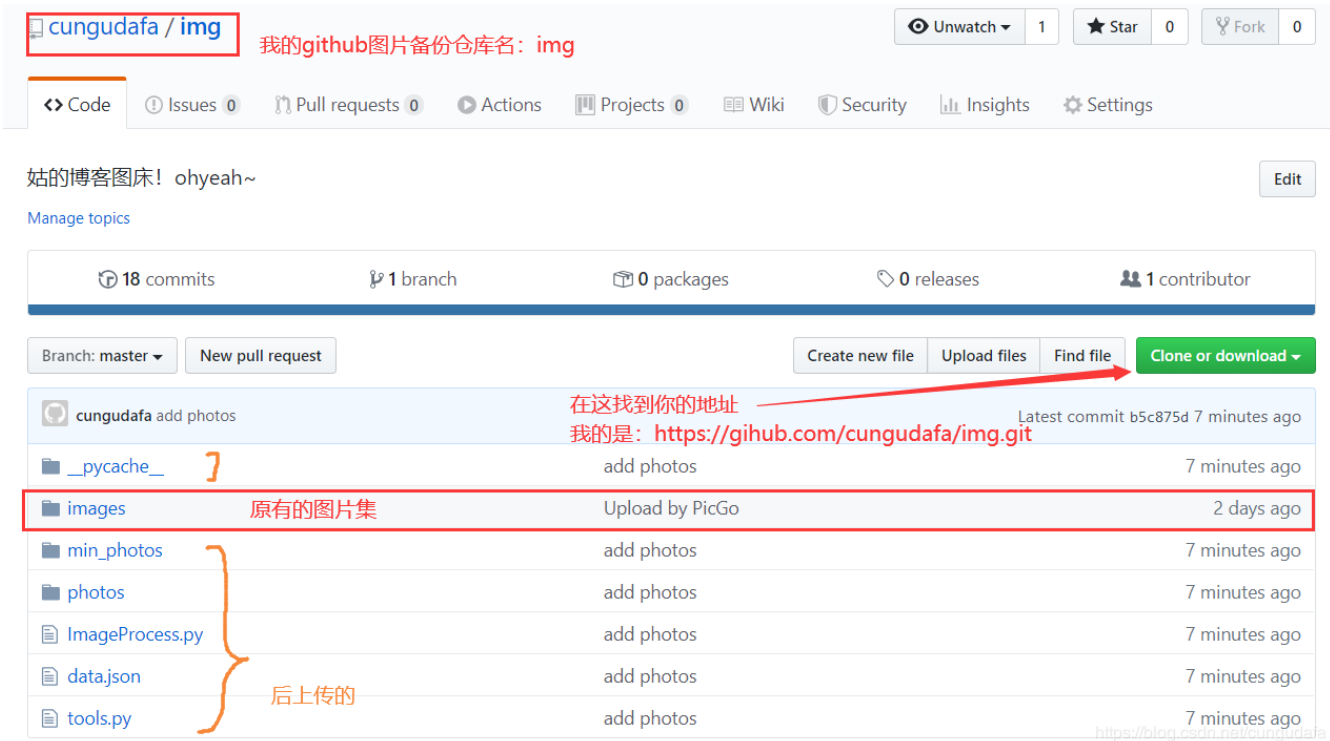


思路：
不能把所有图片都放在source静态资源下，一是数据量太大，增加Hexo的负担（所以我们把图片放在github），二是加载高清图资源缓慢（所以我们用python对图片进行批量压缩处理），这样加载呈现压缩图，点击才加载原图。

本文参考于Hexo+Github实现相册功能作者：LawLite，原文主要是思路，实践是有坑的，这又是一篇填坑的文章。

一、图片的处理

1. 首先资源图片是是存放在GitHub上,你在github准备一个仓库。
我github名为cungudafa，仓库名为img，这里找到你的克隆地址，我的地址为：
<https://github.com/cungudafa/img.git>



2. 本地准备一个文件夹来存储和管理图片(Git管理，需要提前安装)

