
sphinx-github-style

Release 1.2.1

Adam Korn

Apr 07, 2024

README

1	About	3
1.1	GitHub Source Code Links	3
1.2	Syntax Highlighting	4
2	Installation	5
3	Configuration	7
3.1	Optional Configuration Variables	7
3.1.1	linkcode_blob	7
3.1.2	linkcode_url	7
3.1.3	linkcode_link_text	8
3.1.4	linkcode_resolve	8
4	The sphinx_github_style Package	9
4.1	CSS Styling for Linkcode Links	9
4.1.1	The github_style.css StyleSheet	9
4.2	GitHub Dark Theme Pygments Style	10
4.3	Pygments Lexer for Syntax Highlighting of Methods, Classes, Type Hints, and more	11
4.4	The sphinx_github_style.utils subpackage	12
4.4.1	The sphinx_github_style.utils.git submodule	12
4.4.2	The sphinx_github_style.utils.linkcode submodule	12
4.4.3	The sphinx_github_style.utils.sphinx submodule	13
5	Indices and tables	15
	Python Module Index	17
	Index	19

GitHub source code links and syntax highlighting for Sphinx documentation

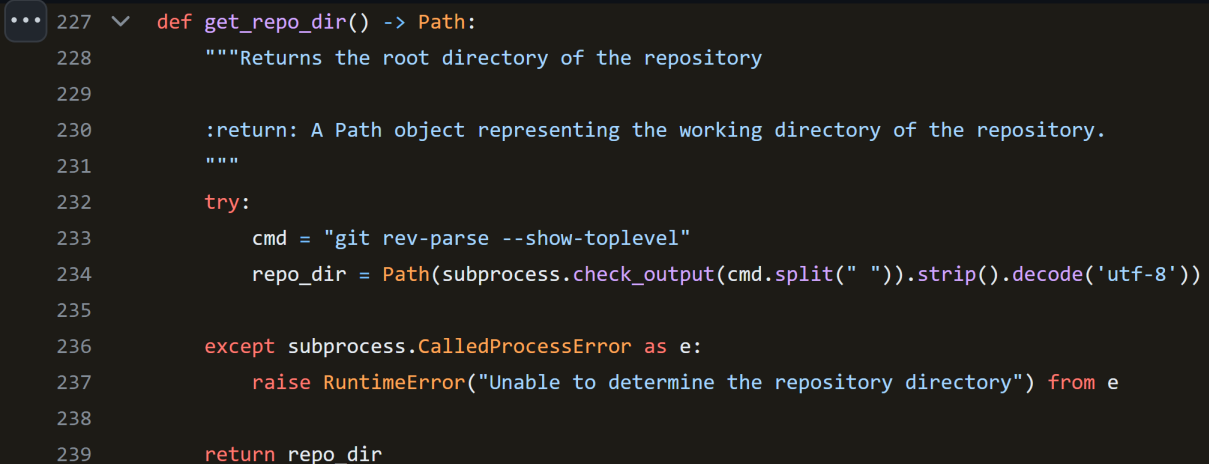
sphinx-github-style is a Sphinx extension that links your documentation to GitHub source code. It also adds syntax highlighting for code blocks similar to GitHub's pretty lights dark theme.

...

1.1 GitHub Source Code Links

Using `sphinx.ext.linkcode`, a View on GitHub link is added to the documentation of every class, method, function, and property:

They link to and highlight the corresponding code block in your GitHub repository:



```
227  def get_repo_dir() -> Path:
228      """Returns the root directory of the repository
229
230      :return: A Path object representing the working directory of the repository.
231      """
232      try:
233          cmd = "git rev-parse --show-toplevel"
234          repo_dir = Path(subprocess.check_output(cmd.split(" ")).strip().decode('utf-8'))
235
236      except subprocess.CalledProcessError as e:
237          raise RuntimeError("Unable to determine the repository directory") from e
238
239      return repo_dir
```

Note: These links can be *styled with CSS* and used with/instead of the links added by `sphinx.ext.viewcode`

1.2 Syntax Highlighting

sphinx-github-style also contains a Pygments style to highlight code blocks similar to GitHub:

INSTALLATION

To install using pip:

```
pip install sphinx-github-style
```


CONFIGURATION

Add the extension to your `conf.py`

```
extensions = [  
    "sphinx_github_style",  
]
```

3.1 Optional Configuration Variables

Add any (or none) of the following configuration variables to your `conf.py`

3.1.1 `linkcode_blob`

`linkcode_blob`

The blob to link to on GitHub - any of "head", "last_tag", or "{blob}"

- head (default): links to the most recent commit hash; if this commit is tagged, uses the tag instead
- last_tag: links to the most recent commit tag on the currently checked out branch
- blob: links to any blob you want, for example "master" or "v2.0.1"

Type

str

Default

"head"

3.1.2 `linkcode_url`

`linkcode_url`

The link to your GitHub repository formatted as `https://github.com/user/repo`

- If not provided, will attempt to create the link from the `html_context` dict

Type

str

Default

```
f"https://github.com/{html_context['github_user']}/  
{html_context['github_repo']}/{html_context['github_version']}"
```

3.1.3 linkcode_link_text

linkcode_link_text

The text to use for the linkcode link

Type

str

Default

"View on GitHub"

3.1.4 linkcode_resolve

linkcode_resolve

A `linkcode_resolve()` function to use when resolving the link target with `sphinx.ext.linkcode`

Type

Callable

Default

Return value from `get_linkcode_resolve()`

THE SPHINX_GITHUB_STYLE PACKAGE

`sphinx_github_style.__init__.setup(app)`

Return type

Dict[str, Any]

4.1 CSS Styling for Linkcode Links

`sphinx_github_style.add_linkcode_class.add_linkcode_node_class(app, doctree, docname)`

Changes every Node added by `sphinx.ext.linkcode` to use the "linkcode-link" class

This creates separation from the nodes added by `sphinx.ext.viewcode`, allowing for different link text and CSS styling

Sets the link text to `linkcode_link_text`, or "View on GitHub" if not provided

4.1.1 The `github_style.css` StyleSheet

Links are styled as follows:

```
/* Change custom linkcode ("View on GitHub") to match viewcode ("[source]") */
.linkcode-link {
    color: #00e;
    font-size: 80%;
    margin-left: 10px;
}
/* Add a GitHub Icon to Linkcode Links */
.linkcode-link::after {
    content: url('data:image/svg+xml;charset=utf-8,<svg xmlns="http://www.w3.org/2000/svg"
↵width="16" height="12" viewBox="0 0 24 22"><path d="M12 0c-6.626 0-12 5.373-12 12 0 5.302
↵3.438 9.8 8.207 11.387.599.111.793-.261.793-.577v-2.234c-3.338.726-4.033-1.416-4.033-1.416-
↵.546-1.387-1.333-1.756-1.333-1.756-1.089-.745.083-.729 1.205.084 1.839 1.237 1.
↵839 1.237 1.07 1.834 2.807 1.304 3.492.997.107-.775.418-1.305.762-1.604-2.665-.305-5.467-1.
↵334-5.467-5.931 0-1.311.469-2.381 1.236-3.221-.124-.303-.535-1.524.117-3.176 0 0 1.008-.
↵322 3.301 1.23.957-.266 1.983-.399 3.003-.404 1.02.005 2.047.138 3.006.404 2.291-1.552 3.
```

(continues on next page)

(continued from previous page)

```
↪297-1.23 3.297-1.23.653 1.653.242 2.874.118 3.176.77.84 1.235 1.911 1.235 3.221 0 4.609-2.  
↪807 5.624-5.479 5.921.43.372.823 1.102.823 2.222v3.293c0 .319.192.694.801.576 4.765-1.589_  
↪8.199-6.086 8.199-11.386 0-6.627-5.373-12-12-12z"/></svg>');  
    margin-left: 4px;  
}
```

To make the viewcode-link links match with the linkcode-link links, add the following to your custom.css file:

```
/* Change viewcode "[source]" link position and colour to match linkcode */  
.rst-content .viewcode-back, .rst-content .viewcode-link {  
    color: #00e;  
    padding-left: 12px;  
}
```

4.2 GitHub Dark Theme Pygments Style

class sphinx_github_style.github_style.GitHubStyle

A Pygments style similar to GitHub's pretty lights dark theme

background_color = '#0d1117'

overall background color (None means transparent)

default_style = ''

```

styles = {Token: '', Token.Comment: '#8b949e', Token.Comment.Hashbang: '#8b949e',
Token.Comment.Multiline: '#8b949e', Token.Comment.Preproc: '#8b949e',
Token.Comment.PreprocFile: '', Token.Comment.Single: '#8b949e', Token.Comment.Special:
'#8b949e', Token.Error: '#f8f8f2', Token.Escape: '', Token.Generic: '#f0f6fc',
Token.Generic.Deleted: '#8b080b', Token.Generic.Emph: '#f8f8f2 underline',
Token.Generic.EmphStrong: '', Token.Generic.Error: '#f8f8f2', Token.Generic.Heading:
'#f8f8f2 bold', Token.Generic.Inserted: '#f8f8f2 bold', Token.Generic.Output: '#adaeb6',
Token.Generic.Prompt: '#f8f8f2', Token.Generic.Strong: '#f8f8f2',
Token.Generic.Subheading: '#f8f8f2 bold', Token.Generic.Traceback: '#f8f8f2',
Token.Keyword: '#ff7b72', Token.Keyword.Constant: '#79c0ff', Token.Keyword.Declaration:
'#ff7b72', Token.Keyword.Namespace: '#ff7b72', Token.Keyword.Pseudo: '#d2a8ff',
Token.Keyword.Reserved: '#79c0ff', Token.Keyword.Type: '#79c0ff', Token.Literal:
'#f8f8f2', Token.Literal.Date: '#f8f8f2', Token.Literal.Number: '#79c0ff',
Token.Literal.Number.Bin: '#79c0ff', Token.Literal.Number.Float: '#79c0ff',
Token.Literal.Number.Hex: '#79c0ff', Token.Literal.Number.Integer: '#79c0ff',
Token.Literal.Number.Integer.Long: '#79c0ff', Token.Literal.Number.Oct: '#79c0ff',
Token.Literal.String: '#a5d6ff', Token.Literal.String.Affix: '#a5d6ff',
Token.Literal.String.Backtick: '#a5d6ff', Token.Literal.String.Char: '#a5d6ff',
Token.Literal.String.Comment: '#8b949e', Token.Literal.String.Delimiter: '',
Token.Literal.String.Doc: '#a5d6ff', Token.Literal.String.Double: '#a5d6ff',
Token.Literal.String.Escape: '#a5d6ff', Token.Literal.String.Heredoc: '#a5d6ff',
Token.Literal.String.Interpol: '#a5d6ff', Token.Literal.String.Other: '#a5d6ff',
Token.Literal.String.Regex: '#a5d6ff', Token.Literal.String.Single: '#a5d6ff',
Token.Literal.String.Symbol: '#a5d6ff', Token.Name: '#e6edf3', Token.Name.Attribute:
'#e6edf3', Token.Name.Builtin: '#d2a8ff', Token.Name.Builtin.Pseudo: '#e6edf3',
Token.Name.Class: '#ffa657', Token.Name.Constant: '#79c0ff', Token.Name.Decorator:
'#d2a8ff', Token.Name.Entity: '#d2a8ff', Token.Name.Exception: '#ffa657',
Token.Name.Function: '#d2a8ff', Token.Name.Function.Magic: '#d2a8ff', Token.Name.Label:
'', Token.Name.Namespace: '#e6edf3', Token.Name.Other: '#e6edf3', Token.Name.Property:
'', Token.Name.Tag: '', Token.Name.Variable: '#e6edf3', Token.Name.Variable.Class:
'#ffa657', Token.Name.Variable.Global: '#ffa657', Token.Name.Variable.Instance:
'#e6edf3', Token.Name.Variable.Magic: '#e6edf3', Token.Operator: '#79c0ff',
Token.Operator.Word: '#79c0ff', Token.Other: '', Token.Other.Constant: '#79c0ff',
Token.Punctuation: '#f8f8f2', Token.Punctuation.Definition.Comment: '#8b949e',
Token.Punctuation.Marker: '', Token.Text: '#e6edf3', Token.Text.Whitespace: '#f0f6fc'}

```

Style definitions for individual token types.

4.3 Pygments Lexer for Syntax Highlighting of Methods, Classes, Type Hints, and more

`sphinx_github_style.lexer.get_builtins()`

Returns a dictionary containing names of built-in functions, classes, and methods

Return type

Dict[str, Set]

class `sphinx_github_style.lexer.GitHubLexer(*args, **kwargs)`

A Pygments Lexer that adds syntax highlighting for the methods, classes, type hints, etc. in a Python package

name = 'TDK'

Full name of the lexer, in human-readable form

`url = 'https://github.com/TDKorn'`

URL of the language specification/definition. Used in the Pygments documentation.

`aliases = ['tdk']`

A list of short, unique identifiers that can be used to look up the lexer from a list, e.g., using `get_lexer_by_name()`.

`get_tokens_unprocessed(text)`

Override to add better syntax highlighting

4.4 The `sphinx_github_style.utils` subpackage

The `sphinx_github_style.utils` subpackage contains a variety of helper functions

4.4.1 The `sphinx_github_style.utils.git` submodule

`sphinx_github_style.utils.git.get_head()`

Gets the most recent commit hash or tag

Returns

The SHA or tag name of the most recent commit, or “master” if the call to git fails.

Return type

str

`sphinx_github_style.utils.git.get_last_tag()`

Get the most recent commit tag on the currently checked out branch

Raises

ExtensionError – if no tags exist on the branch

Return type

str

`sphinx_github_style.utils.git.get_repo_dir()`

Returns the root directory of the repository

Returns

A Path object representing the working directory of the repository.

Return type

Path

4.4.2 The `sphinx_github_style.utils.linkcode` submodule

`sphinx_github_style.utils.linkcode.get_linkcode_revision(blob)`

Get the blob to link to on GitHub

Note: The value of `blob` can be any of “head”, “last_tag”, or “{blob}”

- head (default): links to the most recent commit hash; if this commit is tagged, uses the tag instead
- last_tag: links to the most recent commit tag on the currently checked out branch
- blob: links to any blob you want, for example “master” or “v2.0.1”

Return type*str*

`sphinx_github_style.utils.linkcode.get_linkcode_url(blob=None, context=None, url=None)`

Get the template URL for linking to highlighted GitHub source code with `sphinx.ext.linkcode`

Formatted into the final link by a `linkcode_resolve()` function

Parameters

- **blob** (*Optional[str]*) – The Git blob to link to
- **context** (*Optional[Dict]*) – The `html_context` dictionary
- **url** (*Optional[str]*) – The base URL of the repository (ex. `https://github.com/TDKorn/sphinx-github-style`)

Return type*str*

`sphinx_github_style.utils.linkcode.get_linkcode_resolve(linkcode_url, repo_dir=None)`

Defines and returns a `linkcode_resolve` function for your package

Used by default if `linkcode_resolve` isn't defined in `conf.py`

Parameters

- **linkcode_url** (*str*) – The template URL to use when resolving cross-references with `sphinx.ext.linkcode`
- **repo_dir** (*Optional[Path]*) – The root directory of the Git repository.

Return type*Callable*

4.4.3 The `sphinx_github_style.utils.sphinx` submodule

`sphinx_github_style.utils.sphinx.get_conf_val(app, attr, default=None)`

Retrieve the value of a `conf.py` config variable

Parameters

- **attr** (*str*) – the config variable to retrieve
- **default** (*Optional[Any]*) – the default value to return if the variable isn't found

Return type*Any*

`sphinx_github_style.utils.sphinx.set_conf_val(app, attr, value)`

Set the value of a `conf.py` config variable

Parameters

- **attr** (*str*) – the config variable to set
- **value** (*Any*) – the variable value

INDICES AND TABLES

- `genindex`
- `modindex`
- *Full Table of Contents*

PYTHON MODULE INDEX

S

- `sphinx_github_style.__init__`, [9](#)
- `sphinx_github_style.add_linkcode_class`, [9](#)
- `sphinx_github_style.github_style`, [10](#)
- `sphinx_github_style.lexer`, [11](#)
- `sphinx_github_style.utils`, [12](#)
- `sphinx_github_style.utils.git`, [12](#)
- `sphinx_github_style.utils.linkcode`, [12](#)
- `sphinx_github_style.utils.sphinx`, [13](#)

A

`add_linkcode_node_class()` (in module `sphinx_github_style.add_linkcode_class`), 9
`aliases` (`sphinx_github_style.lexer.GitHubLexer` attribute), 12

B

`background_color` (`sphinx_github_style.github_style.GitHubStyle` attribute), 10

C

configuration value
`linkcode_blob`, 7
`linkcode_link_text`, 8
`linkcode_resolve`, 8
`linkcode_url`, 7

D

`default_style` (`sphinx_github_style.github_style.GitHubStyle` attribute), 10

G

`get_builtins()` (in module `sphinx_github_style.lexer`), 11
`get_conf_val()` (in module `sphinx_github_style.utils.sphinx`), 13
`get_head()` (in module `sphinx_github_style.utils.git`), 12
`get_last_tag()` (in module `sphinx_github_style.utils.git`), 12
`get_linkcode_resolve()` (in module `sphinx_github_style.utils.linkcode`), 13
`get_linkcode_revision()` (in module `sphinx_github_style.utils.linkcode`), 12
`get_linkcode_url()` (in module `sphinx_github_style.utils.linkcode`), 13
`get_repo_dir()` (in module `sphinx_github_style.utils.git`), 12
`get_tokens_unprocessed()` (`sphinx_github_style.lexer.GitHubLexer` method), 12

`GitHubLexer` (class in `sphinx_github_style.lexer`), 11
`GitHubStyle` (class in `sphinx_github_style.github_style`), 10

L

`linkcode_blob`
configuration value, 7
`linkcode_link_text`
configuration value, 8
`linkcode_resolve`
configuration value, 8
`linkcode_url`
configuration value, 7

M

module
`sphinx_github_style.__init__`, 9
`sphinx_github_style.add_linkcode_class`, 9
`sphinx_github_style.github_style`, 10
`sphinx_github_style.lexer`, 11
`sphinx_github_style.utils`, 12
`sphinx_github_style.utils.git`, 12
`sphinx_github_style.utils.linkcode`, 12
`sphinx_github_style.utils.sphinx`, 13

N

`name` (`sphinx_github_style.lexer.GitHubLexer` attribute), 11

S

`set_conf_val()` (in module `sphinx_github_style.utils.sphinx`), 13
`setup()` (in module `sphinx_github_style.__init__`), 9
`sphinx_github_style.__init__`
module, 9
`sphinx_github_style.add_linkcode_class`
module, 9
`sphinx_github_style.github_style`
module, 10
`sphinx_github_style.lexer`
module, 11
`sphinx_github_style.utils`

- module, [12](#)
- sphinx_github_style.utils.git
 - module, [12](#)
- sphinx_github_style.utils.linkcode
 - module, [12](#)
- sphinx_github_style.utils.sphinx
 - module, [13](#)
- styles (*sphinx_github_style.github_style.GitHubStyle*
attribute), [10](#)

U

- url (*sphinx_github_style.lexer.GitHubLexer* attribute),
[11](#)