How to use sphinx

version

Hubery-Lee(李会)

二月 14, 2021

目录

Welcome to A Brief Tutorial On Making Beauty Documents!

- 邮箱（email）：hrbeulh@126.com
- 主页（homepage）：http://hubery-lee.github.io

> Hint
>
> PDF 版文档可以点这里查看

什么是sphnix?

Sphinx 是种令人可以轻松撰写出优美文档的工具，由 Georg Brandl 在BSD 许可证下创造，它允许开发人员以纯文本格式编写文档，以便采用满足不同需求的格式轻松生成输出. 这在使用 Version 》> Control System 追踪变更时非常有用. 纯文本文档对不同系统之间的协作者也非常有用. 纯文本是当前可以采用的最便捷的格式之一.

虽然 Sphinx 是用 Python 编写的，并且最初是为 Python 语言文档而创建，但它并不一定是以语言为中心，在某些情况下，甚至不是以程序员为中心. Sphinx 有许多用处，比如可以用它来发布你的项目文档，或编写整本书！

Sphinx官网：http://www.sphinx-doc.org/en/stable/，

为什么选择sphinx?

市面上有gitbook 和sphinx两种技术类数目编辑器，为什么想选择sphinx,而不是gitbook呢？

[具体介绍参见](https://stackshare.io/stackups/gitbook-vs-sphinx)

Gitbook vs Sphinx：有什么区别?

开发人员将Gitbook描述为“为您，您的用户和您的团队记录所有内容！”。这是一个现代化的文档平台，团队可以在其中记录从产品到API以及内部知识库的所有内容。这是一个为您和您的团队思考和跟踪想法的地方。另一方面，Sphinx被详细描述为“开源全文搜索服务器，从头开始设计时就考虑了性能，相关性（即搜索质量）和集成简便性”。通过Sphinx，您可以快速轻松地批处理索引和搜索存储在SQL数据库，NoSQL存储中的数据，或者只是文件，也可以快速地对索引和搜索数据进行索引，使用Sphinx就像在数据库服务器上一样。多种文本处理功能可根据您的特定应用需求对Sphinx进行微调，并且许多相关功能可确保您也可以调整搜索质量。

Gitbook属于技术堆栈的“文档作为服务和工具”类别，而Sphinx可以主要归类于“搜索引擎”下。

根据StackShare社区的说法，Sphinx得到了更广泛的认可，在**38**个公司堆栈和**14**个开发人员堆栈中被提及。与Gitbook相比，后者被列在**15**个公司堆栈和**4**个开发人员堆栈中。

Sphinx的特点

那么如何安装和配置sphinx呢？

- [ ] 支持markdown、reStructuredText语言
- [ ] 特点在写大型技术文档

Hint

reStructuredText语言的特点

擅长于写长篇技术文档，特点是功能更丰富，上手时间相对要长，但可以采用[](https://pandoc.org)进行转换成其他格式（markdown word epud pdf …）

那么如何安装和配置sphinx呢?

效果预览

https://how-to-use-sphinx-write.readthedocs.io/zh_CN/latest/

安装

准备

安装之前假设你已经安装好了python3

sphinx版本:3.4　以下以Linux命令为主,window可以手动创建目录,然后在cmd执行pip和初始化即可.　..
code-block:

```
#■■sphinx/■■/markdown
pip install sphinx recommonmark sphinx_rtd_theme
#■■■■■■docs
mkdir docs
#■■■docs■■
cd docs
#■■■■■sphinx
sphinx-quickstart
```

在执行命令sphinx-quickstart的时候，会让你输入配置，除了以下几个个性化配置外，其他的都可以按照默认的来（回车默认配置）。

```
Separate source and build directories (y/n) [n]:n
Project name: how-to-use-sphinx
Author name(s): jonnyan404
Project release []: 0.1
Project language [en]: zh_CN
```

执行完毕后，就可以看见创建的工程文件

_build：文件夹，当你执行make html的时候，生成的html静态文件都存放在这里
_static：文件夹：图片,js等存放地址
_templates：文件夹:模板文件存放
make.bat：bat脚本
Makefile:编译文件
index.rst:索引文件,文章目录大纲
conf.py:配置文件

那么如何安装和配置sphinx呢？

编写文章

在docs目录下新建hello.rst，内容如下：

```
hello,world
=============
```

如果会markdown语法，无需学习rst语法，可参考文末语法转换网站.

index.rst修改如下：

```
.. toctree::
   :maxdepth: 2

   hello
```

然后在docs目录下执行 make html，进入 _build/html 目录后用浏览器打开 index.html

更改主题和添加md支持

vim conf.py #更改如下配置：

```
html_theme = "sphinx_rtd_theme"
extensions = ['recommonmark']
```

然后再次运行 make html 即可.
关于markdown的用法形式与rst一样，直接更换后缀并在文件内已markdown语法写内容即可.

与GitHub联动

上传代码至GitHub仓库，然后去 https://readthedocs.org/ 注册账号，并关联GitHub仓库.
然后需要在GitHub仓库根目录下，增加一个名称为 .readthedocs.yml 的配置文件：

> ### Caution!
>
> github 已经将默认`master`分支改为`main`，所以需要重新建立master分支并推送到github仓库

> ### Note
>
> git的快速入门可以参考 OSChina这篇入门实例交互应用 git的使用笔记请查阅 Hubery-Lee Git入门笔记

```
# .readthedocs.yml
# Read the Docs configuration file
# See https://docs.readthedocs.io/en/stable/config-file/v2.html for details

# Required
```

如何使用

```
version: 2

# Build documentation in the docs/ directory with Sphinx
sphinx:
  configuration: docs/conf.py

# Build documentation with MkDocs
#mkdocs:
#  configuration: mkdocs.yml

# Optionally build your docs in additional formats such as PDF
formats:
  - pdf

# Optionally set the version of Python and requirements required to build your docs
python:
  version: 3.7
  install:
    - requirements: docs/requirements.txt
```

再去 docs 目录下,新建一个名称为 requirements.txt 的文件,在这个文件内增加你所使用的包名称.
例如我的是:

```
sphinx
sphinx-rtd-theme
recommonmark
```

如果以上两个文件不添加,那么自动构建出来的文章,与你在本地的生成的会不一致,因为 readthedocs 网站默认使用mkdocs来构建.

后记

https://www.cnblogs.com/jonnyan/p/14207711.html

如何使用

该手册采用`reStructuredText <http://docutils.sourceforge.net/rst.html>`_标记语言撰写,使用Sphinx进行发布.

属性	值
开发语言	reStructuredText
发布工具	Sphinx

快速入门

Quick reStructuredText

如何使用

行内markup

Plain text	Typical result	Notes
`*emphasis*`	*emphasis*	Normally rendered as italics.
`**strong emphasis**`	**strong emphasis**	Normally rendered as boldface.
`` `interpreted text` ``	(see note at right)	The rendering and *meaning* of interpreted text is domain- or application-dependent. It can be used for things like index entries or explicit descriptive markup (like program identifiers).
`` ``inline literal`` ```	`inline literal`	Normally rendered as monospaced text. Spaces should be preserved, but line breaks will not be.
`reference_`	reference	A simple, one-word hyperlink reference. See Hyperlink Targets.
`` `phrase reference`_ ``	phrase reference	A hyperlink reference with spaces or punctuation needs to be quoted with backquotes. See Hyperlink Targets.
`anonymous__`	anonymous	With two underscores instead of one, both simple and phrase references may be anonymous (the reference text is not repeated at the target). See Hyperlink Targets.
`_`inline internal target``	inline internal target	A crossreference target within text. See Hyperlink Targets.
`|substitution reference|`	(see note at right)	The result is substituted in from the substitution definition. It could be text, an image, a hyperlink, or a combination of these and others.
`footnote reference [1]_`	footnote reference [1]	See Footnotes.
`citation reference [CIT2002]_`	citation reference [CIT2002]	See Citations.
`http://docutils.sf.net/`	http://docutils.sf.net/	A standalone hyperlink.

结构化

Plain text	Typical result
`=====` `Title` `=====` `Subtitle` `--------` `Titles are underlined (or over-` `and underlined) with a printing` `nonalphanumeric 7-bit ASCII` `character. Recommended choices` `are "` ` ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`".` `The underline/overline must be at` `least as long as the title text.` `A lone top-level (sub)section` `is lifted up to be the document's` `(sub)title.`	**Title** **Subtitle** Titles are underlined (or over- and underlined) with a printing nonalphanumeric 7-bit ASCII character. Recommended choices are "= - ` : ' " ~ ^ _ * + # < >". The underline/overline must be at least as long as the title text. A lone top-level (sub)section is lifted up to be the document's (sub)title.

列表

项目符号列表、数字枚举列表与markdown语法一样

> **Tip**
>
> 域列表和选项列表 markdown语法没有

Field Lists

(details)

Plain text	Typical result
`:Authors:` `Tony J. (Tibs) Ibbs,` `David Goodger` `(and sundry other good-natured folks)` `:Version: 1.0 of 2001/08/08` `:Dedication: To my father.`	**Authors:** Tony J. (Tibs) Ibbs, David Goodger (and sundry other good-natured folks) **Version:** 1.0 of 2001/08/08 **Dedication:** To my father.

Field lists are used as part of an extension syntax, such as options for directives, or database-like records meant for further processing. Field lists may also be used as generic two-column table constructs in documents.

Option Lists

(details)

Plain text	Typical result
`-a command-line option "a"` `-b file options can have arguments` `and long descriptions` `--long options can be long also` `--input=file long options can also have` `arguments` `/V DOS/VMS-style options too`	`-a` command-line option "a" `-b file` options can have arguments and long descriptions `--long` options can be long also `--input=file` long options can also have arguments `/V` DOS/VMS-style options too

如何使用

文字块

Literal Blocks

(details)

Plain text	Typical result
A paragraph containing only two colons indicates that the following indented or quoted text is a literal block. :: Whitespace, newlines, blank lines, and all kinds of markup (like *this* or \this) is preserved by literal blocks. The paragraph containing only '::' will be omitted from the result. The ::: may be tacked onto the very end of any paragraph. The ::: will be omitted if it is preceded by whitespace. The ::: will be converted to a single colon if preceded by text, like this:: It's very convenient to use this form. Literal blocks end when text returns to the preceding paragraph's indentation. This means that something like this is possible:: We start here and continue here and end here. Per-line quoting can also be used on unindented literal blocks:: > Useful for quotes from email and > for Haskell literate programming.	A paragraph containing only two colons indicates that the following indented or quoted text is a literal block. Whitespace, newlines, blank lines, and all kinds of markup (like *this* or \this) is preserved by literal blocks. The paragraph containing only '::' will be omitted from the result. The :: may be tacked onto the very end of any paragraph. The :: will be omitted if it is preceded by whitespace. The :: will be converted to a single colon if preceded by text, like this: It's very convenient to use this form. Literal blocks end when text returns to the preceding paragraph's indentation. This means that something like this is possible: We start here and continue here and end here. Per-line quoting can also be used on unindented literal blocks: > Useful for quotes from email and > for Haskell literate programming.

竖线块

Line Blocks

(details)

Plain text	Typical result
| Line blocks are useful for addresses, | verse, and adornment-free lists. | | Each new line begins with a | vertical bar ("|"). | Line breaks and initial indents | are preserved. | Continuation lines are wrapped portions of long lines; they begin with spaces in place of vertical bars.	Line blocks are useful for addresses, verse, and adornment-free lists. Each new line begins with a vertical bar ("|"). Line breaks and initial indents are preserved. Continuation lines are wrapped portions of long lines; they begin with spaces in place of vertical bars.