1. 在电脑上随便一个位置创建一个新的文件夹，这里我创建文件夹名称叫 `Blog_Album`
2. 然后打开git，cd到此文件夹执行 `git clone https://github.com/cungudafa/img.git` 命令克隆github上仓库。

```
1 | git clone https://github.com/你的github名/仓库名.git
```

克隆完毕在 `Blog_Album` 目录下就会有img这个文件夹（以下img都指这个目录也指我的github仓库名），里面有.git文件和其他图片资源。（我们后序用脚本快速处理相册，会使用到这个.git）

3. 在img文件夹内新建一个文件夹 `/photos` 在这里存放你需要放在博客中的图片。（存放名称格式如下：`2020-02-02_discriptionofyourpic.jpg`；最前面是日期，然后用_进行分隔；后面是图片的描述信息，注意不要包含_和.符号，最好也不要使用中文命名）

1GitHub_cungudafa > cdn > Blog_Album > img > photos

名称	日期	类型	大小
2020-02-18_wang.jpg	2020/2/18 12:59	JPG 文件	735 KB
2020-02-18_wangmeili.jpg	2020/2/18 12:59	JPG 文件	1,029 KB
2020-02-18_xiaowang.jpg	2020/2/18 12:59	JPG 文件	1,013 KB

附：python脚本（需要有python环境哦）快速处理图片命名：（下载ImageRename.py，在cmd中运行 `python ImageRename.py`，能快速在名称前加上最新一次修改日期）

```
D:\myworkspace\1GitHub_cungudafa\MyWeb\themes\sakura\source\insphoto>python ImageRename.py
wang.jpg
2020-02-18_wang.jpg
wangmeili.jpg
2020-02-18_wangmeili.jpg
xiaowang.jpg
2020-02-18_xiaowang.jpg
```

3. Hexo中我们的相册是采用ins样式显示的，为了快速加载显示图片，我们还需要制作**图片的压缩图**和**图片的序列data.json**。同样这里才用一个Python小脚本对图片进行压缩编辑，并且上传到自己的GitHub。

1. 下载脚本ImageProcess.py和tool.py文件至此文件夹内（`Blog_Album\img`）

kspace > 1GitHub_cungudafa > cdn > Blog_Album > img

名称	修改日期	类型	大小
.git	2020/2/18 17:01	文件夹	
__pycache	2020/2/18 16:43	文件夹	
images	2020/2/18 16:31	文件夹	
min_photos	2020/2/18 16:55	文件夹	
photos	2020/2/18 16:16	文件夹	
data.json	2020/2/18 16:55	JSON File	1 KB
ImageProcess.py	2020/2/18 12:47	Python File	3 KB
ImageRename.py	2020/2/18 14:21	Python File	1 KB
tools.py	2020/2/18 16:56	Python File	7 KB

其他文件
1 压缩图
2 相册
3 序列
下载到此处

说明：ImageProcess为裁剪方法，tools工具类；还创建了min_photos文件夹存放缩略图；json由tools脚本生成的，后面hexo用到的。

2. 运行tool.py脚本要安装python环境哦！（`python tool.py` 或者用Edit with IDLE也可以）
如果执行tool.py的时候如果没有PIL库的话会报错，安装即可。安装PIL只需在cmd中执行：`pip install Pillow` 就可以了。

其中，tools由LawLite大牛编写，这里引用；其中包含压缩图片，裁剪图片，github提交，json处理四个部分。（github提交部分需要把git.exe的位置（例：D:\mydownload\Git\bin）添加到path环境变量中去；由于我们是克隆的img项目，此时用到的就是.git提交，非常方便了！）

```
>>>
=== RESTART: D:\myworkspace\lGitHub_cungudafa\cdn\Blog_Album\img\tools.py ===

This program helps compress many image files
you can choose which scale you want to compress your img(jpg/png/etc)
1) normal compress(4M to 1M around)
2) small compress(4M to 500K around)
3) smaller compress(4M to 300K around)

C:\WINDOWS\system32\cmd.exe
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 186.57 KiB | 31.09 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)

https://blog.csdn.net/cungudafa
```

