sphinxcontrib-viewsource

Release 0.1.4

Aug 03, 2019

Table of Contents

Add an "Edit on GitHub" button to your code examples.

sphinxcontrib-viewsource is a Sphinx extension that enhances the `literalinclude` directive, adding a caption automatically with a link pointing to the source file.

Use it to add an **Edit on GitHub** button to your code source files, or to **show the complete code** when the `literalinclude` directive renders a subset of the file.

Example

Listing 1: Edit on GitHub

```python
print('Hello World!')
```

Add the extension to your project following the *installation instructions*.

Installation

1.1 Getting Started

1. Install sphinxcontrib-viewsource using pip.

```
pip install sphinxcontrib-viewsource
```

2. Add the extension to your Sphinx project `conf.py` file.

```
extensions = ['sphinxcontrib.viewsource']
```

3. Define a caption text and link resolution function at the end of your `conf.py` file:

```python
viewsource_title = 'Edit on GitHub'

def viewsource_resolve_link(file_path, language=None):
  base_url = 'https://github.com/dgarcia360/sphinxcontrib-viewsource/blob/master/docs/
↪snippets/'
  # get the name of the file
  path_split = file_path.split('/')
  file = path_split[len(path_split)-1]
  return base_url + file
```

Find here other *link resolution examples*.

4. Use `viewsource` directive instead of `literalinclude` when rendering code:

```
.. viewsource:: snippets/hello.py
   :language: python
   :lines: 2
```

The `viewsource` extends from `literalinclude`, so you can still use all the directive options.

Note: The directive overwrites the `caption` option with the specified link. If you set the caption option, the extension will not render the link.

5. You can apply custom CSS to style the caption:

```
.code-block-caption .caption-text > a {
  float: right;
}
```

6. Build the docs.

Resolve Link

The `viewsource_resolve_link` function set in `config.py` is editable to fit your documentation needs.

viewsource_resolve_link(*file_path*, *language=None*)
Defines the link resolution strategy for each code of block.

> **Parameters**
>
> - **file_path** (*string*) – Contains the path of the file included.
>
> - **language** (*string*) – Contains the name of the highlighting language if set.
>
> **Returns** URL pointing to the file as string

Here there are some popular configurations that you can use to set up the link resolution strategy.

2.1 Point to an internal file

```python
viewsource_title = 'Get the Code'
def viewsource_resolve_link(file_path, language=None):
  snippets_folder = '/snippets/'
  # get the name of the file
  path_split = file_path.split('/')
  file = path_split[len(path_split)-1]
  return snippets_folder + file
```

2.2 Point to an external file

```python
viewsource_title = 'Edit on Github'

def viewsource_resolve_link(file_path, language=None):
  base_url = 'https://github.com/dgarcia360/sphinxcontrib-viewsource/blob/master/docs/
↪snippets/'
```

(continues on next page)

(continued from previous page)

```python
  # get the name of the file
  path_split = file_path.split('/')
  file = path_split[len(path_split)-1]
  return base_url + file
```

2.3 Resolution strategy per language

```python
viewsource_title = 'Edit on Github'

def viewsource_resolve_link(file_path, language=None):
  # get the name of the file
  path_split = file_path.split('/')
  file = path_split[len(path_split)-1]

  url = 'https://github.com/dgarcia360/sphinxcontrib-viewsource/blob/master/docs/
↪python/%s' % file
  if language == 'java':
      return 'https://github.com/dgarcia360/sphinxcontrib-viewsource/blob/master/docs/
↪java/src/%s' % file
  return url
```

Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

3.1 [0.1.3] - 18 April 2019

3.1.1 Added

- Initial code release

Index

V

viewsource_resolve_link() (*built-in function*),