超链接目标

外部超链接，内部超链接

External Hyperlink Targets

Plain text	Typical result
External hyperlinks, like Python_. .. _Python: http://www.python.org/	*Fold-in form* External hyperlinks, like Python. *Call-out form* External hyperlinks, like *Python*. *Python:* http://www.python.org/

"*Fold-in*" is the representation typically used in HTML documents (think of the indirect hyperlink being "folded in" like ingredients into a cake), and "*call-out*" is more suitable for printed documents, where the link needs to be presented explicitly, for example as a footnote. You can force usage of the call-out form by using the "target-notes" directive.

reStructuredText also provides for **embedded URIs** (details), a convenience at the expense of readability. A hyperlink reference may directly embed a target URI inline, within angle brackets. The following is exactly equivalent to the example above:

Plain text	Typical result
External hyperlinks, like `Python <http://www.python.org/>`_.	External hyperlinks, like Python.

Internal Hyperlink Targets

Plain text	Typical result
Internal crossreferences, like example_. .. _example: This is an example crossreference target.	*Fold-in form* Internal crossreferences, like example This is an example crossreference target. *Call-out form* Internal crossreferences, like example *example:* This is an example crossreference target.

如何使用

图片的直接使用与替代参考与定义

Directives

(details)

Directives are a general-purpose extension mechanism, a way of adding support for new constructs without adding new syntax. For a description of all standard directives, see reStructuredText Directives.

Plain text	Typical result
For instance: .. image:: images/ball1.gif	For instance:

Substitution References and Definitions

(details)

Substitutions are like inline directives, allowing graphics and arbitrary constructs within text.

Plain text	Typical result
The |biohazard| symbol must be used on containers used to dispose of medical waste. .. |biohazard| image:: biohazard.png	The ☣ symbol must be used on containers used to dispose of medical waste.

参考资料

Citations

(details)

Plain text	Typical result
Citation references, like [CIT2002]_. Note that citations may get rearranged, e.g., to the bottom of the "page". .. [CIT2002] A citation (as often used in journals). Citation labels contain alphanumerics, underlines, hyphens and fullstops. Case is not significant. Given a citation like [this]_, one can also refer to it like this_. .. [this] here.	Citation references, like [CIT2002]. Note that citations may get rearranged, e.g., to the bottom of the "page". Citation labels contain alphanumerics, underlines, hyphens and fullstops. Case is not significant. Given a citation like [this], one can also refer to it like this. --- **[CIT2002]** A citation (as often used in journals). **[this]** here.

表格

Tables

(details)

There are two syntaxes for tables in reStructuredText. Grid tables are complete but cumbersome to create. Simple tables are easy to create but limited (no row spans, etc.).

Plain text	Typical result
Grid table: +------------+------------+-----------+ | Header 1 | Header 2 | Header 3 | +============+============+===========+ | body row 1 | column 2 | column 3 | +------------+------------+-----------+ | body row 2 | Cells may span columns.| +------------+------------+-----------+ | body row 3 | Cells may | - Cells | +------------+ span rows. | - contain | | body row 4 | | - blocks. | +------------+------------+-----------+	Grid table: <table><tr><th>Header 1</th><th>Header 2</th><th>Header 3</th></tr><tr><td>body row 1</td><td>column 2</td><td>column 3</td></tr><tr><td>body row 2</td><td colspan="2">Cells may span columns.</td></tr><tr><td>body row 3</td><td rowspan="2">Cells may span rows.</td><td rowspan="2"> - Cells - contain - blocks. </td></tr><tr><td>body row 4</td></tr></table>
Simple table: ===== ===== ====== Inputs Output ------------ ------ A B A or B ===== ===== ====== False False False True False True False True True True True True ===== ===== ======	Simple table: <table><tr><th colspan="2">Inputs</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>A or B</th></tr><tr><td>False</td><td>False</td><td>False</td></tr><tr><td>True</td><td>False</td><td>True</td></tr><tr><td>False</td><td>True</td><td>True</td></tr><tr><td>True</td><td>True</td><td>True</td></tr></table>

注释

Transitions

(details)

Plain text	Typical result
A transition marker is a horizontal line of 4 or more repeated punctuation characters. ------------ A transition should not begin or end a section or document, nor should two transitions be immediately adjacent.	A transition marker is a horizontal line of 4 or more repeated punctuation characters. --- A transition should not begin or end a section or document, nor should two transitions be immediately adjacent.