运行效果：
已上传内容：

cungudafa add photos			Latest commit e0654da 2 minutes ago
__pycache__	add photos		1 hour ago
images	Upload by PicGo		2 days ago
min_photos	add photos		2 minutes ago
photos	add photos		2 minutes ago
ImageProcess.py	add photos		1 hour ago
ImageRename.py	add photos		21 minutes ago
data.json	add photos		1 hour ago
tools.py	add photos		21 minutes ago

每次更新部分

cungudafa / **img**

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branch: master **img / photos /**

Create new file Upload files Find file History

cungudafa add photos			Latest commit e0654da 1 minute ago
..			
2020-02-18_wang.jpg	add photos		1 hour ago
2020-02-18_wangmeili.jpg	add photos		1 hour ago
2020-02-18_xiaowang.jpg	add photos		1 hour ago
2020-2-13_HyoJoo.jpg	add photos		1 minute ago

说明：执行后会生成 data.json 文件，此文件存储了图片的链接，名称和简介。

二、博客页面

1. 修改 themes\sakura\languages\zh-cn.yml 的 photo：图集为相册

1 | photos: 相册

2. 修改配置 themes\sakura_config.yml 修改 图集: {path: /tags/图集/, fa: fa-photo }改为或添加相册

1 | 相册: {path: /photos/, fa: fa-photo }

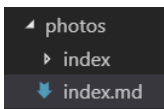
```

menus:
  首页: { path: /, fa: fa-fort-awesome faa-shake }
  归档: { path: /archives, fa: fa-archive faa-shake, submenus: {
    技术: {path: /categories/技术/, fa: fa-code },
    生活: {path: /categories/生活/, fa: fa-file-text-o },
    资源: {path: /categories/资源/, fa: fa-cloud-download },
    随想: {path: /categories/随想/, fa: fa-commenting-o },
    转载: {path: /categories/转载/, fa: fa-book }
  } }
  清单: { path: javascript:;, fa: fa-list-ul faa-vertical, submenus: {
    书单: {path: /tags/悦读/, fa: fa-th-list faa-bounce },
    番組: {path: /bangumi/, fa: fa-film faa-vertical },
    歌单: {path: /music/, fa: fa-headphones },
    相册: {path: /photos/, fa: fa-photo }
  } }
}

```

<https://blog.csdn.net/cungudafa>

3. 在博客根目录下添加photo页面 `hexo n page photos` , 会在source下创建以下目录 :



4. 把data.json复制到 \photos 下, 为了方便, 也可以修改tools.py的大约135行位置处, 以后在运行脚本时, 直接保存到此处 :

```

1  ...
2      # 这里可以直接修改到你的hexo资源目录
3      #with open("data.json","w") as fp:
4      with open("D:/myworkspace/1GitHub_cungudafa/MyWeb/source/photos/data.json","w") as fp:
5          json.dump(final_dict, fp)
6          json.dump(final_dict, fp)
7  ...

```

5. 在photos中的 index.md 加载相册内容部分。

说明：原作者用的index.ejs（主题yilia），我修改到index.md中（主题是sakura），由此生成index.html；如果需要修改样式或添加内容，改这里即可。

6. 因为相册渲染用不到sakura主题，所以还需要额外添加css文件和js文件，我的样式链接js_css，将js和css中的内容也复制到 MyWeb\source\photos 下；

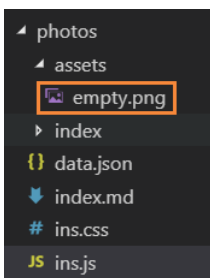
kspace > 1GitHub_cungudafa > MyWeb > source > photos					搜索"photos"
名称	修改日期	类型	大小		
assets	2020/2/18 19:30	文件夹			
index	2020/2/18 20:13	文件夹			
data.json	2020/2/18 18:02	JSON File	1 KB		
index.md	2020/2/18 20:39	MD 文件	1 KB		
ins.css	2020/2/18 12:13	层叠样式表文档	3 KB		
ins.js	2020/2/18 20:22	JS 文件	26 KB		

