sphinx-inlinecode

Release 2.0.0

Adam Korn

Feb 18, 2024

README

CHAPTER

ONE

SPHINX-INLINECODE

`sphinx-inlinecode` is a Sphinx extension that embeds source code blocks directly into your documentation as a dropdown.

Example

sphinx_inlinecode.get_code_block(*qualname, obj, highlighter*) View on GitHub

▼View Source Code

```python
def get_code_block(qualname: str, obj: Any, highlighter: PythonLexer) -> BeautifulSoup:
    """Parses and highlights the source code lines of the provided object

    :param qualname: the fully qualified name of the object
    :param obj: the actual object to retrieve the source code lines from
    :param highlighter: the Pygments lexer to highlight the code block with
    :return: the highlighted and fully formatted HTML codeblock to insert
    """
    sourcelines, _ = inspect.getsourcelines(obj)
    initial_indent = len(sourcelines[0]) - len(sourcelines[0].lstrip())

    if initial_indent:  # Remove common leading whitespace
        pattern = fr"[ ]{{{initial_indent}}}(.*)"
        sourcelines = (
            re.sub(pattern, r"\1", line)
            for line in sourcelines
        )
    # Convert to HTML code block with syntax highlighting
    highlighted = highlighter.highlight_block(
        source=''.join(sourcelines),
        lang='python',
        linenos=False
    )
    lines = highlighted.splitlines()
    before, after = re.split(r"<pre>(?:<span></span>)?", lines[0])
    lines[0] = f'{before}<pre><div class="viewcode-block" id="{qualname}">{after}'

    code_block = '\n'.join(lines)
    return wrap_code_block(code_block)
```

Parses and highlights the source code lines of the provided object

Parameters:

- **qualname** (*str*) – the fully qualified name of the object
- **obj** (*Any*) – the actual object to retrieve the source code lines from
- **highlighter** (*PythonLexer*) – the Pygments lexer to highlight the code block

Unlike `sphinx.ext.viewcode`, source code blocks will also be added for `property` and `cached_property` entries

1.1 Installation

To install `sphinx-inlinecode` via pip:

```
pip install sphinx-inlinecode
```

1.2 Configuration

Add the extension to your `conf.py`

```
extensions = [
    "sphinx_inlinecode",
]
```

CHAPTER

TWO

SPHINX_INLINECODE PACKAGE

sphinx_inlinecode.**BAD_OBJTYPES = ('attribute', 'data', 'decorator')**

Object types that can't have code blocks inserted

sphinx_inlinecode.**setup**(*app*)

Return type
Dict[*str*, *Any*]

sphinx_inlinecode.**add_static_path**(*app*)

Add the path for the `_static` folder

sphinx_inlinecode.**add_source_code**(*app*, *exception*)

Inserts source code blocks into documentation entries.

sphinx_inlinecode.**parse_py_domain**(*app*)

Parses all Python objects in the package from the `BuildEnvironment`

Returns
a dictionary mapping fully qualified object names to the actual objects

Return type
Dict[*str*, *Any*]

sphinx_inlinecode.**add_code_blocks**(*file*, *objects*, *highlighter*)

Inserts source code blocks into the provided HTML file and writes the output.

Parameters

- **file** (`Path`) – path to the HTML file.
- **objects** (`Dict[str, Any]`) – dictionary containing the objects in the package
- **highlighter** (`PythonLexer`) – the Pygments lexer to highlight source code blocks with

sphinx_inlinecode.**get_target**(*viewcode_link*)

Parses the fully qualified object name from the viewcode internal reference

Parameters
viewcode_link (`Tag`) – the viewcode internal reference

Returns
the fully qualified name of the referenced object

Return type
str

sphinx_inlinecode.**get_code_block**(*qualname*, *obj*, *highlighter*)

Parses and highlights the source code lines of the provided object

Parameters

- **qualname** (str) – the fully qualified name of the object
- **obj** (Any) – the actual object to retrieve the source code lines from
- **highlighter** (PythonLexer) – the Pygments lexer to highlight the code block with

Returns
the highlighted and fully formatted HTML codeblock to insert

Return type
BeautifulSoup

sphinx_inlinecode.**wrap_code_block**(*code_block*)

Wraps the given code block inside a <details> HTML element

Parameters
code_block (str) – HTML of the code block to wrap

Returns
the wrapped code block

Return type
BeautifulSoup

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX