
learn-Build

2020 年 06 月 17 日

从零搭建产品文档

1	快速搭建 Sphinx 之基础准备	3
1.1	安装 python	3
1.2	安装 sphinx 并且快速构建一个文档	3
1.3	配置文件修改	6
1.4	快速了解文档规则	7
2	使用 Markdown 来编写文档	9
3	Github 托管代码	11
4	导入 ReadtheDocs 并建立钩子	13

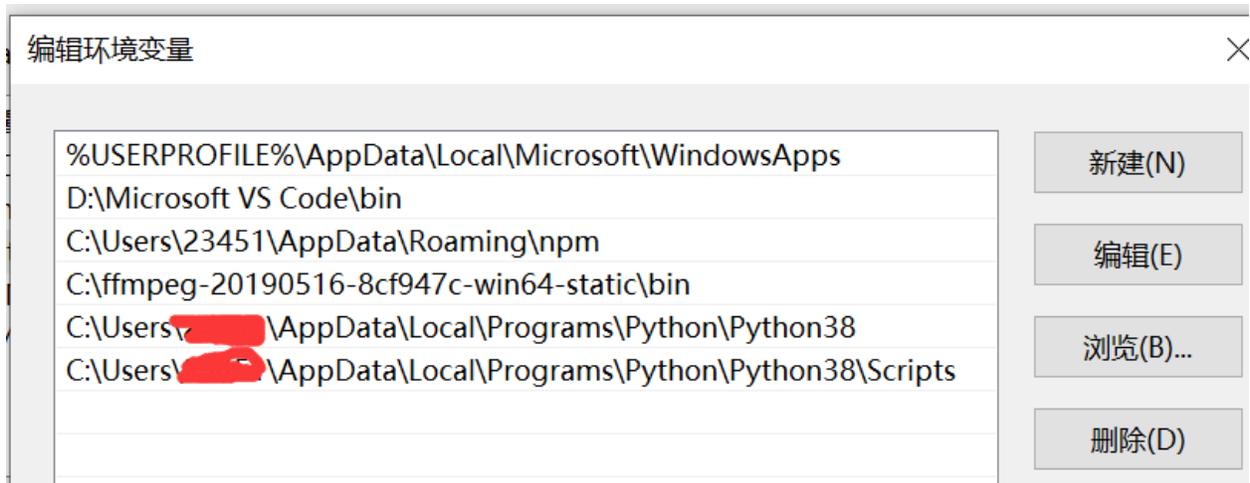
提示: 本篇文档将按步骤教你如何从零搭建开始搭建一个在线技术文档。

快速搭建 Sphinx 之基础准备

1.1 安装 python

由于 sphinx 依赖于 python, 所以我们需要先安装 python

- 点击随指引安装 python [install Python](#)
- 添加 python 和 pip 的环境变量



1.2 安装 sphinx 并且快速构建一个文档

- cmd 安装 sphinx

“bash pip install sphinx

““

- 接着在 cmd 进行如下操作，建立 sphinx 框架文档

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.592]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\<redacted>>e:
E:\>mkdir learn-Build
E:\>cd learn-Build
E:\learn-Build>sphinx-quickstart
Welcome to the Sphinx 2.4.0 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Selected root path: .

You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/n) [n]: y

The project name will occur in several places in the built documentation.
> Project name: learn-Build
> Author name(s): hogan
> Project release []:

If the documents are to be written in a language other than English,
you can select a language here by its language code. Sphinx will then
translate text that it generates into that language.

For a list of supported codes, see
https://www.sphinx-doc.org/en/master/usage/configuration.html#confval-language.
> Project language [en]:

Creating file .\source\conf.py.
Creating file .\source\index.rst.
Creating file .\Makefile.
Creating file .\make.bat.

Finished: An initial directory structure has been created.
```

作者名和项目名自由

空的地方直接回车

- 可以看到文件夹下面已经初始好了基本文件

此电脑 > 本地磁盘 (E:) > learn-Build

名称	修改日期	类型	大小
build	2020/2/12 20:37	文件夹	
source	2020/2/12 20:37	文件夹	
make.bat	2020/2/12 20:37	Windows 批处理文件	1 KB
Makefile	2020/2/12 20:37	文件	1 KB

- 在有 make.bat 的文件目录下，构建 sphinx 快速开始预设的的基础文档

