sphinx-exec-code

Release 0.12

spacemanspiff2007

Jan 09, 2024

CONTENTS:

CHAPTER

ONE

INSTALLATION AND CONFIGURATION

1.1 Installation

```
pip install sphinx-exec-code
```

To use this extension just add it to the `extensions` in your `conf.py`

```
extensions = [
    'sphinx_exec_code',
]
```

1.2 Configuration

The following configuration parameters are available:

Name	Type	Description
`exec_code_working_dir`	`Path` or `str`	The working directory where the code will be executed.
`exec_code_source_folders`	`List` of `Path` or `str`	Additional folders that will be added to PYTHONPATH. Use this to e.g. make imports available.
`exec_code_example_dir`	`Path` or `str`	The directory that is used to create the path to the example files. Defaults to the parent folder of the `conf.py`.
`exec_code_set_utf8_encoding`	`True` or `False`	True enforces utf-8 encoding (can fix encoding errors). Default is `False` except on Windows where it is `True`.

If it's a relative path it will be resolved relative to the parent folder of the `conf.py`

Example:

```
exec_code_working_dir = '..'
exec_code_source_folders = ['../my_src']
exec_code_example_dir = '.'
```

If you are unsure what the values are you can run Sphinx build in verbose mode with `-v -v`. The configured values are logged.

Log output for Example:

```
[exec-code] Working dir: C:\Python\sphinx-exec-code
[exec-code] Source folders: C:\Python\sphinx-exec-code\my_src
[exec-code] Example dir: C:\Python\sphinx-exec-code\doc
[exec-code] Set utf8 encoding: True
```

CHAPTER

TWO

USAGE

2.1 Without options

```
.. exec_code::

   print('Easy!')
```

Generated view

```
print('Easy!')
```

```
Easy!
```

2.2 Options

It's possible to further configure both the code block and the output block with the following options:

hide_code/hide_output:
Will hide the corresponding block

caption/caption_output
Will add a caption above the block

linenos/linenos_output
Will add line numbers

language/language_output:

Will add syntax highlighting for the specified language
The default for the code block is python, the default for the output block is plain text

Example:

```
.. exec_code::
   :linenos:
   :hide_output:
   :caption: This is an important caption
```

(continues on next page)

(continued from previous page)

```
  print('Easy!')
```

Generated view

This is an important caption

```
1 print('Easy!')
```

2.3 Code from files

It's possible to have code in example files with the `filename` option. The folder that is used to resolve to a file name can be *configured*.

Example:

```
.. exec_code::
   :filename: file_example.py
```

Generated view

```
print('This is code from an example file')
```

```
This is code from an example file
```

2.4 Code Markers

It's possible to hide parts of the code (e.g. to setup a working example) and it's possible to skip part of the code execution. This is possible with the `#hide:[start|stop|toggle]` or `#skip:[start|stop|toggle]` marker in the code. Empty lines after a disabling marker will be ignored.

Spaces and dashes are ignored for the case insensitive marker detection so these are all the same:

```
#HIDE:START
# hide: start
# ----- hide: start -----
    # ----- hide: start -----
```

2.4.1 Hiding code parts

```
.. exec_code::

   # --- hide: start ---
   print('Setup!')
   #hide:toggle

   print('Easy!')

   # --- hide: start ---
   print('Hidden!')
   # --- hide: stop ---

   # Note the missing entries!
   print('Visible!')
```

Generated view (note the skipped empty lines after the stop and disabling toggle marker)

```
print('Easy!')

# Note the missing entries!
print('Visible!')
```

```
Setup!
Easy!
Hidden!
Visible!
```

2.4.2 Skipping code parts

```
.. exec_code::

   # --- skip: start ---
   print(f'1 / 0 = {1 / 0}')
   # --- skip: stop ---

   # --- hide: start ---
   print('1 / 0 = 0')
   # --- hide: stop ---
```

Generated view

```
print(f'1 / 0 = {1 / 0}')
```

```
1 / 0 = 0
```

With the combination of `skip` and `hide` it's possible to "simulate" every code.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search