
sphinx-readme

Release 1.2.1

Adam Korn

Mar 30, 2024

README

1	Sphinx README - Generate Beautiful README.rst for GitHub, PyPi, GitLab, BitBucket	1
1.1	About Sphinx README	1
1.2	Features	2
1.3	Installation	2
1.4	Configuration	2
2	Extension Configuration	3
2.1	Mandatory conf.py values	3
2.1.1	html_context	3
2.1.2	html_baseurl	3
2.1.3	readme_src_files	4
2.1.4	readme_docs_url_type	4
2.2	Optional conf.py values	4
2.2.1	readme_out_dir	4
2.2.2	readme_replace_attrs	5
2.2.3	readme_inline_markup	5
2.2.4	readme_raw_directive	5
2.2.5	readme_tags	6
2.2.6	readme_include_directive	6
2.2.7	readme_admonition_icons	6
2.2.8	readme_default_admonition_icon	7
2.2.9	readme_rubric_heading	7
2.2.10	readme_blob	8
2.2.11	linkcode_resolve	8
3	Sample conf.py	9
4	The sphinx_readme package	11
4.1	The READMEParser Class	11
4.2	The READMEConfig Class	17
4.3	The sphinx_readme.utils subpackage	19
4.3.1	The sphinx_readme.utils.git submodule	20
4.3.2	The sphinx_readme.utils.linkcode submodule	22
4.3.3	The sphinx_readme.utils.rst submodule	23
4.3.4	The sphinx_readme.utils.sphinx submodule	26
4.3.5	The sphinx_readme.utils.docutils submodule	27
5	Indices and tables	29
	Python Module Index	31

SPHINX README - GENERATE BEAUTIFUL README.RST FOR GITHUB, PYPI, GITLAB, BITBUCKET

A Sphinx extension to generate `README.rst` files that render beautifully on GitHub, PyPi, GitLab, BitBucket

1.1 About Sphinx README

What's Sphinx README?

`sphinx_readme` is a `reStructuredText` parser that uses Sphinx to generate `rst` files that render beautifully on GitHub, PyPi, GitLab, and BitBucket.

With `sphinx_readme`, there's no need to rewrite your `README.rst` as a `README.md` file

Files generated by `sphinx_readme` have nearly identical appearance and functionality as `html` builds, including `sphinx.ext.autodoc` cross-references!

1.2 Features

sphinx_readme adds support for the following sphinx and docutils directives and features:

- `sphinx.ext.autodoc` cross-references (`:mod:`, `:class:`, `:meth:`, `:func:`, and `:attr:`)
- Standard cross-reference roles (`:doc:` and `:ref:`)
- Generic and Specific Admonitions
- Only directives
- Toctrees
- Rubrics
- Images

1.3 Installation

Install using pip:

```
pip install sphinx-readme
```

Add the extension to your `conf.py`:

```
extensions = [  
    'sphinx_readme',  
]
```

1.4 Configuration

Please see *Extension Configuration*

EXTENSION CONFIGURATION

Install using pip:

```
pip install sphinx-readme
```

Add the extension to your `conf.py`:

```
extensions = [  
    "sphinx_readme",  
]
```

For an example of the minimum required configuration, please see the [sample `conf.py` file](#)

2.1 Mandatory `conf.py` values

2.1.1 `html_context`

`html_context`

A dictionary containing info about your repository (`html_context`)

- At minimum, the username and repository name must be specified
- Please see [HTML Context Settings](#) to determine the correct dictionary keys for your hosting platform

Type

Dict

2.1.2 `html_baseurl`

`html_baseurl`

The base URL which points to the root of the HTML documentation (`html_baseurl`)

Type

str

2.1.3 `readme_src_files`

`readme_src_files`

An individual/list of `rst` source files to parse, or a dictionary of source files mapped to output files

Important: Filepaths should be specified relative to the source directory and `output directory`

Type

`Union[str, List]`

2.1.4 `readme_docs_url_type`

`readme_docs_url_type`

The documentation source to link to when resolving `autodoc` cross-references

Must be either `"code"` or `"html"`

- `"code"`: uses `sphinx.ext.linkcode` to replace references with links to highlighted source code

Example

```
parse_intersphinx_nodes()
```

- `"html"`: replaces references with links to HTML documentation entries

Example

```
parse_intersphinx_nodes()
```

Note: If set to `code`, non-external `:attr:` cross-references will not be replaced with links

- Instead, they'll be replaced with `inline literals` or left as is
 - Please see `readme_replace_attrs` and `readme_inline_markup`
-

2.2 Optional `conf.py` values

2.2.1 `readme_out_dir`

`readme_out_dir`

Specifies the path of the directory to save generated `rst` files to

The value should be provided as either

1. An absolute path; or
2. A path relative to the source directory

Type

`Union[str, Path]`

Default

the root directory of your repository, via `get_repo_dir()`

2.2.2 `readme_replace_attrs`

`readme_replace_attrs`

Specifies if non-external attribute (`:attr:`) cross-references should be replaced

...

If True, the value of `readme_docs_url_type` determines how replacements are made:

- "html": references are replaced with a link to the corresponding documentation entry

Example

`:attr:`icon_map`` would be replaced with `icon_map` or `icon_map`, depending on the value of `readme_inline_markup`

- "code": attributes are replaced with inline markup

Example

`:attr:`icon_map`` would be replaced with `icon_map`

Note: External attribute cross-references are always replaced with links to documentation

Type

bool

Default

True

2.2.3 `readme_inline_markup`

`readme_inline_markup`

Specifies if replacements for cross-references should use inline markup

- If True, the substitution for `:attr:`icon_map`` would be `icon_map`
- If False, the substitution for `:attr:`icon_map`` would be `icon_map`

Type

bool

Default

True

2.2.4 `readme_raw_directive`

`readme_raw_directive`

Specifies if the raw directive is supported by the platform you intend to render the generated file on

Tip:

- GitHub supports the raw directive
- PyPi, GitLab, and BitBucket do not support the raw directive

If set to False,

- Admonitions will be replaced with the `list-table` directive instead of HTML tables
- All raw directives in the file will be removed

Type*bool***Default**

True

2.2.5 `readme_tags`

`readme_tags`

Specifies `tags` to use when evaluating the `<expression>` argument of `only` directives

Tip: See `replace_only_directives()` for more detail

Type*List[str]***Default**

["readme"]

2.2.6 `readme_include_directive`

`readme_include_directive`

Specifies if include directives should be parsed (file content inserted) or removed

Type*bool***Default**

True

2.2.7 `readme_admonition_icons`

`readme_admonition_icons`

An optional mapping of admonition classes and their icons

- If specified, will be used to update the `icon_map` (below)

Type*Optional[Dict[str, str]]***Default**

```
{
  'attention': '',
  'caution': '',
  'danger': '',
  'error': '',
  'hint': '',
  'important': '',
  'note': '',
  'tip': '',
  'warning': '',
  'default': ''
}
```

2.2.8 readme_default_admonition_icon

readme_default_admonition_icon

The icon to use when a generic admonition either

- Has no class
- Uses a class that has no corresponding icon in the `icon_map`

Type

str

Default

""

2.2.9 readme_rubric_heading

readme_rubric_heading

The character to use when replacing `rubric` directives with section headers

Must be one of the following valid section title adornment characters:

```
! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~
```

If not specified, rubrics will be replaced with bold text instead of a heading

Type

str

Default

bold inline markup (`**title**`)

2.2.10 `readme_blob`

`readme_blob`

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

- "head": links to the most recent commit hash; if this commit is tagged, uses the tag instead
- "last_tag": links to the most recently tagged commit; if no tags exist, uses "head"
- "{blob}": links to the specified blob, for example "master" or "v2.0.1"

Type

str

Default

"head"

2.2.11 `linkcode_resolve`

`linkcode_resolve`

A `linkcode_resolve()` function for `sphinx.ext.linkcode` to use when resolving the target of `autodoc` cross-references

Type

Callable

Default

return value of `get_linkcode_resolve()`

SAMPLE CONF.PY

```
extensions = [  
    "sphinx_readme",  
]  
  
html_context = {  
    'display_github': True,  
    'github_user': 'TDKorn',  
    'github_repo': 'sphinx-readme',  
}  
  
html_baseurl = "https://sphinx-readme.readthedocs.io/en/latest"  
  
readme_src_files = "README.rst"  
  
readme_docs_url_type = "code"
```

Important: For platforms that don't support the raw directive (PyPi, GitLab, and BitBucket), be sure to disable `readme_raw_directive`:

```
readme_raw_directive = False
```

THE SPHINX_README PACKAGE

4.1 The READMEParser Class

```
class sphinx_readme.parser.READMEParser(app)
    Bases: object

    __init__(app)

    config: READMEConfig
        The READMEConfig for the parser
    ref_map: Dict[str, Union[List, Dict]]
        Mapping of info for standard and sphinx.ext.autodoc cross-references
    sources: Dict[str, str]
        Mapping of source files to their content
    toctrees: Dict[str, List[Dict]]
        Mapping of source files to their toctree data
    admonitions: Dict[str, List[Dict]]
        Mapping of source files to their admonition data
    rubrics: Dict[str, List[str]]
        Mapping of source files to their rubric data
    substitutions: Dict[str, Dict[str, List[str]]]
        Mapping of source files to cross-reference substitution definitions
    titles: Dict[str, str]
        Mapping of docnames to their parsed titles
    domains: Tuple
        Tuple of currently supported Sphinx domains
    roles: Dict[str, List[str]]
        Mapping of domain names to their cross-reference roles
    objtypes: Dict[str, List[str]]
        Mapping of role names to the object types they can cross-reference
```

intersphinx_pkgs: `List[str]`

Packages in the intersphinx cache

inventory: `Dict[str, Dict]`

Easy access to intersphinx inventory

named_inventory: `Dict[str, Dict]`

Easy access to intersphinx named inventory

parse_env(*env*)

Parses domain data and document titles from the `BuildEnvironment`

parse_titles(*env*)

Parses document and section titles from the `BuildEnvironment`

parse_roles(*env*)

Parses the available roles for cross-referencing objects in the `PythonDomain`, `Standard Domain`, and `reStructuredText Domain`

Parameters

env (`BuildEnvironment`) – the `BuildEnvironment`

parse_objtypes(*env*)

Maps cross-reference roles to their corresponding object types in the `PythonDomain`, `Standard Domain`, and `reStructuredText Domain`

Parameters

env (`BuildEnvironment`) – the `BuildEnvironment`

parse_std_domain(*env*)

Parses cross-reference data from the `Standard Domain`

Parameters

env (`BuildEnvironment`) – the `BuildEnvironment`

parse_py_domain(*env*)

Parses cross-reference data for `PythonDomain` objects

Parameters

env (`BuildEnvironment`) – the `BuildEnvironment`

get_py_target(*entry*, *linkcode_resolve=None*)

Resolves the target for a cross-reference to an object in the `PythonDomain`

Parameters

- **entry** (`ObjectEntry`) – the `ObjectEntry` for the object
- **linkcode_resolve** (`Optional[Callable]`) – function to resolve targets when linking to source code

Returns

the link to the object's corresponding documentation entry or highlighted source code

Return type

`Optional[str]`

add_variants(*qualified_name*, *target*, *is_callable=False*)

Adds substitution information for an object to the `ref_map`

This data is used to replace any `autodoc` cross-reference to the object with a substitution, hyper-linked to the corresponding source code or documentation entry

Tip: See `get_all_xref_variants()` and `replace_autodoc_refs()`

Parameters

- **qualified_name** (`str`) – the fully qualified name of an object (ex. "sphinx_readme.parser.add_variants")
- **target** (`str`) – the refuri of the object's corresponding source code or documentation entry
- **is_callable** (`bool`) – specifies if the object is a method or function

parse_doctree(*app*, *doctree*, *docname*)

Parses cross-reference, admonition, rubric, and toctree data from a resolved doctree

parse_admonitions(*app*, *doctree*, *docname*)

Parses data from generic and specific admonitions

Parameters

doctree (document) – the doctree from one of the `src_files`

parse_intersphinx_nodes(*app*, *doctree*, *docname*)

Parses cross-references that utilize `sphinx.ext.intersphinx`

Parameters

doctree (document) – the doctree from one of the `src_files`

parse_external_node(*external*, *role*, *ref_id*)

parse_tocrees(*app*, *doctree*, *docname*)

Parses the caption and entry data from `toctree` nodes

<p>Caution: Tocrees are currently parsed as if the directive has the <code>:titlesonly:</code> option</p>
--

Parameters

doctree (document) – the doctree from one of the `src_files`

parse_rubrics(*app*, *doctree*, *docname*)

Parses the content from `rubric` directives

get_external_ref(*external*, *objtype*, *ref_id*)

Retrieves external cross-reference data from the `sphinx.ext.intersphinx` inventory

Parameters

- **external** (*str*) – the `:external:` or `:external+pkg:` portion of the xref, if present
- **objtype** (*str*) – the name of the object type being referenced
- **ref_id** (*str*) – the target of the cross-reference

Returns

an `ExternalRef` object if the lookup was successful, otherwise `None`

Return type

Optional[`ExternalRef`]

get_external_id(*external*, *role*, *ref_id*)

Helper function to get the `ref_id` when replacing external xrefs

Return type

Optional[*str*]

is_external_xref(*external*, *role*, *ref_id*)

Helper function to check if a cross-reference is explicitly external

Return type

bool

resolve()

Uses parsed data from to replace cross-references and directives in the `src_files`

Once resolved, files are written to the `out_dir`.

replace_admonitions(*rst_src*, *rst*)

Replaces generic and specific admonition directives with HTML tables or list-table directives, depending on the value of `readme_raw_directive`

Customizing Admonitions

The `icon_map` can be overridden to use custom admonition icons/classes

- See `readme_admonition_icons` and `readme_default_admonition_icon`
-

Parameters

- **rst_src** (*str*) – absolute path of the source file
- **rst** (*str*) – content of the source file

Return type

str

replace_toc trees(*rst_src*, *rst*)

Replaces `toc tree` directives with hyperlinked bullet lists

Note: Entries will link to HTML documentation regardless of the value of `readme_docs_url_type`

Parameters

- **rst_src** (*str*) – absolute path of the source file

- **rst** (*str*) – content of the source file

Return type*str***replace_rst_images**(*rst_src*, *rst*)

Replaces filepaths in image directives with repository links

Example:

```
.. image:: /_static/logo_readme.png
```

would be replaced with

```
.. image:: https://raw.githubusercontent.com/tdkorn/sphinx-readme/main/docs/
source/_static/logo_readme.png
```

Note: Your repository will be used as the image source regardless of the value of `readme_docs_url_type`

Parameters

- **rst_src** (*str*) – absolute path of the source file
- **rst** (*str*) – content of the source file

Return type*str***replace_rubrics**(*rst_src*, *rst*, *force_markup=False*)Replaces `rubric` directives with the section heading character specified by `readme_rubric_heading`If `readme_rubric_heading` is not specified, the rubric will be replaced with bold text instead

...

Example:Consider a source file that contains `.. rubric:: This is a ``rubric`` directive`

- Replacement without specifying `readme_rubric_heading`:

```
**This is a** ``rubric`` **directive**
```

- Replacement if `readme_rubric_heading = "^"`:

```
This is a ``rubric`` directive
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Parameters

- **rst_src** (*str*) – absolute path of the source file
- **rst** (*str*) – content of the source file

Return type*str*

replace_xrefs(*rst_src*, *rst*)

Replaces cross-references from the [Standard Domain](#) and [reStructuredText Domain](#) with substitutions or inline links

Tip: This includes cross-references for any custom objects added by `Sphinx.add_object_type()`

Parameters

- **rst_src** (*str*) – absolute path of the source file
- **rst** (*str*) – content of the source file

Returns

the *rst* with all applicable cross-references replaced by links/substitutions

Return type

str

replace_py_xrefs(*rst_src*, *rst*)

Replace [PythonDomain](#) cross-references with substitutions

These substitutions will be hyperlinked to the corresponding source code or HTML documentation entry, depending on the value of [readme_docs_url_type](#)

Parameters

- **rst_src** (*str*) – absolute path of the source file
- **rst** (*str*) – content of the source file

Return type

str

replace_unresolved_xrefs(*rst*)

Replaces any unresolved cross-references from all domains with inline literals

Return type

str

get_xref_regex(*domains*, *roles=None*, *targets=None*, *xref_type=None*)

Returns the regex to match cross-references

Note: The patterns have the following match groups:

Regular Cross-References

1. The full cross-reference
2. The external role, if present
3. The cross-reference role
4. The cross-reference target

Explicit Title Cross-References

1. The full cross-reference
2. The external role, if present
3. The cross-reference role

4. The explicit title
 5. The cross-reference target
-

Parameters

- **domains** (`Union[str, Iterable[str]]`) – an individual or list of Sphinx object domains to match
- **roles** (`Optional[Union[str, Iterable[str]]]`) – an individual or list of cross-reference roles to match; matches all domain roles if not provided
- **targets** (`Optional[Union[str, Iterable[str]]]`) – an individual or list of targets to match; matches all xrefs if not provided
- **xref_type** (`Optional[str]`) – the xref type to match ("regular" or "title"); returns both if not specified

Returns

the regex pattern to match regular xrefs, xrefs with explicit titles, or a tuple containing both

Return type

`Union[str, Tuple[str, str]]`

get_admonition_regex(*admonition*)

Returns the regex to match a specific admonition directive

Parameters

admonition (`Dict[str, str]`) – a dict containing admonition data

Return type

`str`

get_admonition_icon(*admonition*)

Returns the icon to use for an admonition

Parameters

admonition (`dict`) – a dict of admonition data

Return type

`str`

4.2 The READMEConfig Class

class sphinx_readme.config.READMEConfig(*app*)

Bases: `object`

__init__(*app*)

repo_blob: `str`

The git blob to use when linking to the project's repository

repo_url: *str*

The base URL of the project's repository

blob_url: *str*

The base URL for the `repo_blob` blob of the project's repository

docs_url: *str*

The URL to use when resolving `autodoc` cross-references

get_repo_url()

Generates the URL of the project's repository from the `html_context` dict

Raises

ExtensionError – if `html_context` is undefined or missing values

Return type

str

get_docs_url()

Returns the base URL of the documentation source to use when resolving `autodoc` cross-references

If `docs_url_type` is

- "code": uses the `blob_url`
- "html": uses the `html_baseurl`

Raises

ExtensionError – if `html_baseurl` is undefined

Return type

str

setup_linkcode_resolve(*app*)

Retrieves or defines a `linkcode_resolve()` function for your package

read_rst(*rst_file*, *replace_only=True*, *is_included=False*)

Reads and partially parses an `rst` file

Tip: Files are parsed as follows:

1. If `replace_only` is `True`, only directives are replaced via `replace_only_directives()`
 2. If `readme_include_directive` is `True`, include directives are replaced with the content of the included file; otherwise, the directives will be removed
 3. If `readme_raw_directive` is `False`, raw directives are removed
-

Parameters

- **rst_file** (`Union[str, Path]`) – the `rst` file to read
- **replace_only** (`bool`) – specifies if `only` directives should be replaced or not

Return type

str

`parse_include_directives(rst, rst_file, replace_only=True)`

property `src_files`: `Dict[str, Path]`

Absolute paths of the `readme_src_files` mapped to corresponding output files

property `sources`: `Dict[str, str]`

Absolute paths of source files mapped to their file content

property `out_dir`: `Path`

Absolute path of the directory to save generated rst files to (from `readme_out_dir`)

property `docs_url_type`: `str`

Documentation source type ("code" or "html") for resolving `autodoc` cross-references

See `readme_docs_url_type`

property `repo_host`: `str`

The platform that the project's repository is hosted on

property `image_baseurl`: `str`

The base URL to use when replacing images with `replace_rst_images()`

property `icon_map`: `Dict[str, str]`

A mapping of admonition classes to their icons

The default mapping is as follows, but can be modified via `readme_admonition_icons`

```
{
    'attention': '',
    'caution': '',
    'danger': '',
    'error': '',
    'hint': '',
    'important': '',
    'note': '',
    'tip': '',
    'warning': '',
    'default': ''
}
```

property `admonition_template`: `str`

The template to use when replacing admonitions with `replace_admonitions()`

4.3 The `sphinx_readme.utils` subpackage

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

4.3.1 The `sphinx_readme.utils.git` submodule

`sphinx_readme.utils.git.get_repo_url(context)`

Parses the repository URL from the Sphinx `html_context` dict

Parameters

context (`Dict`) – the `html_context` dict

Returns

the base URL of the project's repository

Raises

ExtensionError – if the repository URL cannot be parsed from `html_context`

Return type

`str`

`sphinx_readme.utils.git.is_valid_username(username, host)`

Validates a username on the given hosting platform.

Username Constraints

GitHub

Username can only contain alphanumeric characters and hyphens; can't start or end with a hyphen or have consecutive hyphens; must be between 1 and 39 characters long.

GitLab

Username can only contain alphanumeric characters, underscores, hyphens, and periods; can't start or end with special characters or contain consecutive special characters; must be between 2 and 255 characters long.

BitBucket

Username can only contain alphanumeric characters, hyphens, and underscores.

Parameters

- **username** (`str`) – the username to validate.
- **host** (`str`) – the platform where the repository is hosted.

Returns

True if the username is valid for the given platform, otherwise False.

Return type

`bool`

`sphinx_readme.utils.git.is_valid_repo(repo, host)`

Validates the name of a git repository on the given hosting platform.

Repository Name Constraints

GitHub

Repository names can only contain alphanumeric characters, hyphens, underscores, and periods; must be between 1 and 100 characters long.

GitLab

Repository name can only contain alphanumeric characters, hyphens, underscores, periods, “+”, or spaces. It must start with a letter, digit, or ‘_’ and be at least 1 character long.

BitBucket

Repository name can only contain alphanumeric characters, hyphens, underscores, and periods. It must start with an alphanumeric character, underscore or period and can’t have consecutive hyphens or end with a hyphen. Must be between 1 and 62 characters long.

Parameters

- **repo** (*str*) – the name of the repository.
- **host** (*str*) – the platform where the repository is hosted.

Returns

True if the repository name is valid, otherwise False.

Return type

bool

`sphinx_readme.utils.git.get_blob_url(repo_url, blob=None, context=None)`

Generates the base URL for a specific blob of a repository

Parameters

- **repo_url** (*str*) – the base URL of the repository
- **blob** (*Optional[str]*) – the blob of the repository to generate the link for
- **context** (*Optional[Dict]*) – the Sphinx `html_context` dict

Return type

str

`sphinx_readme.utils.git.get_repo_host(repo_url)`

Returns the hosting platform of a repository

```
>>> get_repo_host("https://github.com/TDKorn/sphinx-readme")
'github'
```

Parameters

repo_url (*str*) – the URL of the repository

Return type

Optional[str]

`sphinx_readme.utils.git.get_blob(blob)`

Returns the git blob corresponding to blob

The value of blob can be any of "head", "last_tag", or "{blob}"

- "head": returns the hash of the most recent commit; if this commit is tagged, uses the tag instead
- "last_tag": returns the most recent commit tag; if no tags exist, uses "head"
- "{blob}": returns the specified blob as is, for example "master" or "v2.0.1"

Return type*str*`sphinx_readme.utils.git.get_head()`

Returns the hash of the most recent commit

If the most recent commit is tagged, the tag is returned instead

Raises

RuntimeError – if the most recent commit can't be found

Return type*str*`sphinx_readme.utils.git.get_last_tag()`

Returns the most recent commit tag

Raises

RuntimeError – if there are no tagged commits

Return type*str*`sphinx_readme.utils.git.get_repo_dir()`

Returns the root directory of the repository

Returns

A Path object representing the working directory of the repository.

Raises

RuntimeError – if the directory can't be determined

Return type*Path*

4.3.2 The `sphinx_readme.utils.linkcode` submodule

`sphinx_readme.utils.linkcode.get_linkcode_url(blob_url=None, repo_url=None, context=None, blob=None)`

Generates the template URL for linking to highlighted source code

Final links are generated from the template by a `linkcode_resolve()` function

Note: Only one of `blob_url`, `repo_url`, or `context` needs to be specified

Parameters

- **blob_url** (`Optional[str]`) – the base URL for a specific blob of a repository
- **repo_url** (`Optional[str]`) – the base URL of a repository
- **context** (`Optional[Dict]`) – the Sphinx `html_context` dict
- **blob** (`Optional[str]`) – the blob of the repository to generate the link for

Raises

ExtensionError – if none of `blob_url`, `repo_url`, or `context` are provided

Return type

str

```
sphinx_readme.utils.linkcode.get_linkcode_resolve(linkcode_url)
```

Defines and returns a `linkcode_resolve()` function for your package

Parameters

`linkcode_url` (str) – the template URL for linking to source code (see `get_linkcode_url()`)

Return type

Callable

4.3.3 The sphinx_readme.utils.rst submodule

```
sphinx_readme.utils.rst.escape_rst(rst)
```

Escape regex special characters from the content of an rst file

Return type

str

```
sphinx_readme.utils.rst.BEFORE_XREF = ':\\[\\{\\(/"\\'\\-'
```

Characters that are allowed directly before a cross-reference

```
sphinx_readme_utils.rst.AFTER_XREF = '\\.:;!\\?,\"'\\/\\\\\\\\\\\\\\\\]\\\\)\\\\}\\\\\\-'
```

Characters that are allowed directly after a cross-reference

```
sphinx_readme.utils.rst.format_hyperlink(target, text, sub_override=None, force_subs=False)
```

Formats a hyperlink, preserving any inline literals within the text

Since nested inline markup isn't possible, substitutions are used when link text contains inline literals

Example:

```
>>> target = "https://www.github.com/tdkorn/sphinx-readme"
>>> format_hyperlink(target, "The Sphinx README Repository")
('`The Sphinx README Repository <https://www.github.com/tdkorn/sphinx-readme>`_', [])
```

```
>>> format_hyperlink(target, "The ``Sphinx README`` Repository")
('|The Sphinx README Repository|_',
['.. |The Sphinx README Repository| replace:: The ``Sphinx README`` Repository',
 '.. _The Sphinx README Repository: https://www.github.com/tdkorn/sphinx-readme'])
```

```
>>> format_hyperlink(target, "The ``Sphinx README`` Repository", sub_override="repo")
('|repo|_',
 ['.. |repo| replace:: The ``Sphinx README`` Repository',
  '.. _repo: https://www.github.com/tdkorn/sphinx-readme'])
```

Parameters

- **target** (`str`) – the link URL
- **text** (`str`) – the link text
- **sub_override** (`Optional[str]`) – overrides the name for the label/substitution, if applicable

- **force_subs** (*bool*) – boolean indicating if substitutions should be used regardless of inline markup being present

Returns

a tuple containing the formatted hyperlink and a list of substitution definitions

Return type

Tuple[str, List[Optional[str]]]

`sphinx_readme.utils.rst.format_rst(inline_markup, rst, replace_attributes=False)`

Formats text with the specified type of inline markup

Preserves any inline literals, |substitutions|, and `Custom Link Text <https://website.com>`_ within the text

Example:

```
>>> format_rst("bold", "This is part of the ``sphinx_readme.utils`` module")
**This is part of the** ``sphinx_readme.utils`` module**
```

This is part of the sphinx_readme.utils module

Parameters

- **inline_markup** (*str*) – either “bold” or “italic”
- **rst** (*str*) – the rst content to format

Return type

str

`sphinx_readme.utils.rst.replace_only_directives(rst, tags)`

Replaces and removes *only* directives.

The `readme_tags` are temporarily added as *tags*, then the <expression> argument of the directive is evaluated.

- If True, the content will be used
- If False, the directive is removed

Tip: The default value of `readme_tags` is ["readme"]

Expression Examples:

Using default value of `readme_tags = ["readme"]`:

```
.. only:: readme

    This will be included in the generated file

.. only:: html

    This will be excluded from the generated file

.. only:: readme or html

    This will be included in the generated file
```

(continues on next page)

(continued from previous page)

```
.. only:: readme and html
```

This will be excluded from the generated file.

Setting `readme_tags = ["pypi"]` in `conf.py`:

```
.. only:: pypi
```

This will be included in the generated file

```
.. only:: readme
```

This will be excluded from the generated file

```
.. only:: readme or pypi
```

This will be included in the generated file

Parameters

- **rst** (*str*) – the content of an rst file
- **tags** (Tags) – the `sphinx.util.tags.Tags` object

Return type

str

`sphinx_readme.utils.rst.remove_raw_directives(rst)`

Removes all raw directives from `rst`

Parameters

- **rst** (*str*) – the rst to remove raw directives from

Return type

str

`sphinx_readme.utils.rst.replace_attrs(rst)`

Replaces `:attr:` cross-references with inline literals

Tip: When `readme_replace_attrs` is `True`, this function will be called to replace

1. Non-external and unresolved `:attr:` xrefs when `readme_docs_url_type` is `"code"`
 2. Unresolved `:attr:` xrefs when `readme_docs_url_type` is `"html"`
-

Parameters

- **rst** (*str*) – the rst to replace attribute xrefs in

Return type

str

`sphinx_readme.utils.rst.replace_xrefs(rst, roles=None)`

Replaces cross-references in the `PythonDomain` with inline literals

Parameters

- **roles** (`Optional[Union[str, List[str]]]`) – an individual or list of cross-reference roles to match; replaces all roles if not specified
- **rst** (`str`) – the rst to replace cross-references in

Return type`str``sphinx_readme.utils.rst.get_xref_variants(target)`

Returns a list of ways to make a cross-reference to target

Example:

```
>>> get_xref_variants('mod.Class.meth')
['mod.Class.meth', '.mod.Class.meth', '~mod.Class.meth', '~.mod.Class.meth']
```

Parameters**target** (`str`) – the object to generate cross-reference syntax for**Return type**`List[str]``sphinx_readme.utils.rst.get_all_xref_variants(fully_qualified_name)`

Returns all possible cross-reference targets for an object

Example:

```
>>> get_all_xref_variants("sphinx_readme.utils.get_all_xref_variants")
['get_all_xref_variants', '.get_all_xref_variants', '~get_all_xref_variants',
 '~.get_all_xref_variants', 'utils.get_all_xref_variants', '.utils.get_all_xref_variants',
 '~utils.get_all_xref_variants', '~.utils.get_all_xref_variants',
 'sphinx_readme.utils.get_all_xref_variants', '.sphinx_readme.utils.get_all_xref_variants',
 '~sphinx_readme.utils.get_all_xref_variants', '~.sphinx_readme.utils.get_all_xref_variants']
```

Parameters**fully_qualified_name** (`str`) – the fully qualified name of the target (ex. pkg.module.class.method)**Return type**`List[str]`

4.3.4 The `sphinx_readme.utils.sphinx` submodule

class `sphinx_readme.utils.sphinx.ExternalRef(objtype, pkg, version, target, label, ref_id)`Bases: `object`

Data structure to parse external cross-reference data from intersphinx

`__init__(objtype, pkg, version, target, label, ref_id)`

property `id`

property `label`

`sphinx_readme.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

`sphinx_readme.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`

4.3.5 The `sphinx_readme.utils.docutils` submodule

`sphinx_readme.utils.docutils.get_doctree(app, rst, docname='index')`

Generate doctree from a string of `reStructuredText` using Sphinx application.

Return type

`document`

`sphinx_readme.utils.docutils.parse_node_text(node)`

Parses the text from a node, preserving inline literals

Return type

`str`

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

S

- `sphinx_readme.__init__`, [27](#)
- `sphinx_readme.config`, [17](#)
- `sphinx_readme.parser`, [11](#)
- `sphinx_readme.utils`, [19](#)
- `sphinx_readme.utils.docutils`, [27](#)
- `sphinx_readme.utils.git`, [20](#)
- `sphinx_readme.utils.linkcode`, [22](#)
- `sphinx_readme.utils.rst`, [23](#)
- `sphinx_readme.utils.sphinx`, [26](#)

Symbols

`__init__()` (*sphinx_readme.config.READMEConfig* method), 17
`__init__()` (*sphinx_readme.parser.READMEParser* method), 11
`__init__()` (*sphinx_readme.utils.sphinx.ExternalRef* method), 26

A

`add_variants()` (*sphinx_readme.parser.READMEParser* method), 12
`admonition_template` (*sphinx_readme.config.READMEConfig* property), 19
`admonitions` (*sphinx_readme.parser.READMEParser* attribute), 11
`AFTER_XREF` (in module *sphinx_readme.utils.rst*), 23

B

`BEFORE_XREF` (in module *sphinx_readme.utils.rst*), 23
`blob_url` (*sphinx_readme.config.READMEConfig* attribute), 18

C

`config` (*sphinx_readme.parser.READMEParser* attribute), 11
configuration value
 `html_baseurl`, 3
 `html_context`, 3
 `linkcode_resolve`, 8
 `readme_admonition_icons`, 6
 `readme_blob`, 8
 `readme_default_admonition_icon`, 7
 `readme_docs_url_type`, 4
 `readme_include_directive`, 6
 `readme_inline_markup`, 5
 `readme_out_dir`, 4
 `readme_raw_directive`, 5
 `readme_replace_attrs`, 5
 `readme_rubric_heading`, 7
 `readme_src_files`, 4
 `readme_tags`, 6

D

`docs_url` (*sphinx_readme.config.READMEConfig* attribute), 18
`docs_url_type` (*sphinx_readme.config.READMEConfig* property), 19
`domains` (*sphinx_readme.parser.READMEParser* attribute), 11

E

`escape_rst()` (in module *sphinx_readme.utils.rst*), 23
`ExternalRef` (class in *sphinx_readme.utils.sphinx*), 26

F

`format_hyperlink()` (in module *sphinx_readme.utils.rst*), 23
`format_rst()` (in module *sphinx_readme.utils.rst*), 24

G

`get_admonition_icon()`
 (*sphinx_readme.parser.READMEParser* method), 17
`get_admonition_regex()`
 (*sphinx_readme.parser.READMEParser* method), 17
`get_all_xref_variants()` (in module *sphinx_readme.utils.rst*), 26
`get_blob()` (in module *sphinx_readme.utils.git*), 21
`get_blob_url()` (in module *sphinx_readme.utils.git*), 21
`get_conf_val()` (in module *sphinx_readme.utils.sphinx*), 27
`get_docs_url()` (*sphinx_readme.config.READMEConfig* method), 18
`get_doctree()` (in module *sphinx_readme.utils.docutils*), 27
`get_external_id()` (*sphinx_readme.parser.READMEParser* method), 14
`get_external_ref()` (*sphinx_readme.parser.READMEParser* method), 13
`get_head()` (in module *sphinx_readme.utils.git*), 22
`get_last_tag()` (in module *sphinx_readme.utils.git*), 22

`get_linkcode_resolve()` (in module `sphinx_readme.utils.linkcode`), 23
`get_linkcode_url()` (in module `sphinx_readme.utils.linkcode`), 22
`get_py_target()` (`sphinx_readme.parser.READMEParser` method), 12
`get_repo_dir()` (in module `sphinx_readme.utils.git`), 22
`get_repo_host()` (in module `sphinx_readme.utils.git`), 21
`get_repo_url()` (in module `sphinx_readme.utils.git`), 20
`get_repo_url()` (`sphinx_readme.config.READMEConfig` method), 18
`get_xref_regex()` (`sphinx_readme.parser.READMEParser` method), 16
`get_xref_variants()` (in module `sphinx_readme.utils.rst`), 26

H

`html_baseurl`
configuration value, 3
`html_context`
configuration value, 3

I

`icon_map` (`sphinx_readme.config.READMEConfig` property), 19
`id` (`sphinx_readme.utils.sphinx.ExternalRef` property), 26
`image_baseurl` (`sphinx_readme.config.READMEConfig` property), 19
`intersphinx_pkgs` (`sphinx_readme.parser.READMEParser` attribute), 11
`inventory` (`sphinx_readme.parser.READMEParser` attribute), 12
`is_external_xref()` (`sphinx_readme.parser.READMEParser` method), 14
`is_valid_repo()` (in module `sphinx_readme.utils.git`), 20
`is_valid_username()` (in module `sphinx_readme.utils.git`), 20

L

`label` (`sphinx_readme.utils.sphinx.ExternalRef` property), 27
`linkcode_resolve`
configuration value, 8

M

module
`sphinx_readme.__init__`, 27
`sphinx_readme.config`, 17

`sphinx_readme.parser`, 11
`sphinx_readme.utils`, 19
`sphinx_readme.utils.docutils`, 27
`sphinx_readme.utils.git`, 20
`sphinx_readme.utils.linkcode`, 22
`sphinx_readme.utils.rst`, 23
`sphinx_readme.utils.sphinx`, 26

N

`named_inventory` (`sphinx_readme.parser.READMEParser` attribute), 12

O

`objtypes` (`sphinx_readme.parser.READMEParser` attribute), 11
`out_dir` (`sphinx_readme.config.READMEConfig` property), 19

P

`parse_admonitions()` (`sphinx_readme.parser.READMEParser` method), 13
`parse_doctree()` (`sphinx_readme.parser.READMEParser` method), 13
`parse_env()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_external_node()`
(`sphinx_readme.parser.READMEParser` method), 13
`parse_include_directives()`
(`sphinx_readme.config.READMEConfig` method), 18
`parse_intersphinx_nodes()`
(`sphinx_readme.parser.READMEParser` method), 13
`parse_node_text()` (in module `sphinx_readme.utils.docutils`), 27
`parse_objtypes()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_py_domain()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_roles()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_rubrics()` (`sphinx_readme.parser.READMEParser` method), 13
`parse_std_domain()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_titles()` (`sphinx_readme.parser.READMEParser` method), 12
`parse_tocrees()` (`sphinx_readme.parser.READMEParser` method), 13

R

`read_rst()` (`sphinx_readme.config.READMEConfig` method), 18

readme_admonition_icons
 configuration value, 6
 readme_blob
 configuration value, 8
 readme_default_admonition_icon
 configuration value, 7
 readme_docs_url_type
 configuration value, 4
 readme_include_directive
 configuration value, 6
 readme_inline_markup
 configuration value, 5
 readme_out_dir
 configuration value, 4
 readme_raw_directive
 configuration value, 5
 readme_replace_attrs
 configuration value, 5
 readme_rubric_heading
 configuration value, 7
 readme_src_files
 configuration value, 4
 readme_tags
 configuration value, 6
 READMEConfig (class in *sphinx_readme.config*), 17
 READMEParser (class in *sphinx_readme.parser*), 11
 ref_map (*sphinx_readme.parser.READMEParser* attribute), 11
 remove_raw_directives() (in module *sphinx_readme.utils.rst*), 25
 replace_admonitions()
 (*sphinx_readme.parser.READMEParser* method), 14
 replace_attrs() (in module *sphinx_readme.utils.rst*), 25
 replace_only_directives() (in module *sphinx_readme.utils.rst*), 24
 replace_py_xrefs() (*sphinx_readme.parser.READMEParser* method), 16
 replace_rst_images()
 (*sphinx_readme.parser.READMEParser* method), 15
 replace_rubrics() (*sphinx_readme.parser.READMEParser* method), 15
 replace_toctrees() (*sphinx_readme.parser.READMEParser* method), 14
 replace_unresolved_xrefs()
 (*sphinx_readme.parser.READMEParser* method), 16
 replace_xrefs() (in module *sphinx_readme.utils.rst*), 25
 replace_xrefs() (*sphinx_readme.parser.READMEParser* method), 15
 repo_blob (*sphinx_readme.config.READMEConfig* attribute), 17
 repo_host (*sphinx_readme.config.READMEConfig* property), 19
 repo_url (*sphinx_readme.config.READMEConfig* attribute), 17
 resolve() (*sphinx_readme.parser.READMEParser* method), 14
 roles (*sphinx_readme.parser.READMEParser* attribute), 11
 rubrics (*sphinx_readme.parser.READMEParser* attribute), 11

S
 set_conf_val() (in module *sphinx_readme.utils.sphinx*), 27
 setup_linkcode_resolve()
 (*sphinx_readme.config.READMEConfig* method), 18
 sources (*sphinx_readme.config.READMEConfig* property), 19
 sources (*sphinx_readme.parser.READMEParser* attribute), 11
 sphinx_readme.__init__
 module, 27
 sphinx_readme.config
 module, 17
 sphinx_readme.parser
 module, 11
 sphinx_readme.utils
 module, 19
 sphinx_readme.utils.docutils
 module, 27
 sphinx_readme.utils.git
 module, 20
 sphinx_readme.utils.linkcode
 module, 22
 sphinx_readme.utils.rst
 module, 23
 sphinx_readme.utils.sphinx
 module, 26
 src_files (*sphinx_readme.config.READMEConfig* property), 19
 substitutions (*sphinx_readme.parser.READMEParser* attribute), 11

T
 titles (*sphinx_readme.parser.READMEParser* attribute), 11
 toctrees (*sphinx_readme.parser.READMEParser* attribute), 11