复制到此处 (指向 assets, index, data.json, index.md)

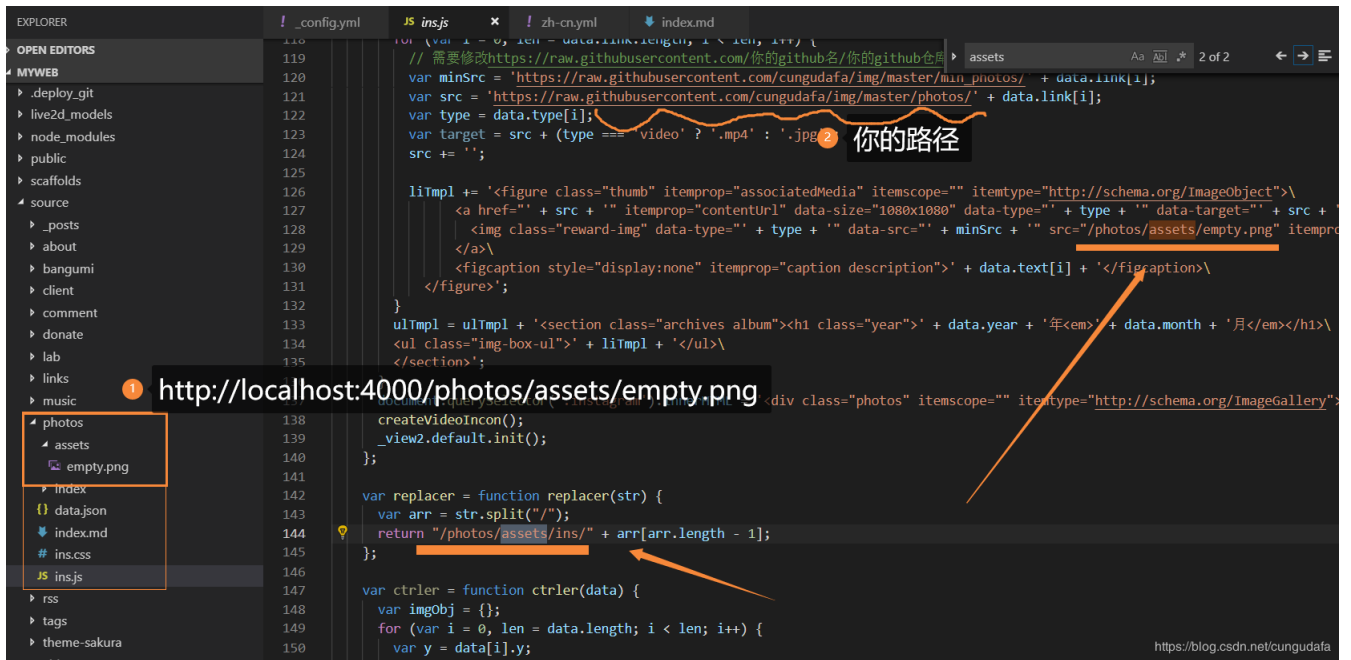
修改 (指向 ins.css, ins.js)