```
E:\learn-Build>make html
Running Sphinx v2.4.0
making output directory... done
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 1 source files that are out of date
updating environment: [new config] 1 added, 0 changed, 0 removed
reading sources... [100%] index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] index
generating indices... genindexdone
writing additional pages... searchdone
copying static files... .. done
copying extra files... done
dumping search index in English (code: en)... done
dumping object inventory... done
build succeeded.
```

The HTML pages are in build\html. ✓

- 构建好的静态页面就是这个 index.html。有了本地构建，我们就能在每次写完文档上传前查看效果

E:\learn-Build\build\html			
名称	修改日期	类型	
文件夹	2020/2/12 20:44	文件夹	
_sources	2020/2/12 20:44	文件夹	
_static	2020/2/12 20:44	文件夹	
.buildinfo	2020/2/12 20:44	BUILDINFO 文件	
genindex.html	2020/2/12 20:44	HTML 文档	
index.html	2020/2/12 20:44	HTML 文档	
objects.inv	2020/2/12 20:44	INV 文件	
search.html	2020/2/12 20:44	HTML 文档	
searchindex.js	2020/2/12 20:44	JavaScript 文件	


```
40
41
42 # -- Options for HTML output -----
43
44 # The theme to use for HTML and HTML Help pages. S
45 # a list of builtin themes.
46 #
47 html_theme = 'sphinx_rtd_theme' ←
48
49 # Add any paths that contain custom static files (s
50 # relative to this directory. They are copied after
51 # so a file named "default.css" will overwrite the
52 html_static_path = ['_static']
```

- 再次 ‘make html’构建文档，可见样式和 learn 一样了

2. 添加两行代码，确保后续构建没问题

```
““bash source_suffix = [ '.rst' , '.md' ] master_doc = 'index'
```

```
““
```

1.4 快速了解文档规则

文档的编写默认是支持 reStructuredText 语言，也可以通过加载插件的方式支持 markdown，但我个人推荐使用 reStructuredText，因为它的使用更灵活，方式更多样，但同时也比较复杂。‘语法参照 [syntax](#)

文档内容放置在 source/文件夹下即可，文档的入口是 index.rst(也就是说打开文档右侧显示的内容就是 index.rst，与此同时左侧的目录也是由 index.rst 文件中配置生成，具体可以参照本文档的 [github](#))。其次是在 index.rst 中我们也可以嵌套其他的 index.rst，但他们不能位于同一个目录级。

使用 Markdown 来编写文档

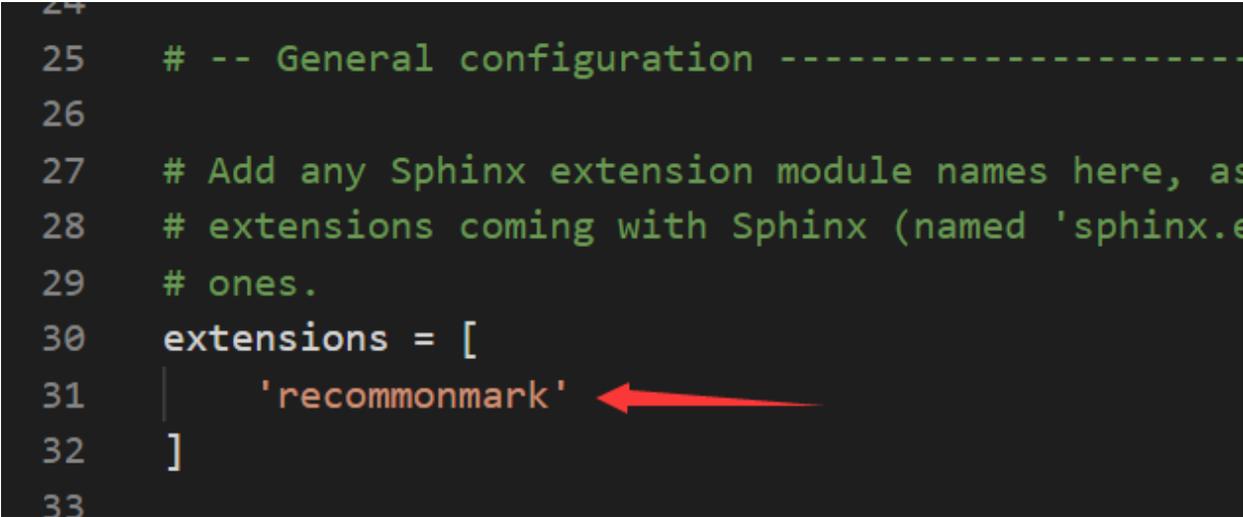
由于 Sphinx 默认的是 .rst，使用 Markdown 则需要扩展支持

1. 首先使用 cmd 本地安装

```
pip install recommonmark
```

2. 在 conf.py 配置文件中加入这个扩展

```
24
25 # -- General configuration -----
26
27 # Add any Sphinx extension module names here, as
28 # extensions coming with Sphinx (named 'sphinx.e
29 # ones.
30 extensions = [
31     'recommonmark'
32 ]
33
```



3. 此时你就可以使用 markdown 进行编写了。当你第一次新增了 markdown 页面时，先在 make.bat 文件目录下使用 make clean 清除 _build 里的构建内容，再重新 make html 构建一次观察效果

需要注意的是，md 的语法和 rst 有很大差别，在 sphinx 中，用 rst 来编写会更完美

Github 托管代码

1. 首先，在 github 创建一个仓库，与本地文件夹的同名即可，创建一个空仓库，根据命令提示进行下面操作
2. 安装 git，参照[安装教程](#)
3. 在根目录下新建一个文件取名 `.gitignore`，打开，填入如下内容

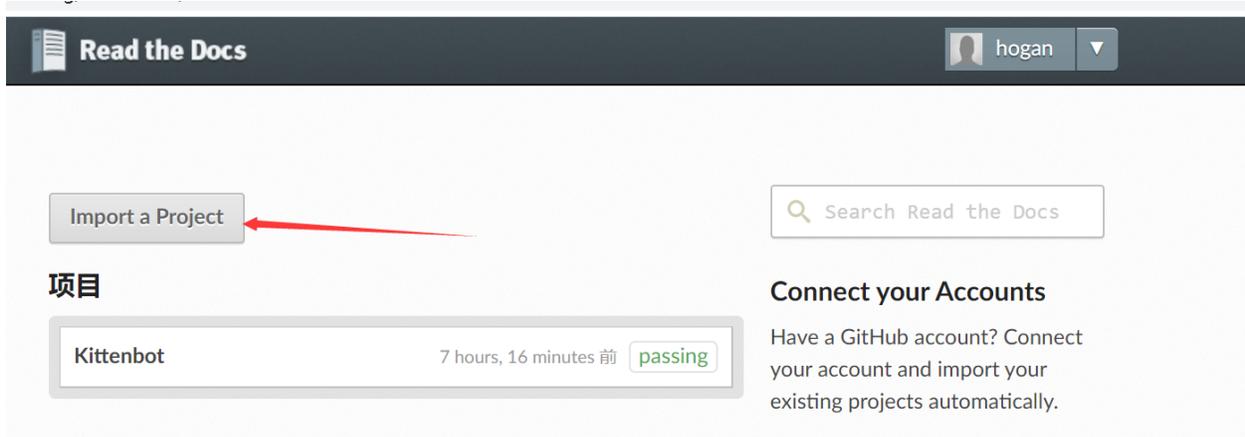
```
# sphinx build folder  
build
```

本地的 build 是无须被上传的，所以我们直接 push 时忽略掉。

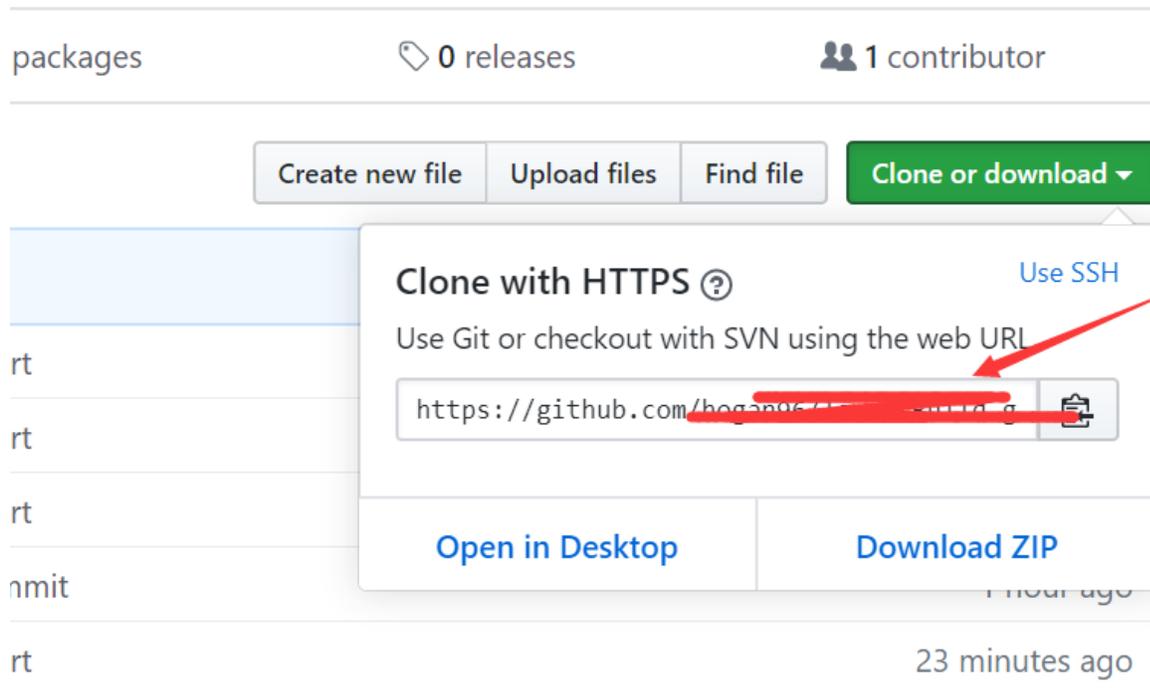
3. 安装 git 成功后，我们来到文件夹的根目录，右键空白处，可见并选择 `Git Bash Here`，随之弹出终端。根据第 1 步中的提示操作注意敲入命令使本地和远程仓库连接并推送。

导入 ReadtheDocs 并建立钩子

1. 前往 ReadtheDocs 页面，并注册账号 <https://readthedocs.org/>
2. 登录并进入我们的项目，选择 Import a Project => 手动导入



3. 填入项目名字，代码链接使用



4. 进入管理页面，在下方语选择繁体，并 Save



语言:

Traditional Chinese ▼

- Ukrainian
- Urdu
- Uspanteko
- Uzbek
- Vietnamese
- Volapuk
- Wolof
- Xhosa
- Xinka
- Yiddish
- Yoruba
- Zhuang
- Chinese
- Zulu
- Norwegian Bokmal
- Brazilian Portuguese
- Mexican Spanish
- Ukrainian
- Simplified Chinese
- Traditional Chinese**

Save or Delete project

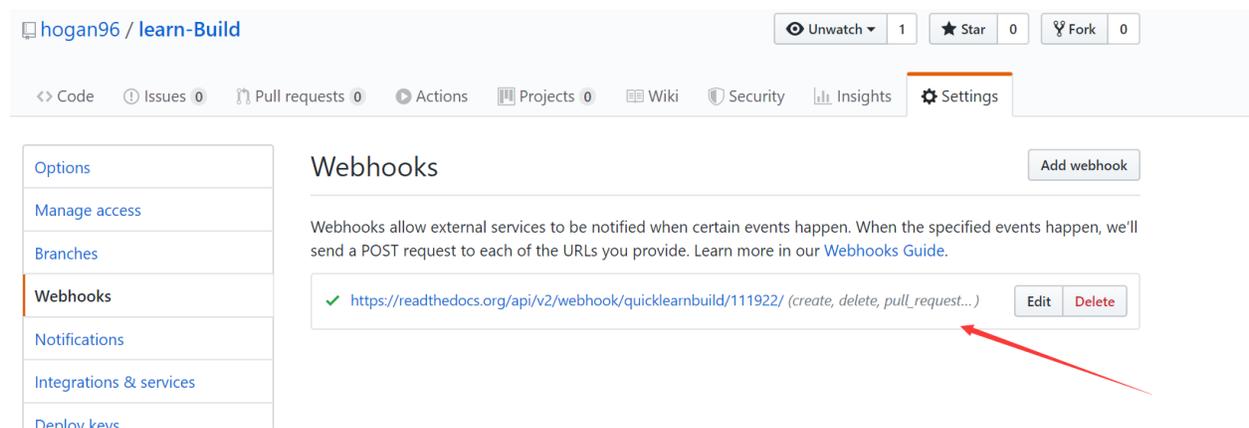
注意：这影响您的

gs.

5. 再次进入管理 => 集成 => 添加集成 => 集成类型默认 Github 进向 webhook=> 添加集成。复制这个链接



6. 在 github 仓库可见 webhook 已经自动连接成功。



7. 每次更新仓库，都会自动触发构建，时间随内容容量几分钟到几十分钟不等。



8. 到这里已经成功了，我们通过概况下的网址已可以搜索到在线文档了。接下来若想要更改域名，只需要购买域名，并在管理 => 域中设置。