如何使用

脚注

Footnotes

(details)

Plain text	Typical result
Footnote references, like [5]_. Note that footnotes may get rearranged, e.g., to the bottom of the "page". .. [5] A numerical footnote. Note there's no colon after the `.]_`.	Footnote references, like [5]. Note that footnotes may get rearranged, e.g., to the bottom of the "page". [5] A numerical footnote. Note there's no colon after the].
Autonumbered footnotes are possible, like using [#]_ and [#]_. .. [#] This is the first one. .. [#] This is the second one. They may be assigned 'autonumber labels' - for instance, [#fourth]_ and [#third]_. .. [#third] a.k.a. third_ .. [#fourth] a.k.a. fourth_	Autonumbered footnotes are possible, like using [1] and [2]. They may be assigned 'autonumber labels' - for instance, [4] and [3]. [1] This is the first one. [2] This is the second one. [3] a.k.a. third [4] a.k.a. fourth
Auto-symbol footnotes are also possible, like this: [*]_ and [*]_. .. [*] This is the first one. .. [*] This is the second one.	Auto-symbol footnotes are also possible, like this: [*] and [†]. [*] This is the first symbol footnote [†] This is the second one.

评论

Comments

(details)

Any text which begins with an explicit markup start but doesn't use the syntax of any of the constructs above, is a comment.

Plain text	Typical result
.. This text will not be shown (but, for instance, in HTML might be rendered as an HTML comment)	
An "empty comment" does not consume following blocks. (An empty comment is ".." with blank lines before and after.) .. So this block is not "lost", despite its indentation.	An "empty comment" does not consume following blocks. (An empty comment is ".." with blank lines before and after.) So this block is not "lost", despite its indentation.

代码块

```
.. math::
α_t(i) = P(O_1, O_2, … O_t, q_t = S_i λ)
```

测试代码

```
>>>print( 'this is my test' )
```

数学公式

采用latex风格的数学输入

```
.. math::
α_t(i) = P(O_1, O_2, … O_t, q_t = S_i λ)
```

$$\alpha_t(i) = P(O_1, O_2, ...O_t, q_t = S_i\lambda)$$

一些特殊块的使用

caution、danger、error、hint、important、note、tip、warning、admonition

详细入门资料请查看 restructuredtext

> ### Caution!
>
> 详细入门资料请查看 restructuredtext

!DANGER!

详细入门资料请查看 restructuredtext

Error

详细入门资料请查看 restructuredtext

Hint

详细入门资料请查看 restructuredtext

Important

详细入门资料请查看 restructuredtext

Note

详细入门资料请查看 restructuredtext

Tip

详细入门资料请查看 restructuredtext

Warning

详细入门资料请查看 restructuredtext

如何部署

采用github 登入readthedoc，授权后，导入托管参考即可

如何转换成pdf

用sphinx写好的文档可以转成pdf格式，以前不是说自己要写一本书么，现在就可以实现了，而且都是可以按照自己的风格来排版，到时候自己出钱就可以出书了，是不是感觉特别棒。

pip nstall rst2pdf

Add rst2pdf to the list of extensions in conf.py extensions = [‘rst2pdf.pdfbuilder’]

This list will be empty if you accepted the defaults when the project was setup. If not, just append ‘rst2pdf.pdfbuilder’ to the list.

Add a pdf_documents variable to conf.py

```
pdf_documents = [('index', u'rst2pdf', u'Sample rst2pdf doc', u'Your Name'),]

# rst2pdf - name of the generated pdf
# Sample rst2pdf doc - title of the pdf
# Your Name - author name in the pdf
```

Generate pdf .. code-block:

```
make pdf
```

The generated pdf will be in the build/pdf directory.

https://www.dazhuanlan.com/2019/10/06/5d99f684a0c43/

中文乱码解决办法

https://www.tutorialfor.com/blog-222028.htm

Indices and tables

- ☐ genindex
- ☐ modindex
- ☐ search