<https://blog.csdn.net/cungudafa>

说明：（1）assets目录下放的是一张empty.png图片，原作者在ins.js中引用了此图片也没有特别说明，我的链接里给出来了，补坑；



（其次，原作者的assets和photos同级目录，我放在photos目录里，特别修改了这两个位置，下载我的js已经修改过了。）

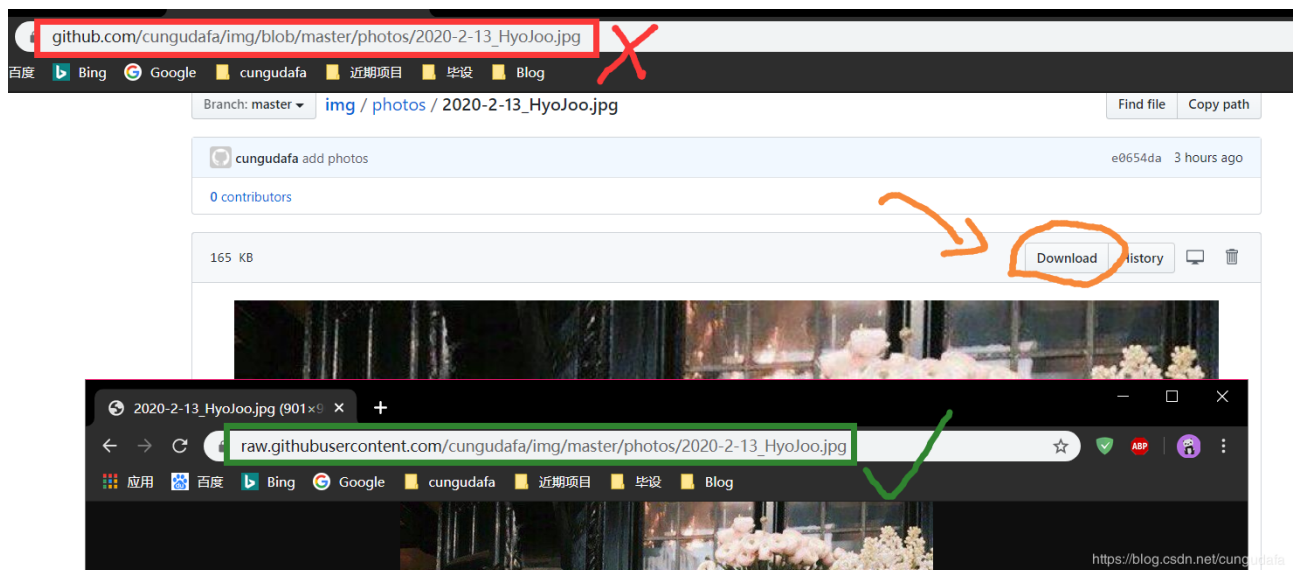


特别修改：（2）自己动手，修改 `ins.js` 里大约在119行左右的链接修改为你的GitHub的仓库地址：（我的示例：`cungudafa/img` 修改为 `你的github名/你的github仓库名`）

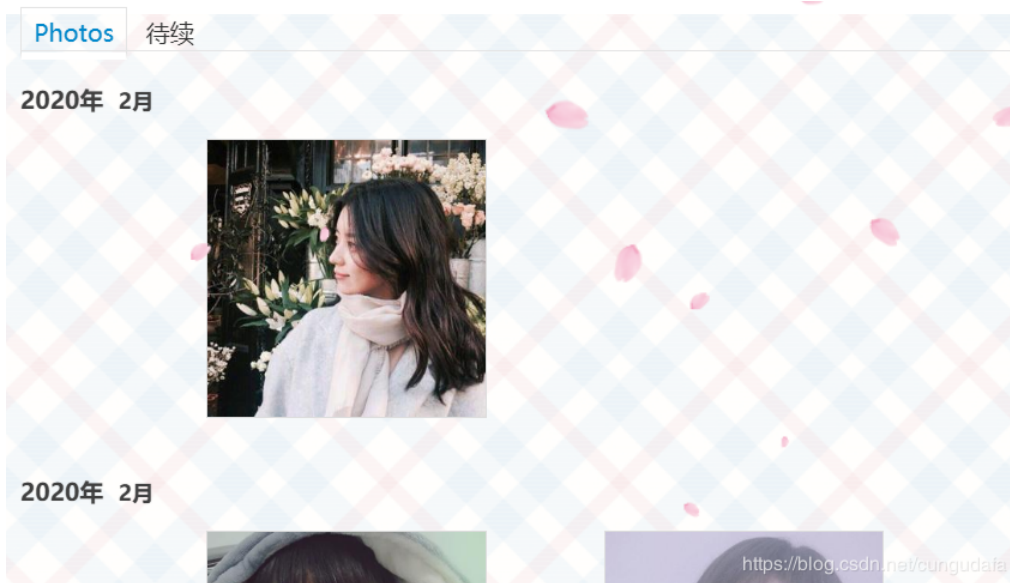
```
1 // 需要修改https://raw.githubusercontent.com/你的github名/你的github仓库名/master/min_photos/即可
2 var minSrc = 'https://raw.githubusercontent.com/cungudafa/img/master/min_photos/' + data.link[i];
3 var src = 'https://raw.githubusercontent.com/cungudafa/img/master/photos/' + data.link[i];
```

注意： 确保你的图片地址是正确的，否则加载不出来，检测方法可以将选择一个图片地址看能不能再浏览器中打开。

如图是点击 **download** 下载后的链接，是 `https://raw.githubusercontent.com` 开头的，不是图片的存储地址，否则加载不出。



7. `hexo clean` 与 `hexo g` 直接生成 `photos\index.html`



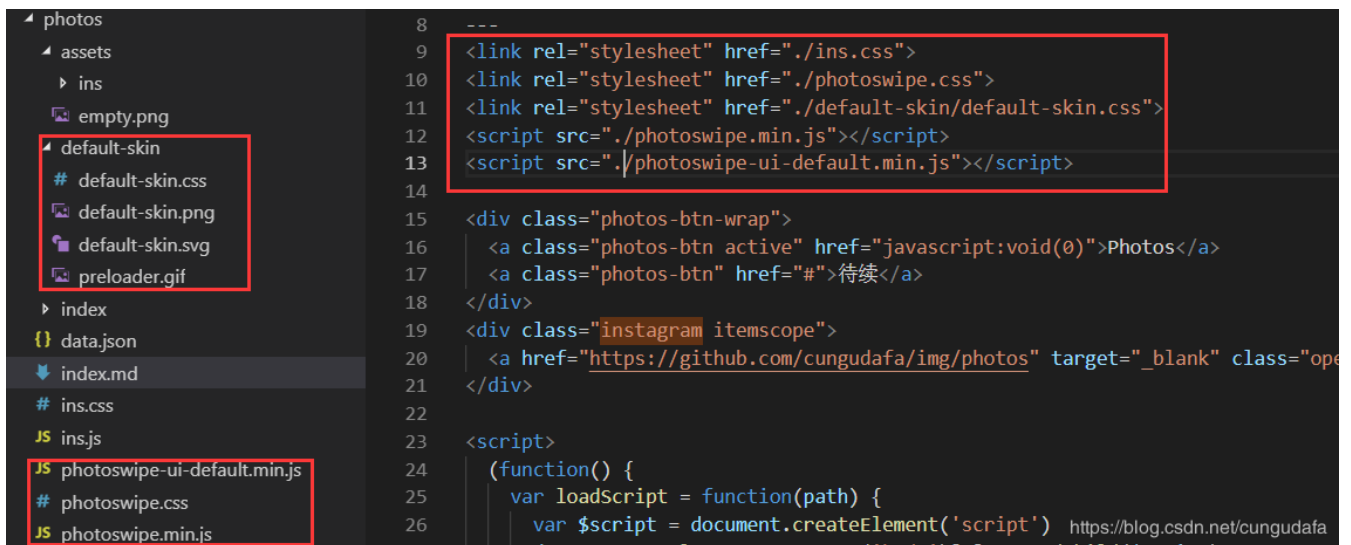
到这里，博客相册功能已经实现了，本文已经结束！
点击可查看效果：我的相册

现在有个问题是不能点击查看大图！坑二：作者用的主题融合了photoswipe插件功能，其他主题或者任意html要想使用，还得单独导入相关css和js。

三、其他配置

1. 下载样式文件（我的img图库已经很大了，不方便下载，这里给一个前人的下载链接）

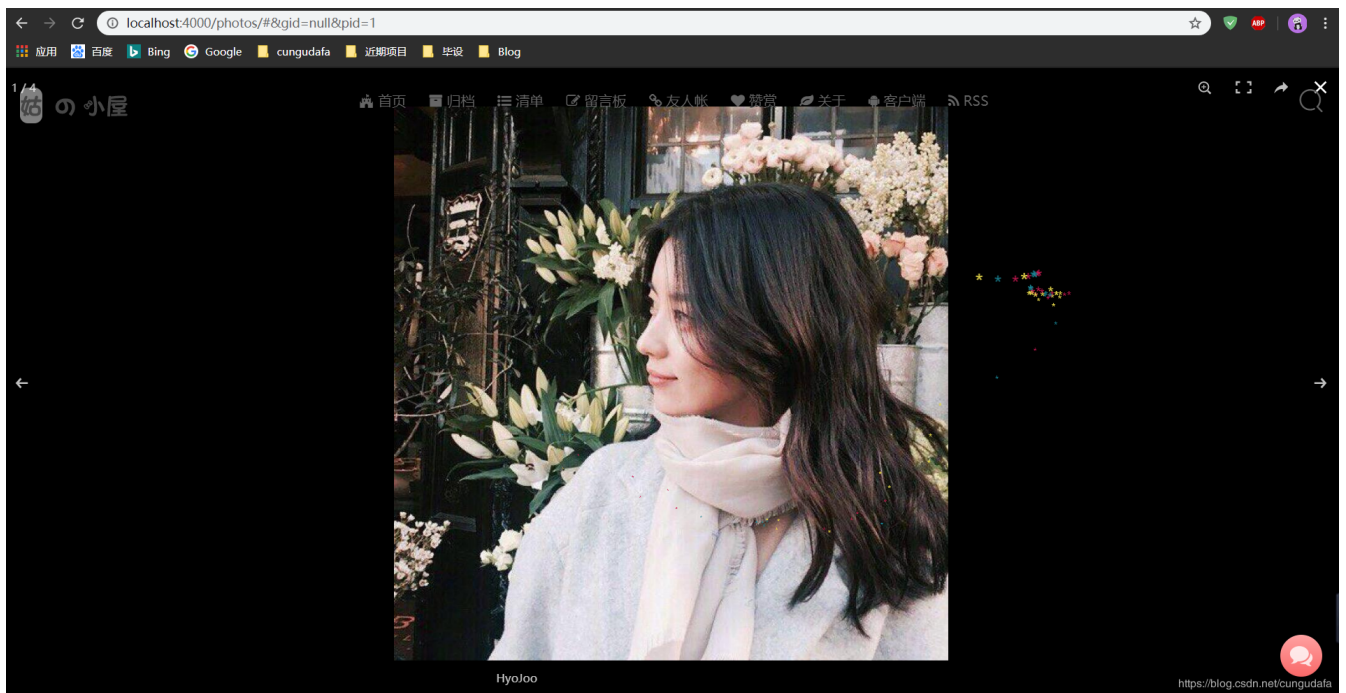
- 1.1 `photos` 目录下添加 `photoswipe.css`、`photoswipe-ui-default.min.js`、`photoswipe.min.js`，
- 1.2 `photos` 目录下新建 `default-skin` 目录，添加如下文件：



1.3 在 `source\photos\index.md` 相册文件前加上样式：

```
1 <link rel="stylesheet" href="./ins.css">
2 <link rel="stylesheet" href="./photoswipe.css">
3 <link rel="stylesheet" href="./default-skin/default-skin.css">
4 <script src="./photoswipe.min.js"></script>
5 <script src="./photoswipe-ui-default.min.js"></script>
```


查看大图：（手机端也适应哦！）



2. 如果想要全局都使用photoswipe效果的话，可以把js和css配置在theme主题中，在head、footer分别加上相应引入，额外还要添加一些配置哦：
photoswipe 移动端PC端兼容图片相册预览。

现在项目相册部分完美！接下来是视频部分~~ 踩坑、填坑、学习记录 走起

👍 点赞 ☆ 收藏 📄 分享 ...



cungudafa

发布了247 篇原创文章 · 获赞 490 · 访问量 19万+

私信

已关注