

Create automatic Python Code Documentation with Sphinx

Catalina Maria Lauer Zegarra
SAFE Research Data Center*

Last Update: December 8, 2023

Sphinx automatic documentation is a process where documentation is automatically extracted from source code comments, utilizing reStructuredText for its plaintext markup syntax. This allows for the generation of various formats, such as HTML and PDF, from the same source files. The system supports syntax highlighting and cross-referencing, enhancing the readability and navigability of the documentation. Additionally, Sphinx offers extensive customization options through themes and plugins, making it adaptable for different documentation needs.

Contents

1	Contact	2
2	Example of Coding Syntax	2
3	How to Compile	2
3.1	Install Sphinx and Related Packages:	2
3.2	Create Folders:	3
3.3	Navigate to the Docs Folder:	3
3.4	Initialize Sphinx Project (CLI):	3
3.5	Configure Sphinx Setup (CLI):	3
3.6	Modify conf.py for Project Settings (Text Editor):	3
3.7	Edit the index.rst File (Text Editor):	4
3.8	Generate Module Documentation (CLI):	4
3.9	Create HTML Documentation (CLI):	4
3.10	Create PDF Documentation (CLI):	4
3.11	Accessing Generated Documentation:	4

*lauer@safe-frankfurt.de

1 Contact

If you encounter any difficulties or just want general information, do not hesitate to contact us.

SAFE Research Data Center: datacenter@safe-frankfurt.de

More information about the SAFE Research Data Center, and further guides can be found [here](#), and [here](#).

2 Example of Coding Syntax

Please find an example of the coding syntax [here](#):

```
1 def download_file(  
2     self,  
3     item_path: str,  
4     max_connections: int = 8,  
5     dest_dir: str | Path | None = None,  
6     verbose: bool = False,  
7 ) -> str:  
8     """Downloads a file by file system path to the current working directory  
9     asynchronously. More details find at :meth:'safe_datacenter.onedrive.  
10    connector.Connector.download_file_by_id'.11  
12    :param item_path: item path from root of the file to be downloaded  
13    :type item_path: str  
14    :param max_connections: max concurrent open http requests, refer Docs  
15    regarding throttling limits, defaults to 8  
16    :type max_connections: int, optional  
17    :param dest_dir: destination directory for the downloaded file, defaults  
18    to None  
19    :type dest_dir: str | Path | None, optional  
20    :param verbose: prints status message during the download process,  
    defaults to False  
    :type verbose: bool, optional  
    :return: returns the path of the file including extension  
    :rtype: str  
    """
```

3 How to Compile

3.1 Install Sphinx and Related Packages:

Open your computer's command line or terminal (CLI). Install **Sphinx** for documentation generation:

```
1 pip install Sphinx
```

Install **rst2pdf** for PDF output:

```
1 pip install rst2pdf
```

Optionally, for a different theme:

```
1 pip install sphinx-rtd-theme
```

(Ensure that you have Python and pip installed on your system for these commands to work.)

3.2 Create Folders:

Manually create a new folder structure in your project directory:

- A docs folder for Sphinx documentation.
- A package folder for your Python code.

Inside package, include your Python scripts, such as `test_docstring.py`.

3.3 Navigate to the Docs Folder:

In your command line or terminal, navigate to the docs folder using the `cd` command

```
1 local_path\docs>
```

3.4 Initialize Sphinx Project (CLI):

Within the docs folder in your CLI, run:

```
1 local_path\docs> sphinx-quickstart
```

This command will prompt you to answer setup questions to configure your Sphinx project.

3.5 Configure Sphinx Setup (CLI):

- Answer the `sphinx-quickstart` setup questions in the CLI.
- Choose `y` for separate source and build directories.
- Provide your project name, author name, project release, and language.

```
1 Separate source and build directories (y/n) [n]: y
2 Project name: test package
3 Author name(s): god
4 Project release []: 1.0
5 Project language [en]: y
```

3.6 Modify `conf.py` for Project Settings (Text Editor):

Open `docs/source/conf.py` in a text editor.

Add at the beginning of the file:

```
1 import os
2 import sys
3 sys.path.insert(0, os.path.abspath(os.path.join('.', '..', 'package')))
```

Then, at the end of the file

```
1 extensions = ['sphinx.ext.autodoc', 'sphinx.ext.napoleon', 'rst2pdf.pdfbuilder',
2              '']
3 pdf_documents = [('index', u'docu', u'Sample doc', u'god'),]
4 pdf_stylesheets = ['twocolumn', 'sphinx']
```

3.7 Edit the index.rst File (Text Editor):

Open docs/source/index.rst in a text editor and modify this file with the following code:

```
1 .. test package documentation master file, created by sphinx-quickstart on
   Wed May 17 16:25:19 2023. You can adapt this file completely to your
   liking, but it should at least contain the root 'toctree' directive.
2
3 Welcome to test package's documentation!
4 =====
5 .. toctree::
6    :maxdepth: 2
7    :caption: Contents:
8
9 modules
10
11 Indices and tables
12 =====
13
14 * :ref:'genindex'
15 * :ref:'modindex'
16 * :ref:'search'
```

3.8 Generate Module Documentation (CLI):

Back in the CLI, within the docs folder, run

```
1 sphinx-apidoc -o ./source ../package
```

to create documentation source files from your Python modules.

3.9 Create HTML Documentation (CLI):

In the CLI, execute

```
1 sphinx-build -b html source/ build/html
```

to generate the HTML documentation.

3.10 Create PDF Documentation (CLI):

Run

```
1 sphinx-build -b pdf source build/pdf
```

in the CLI to generate PDF documentation.

3.11 Accessing Generated Documentation:

Find the HTML documentation in docs/build/html and open index.html in a web browser.
Find the PDF documentation in docs/build/pdf.

Note: For any issues, refer to the Sphinx [documentation](#) and ensure all file paths and commands are correct.