HPCTOOLS

Aug 29, 2023

Contents:

The generated documentation will look like figure *Fig.1*

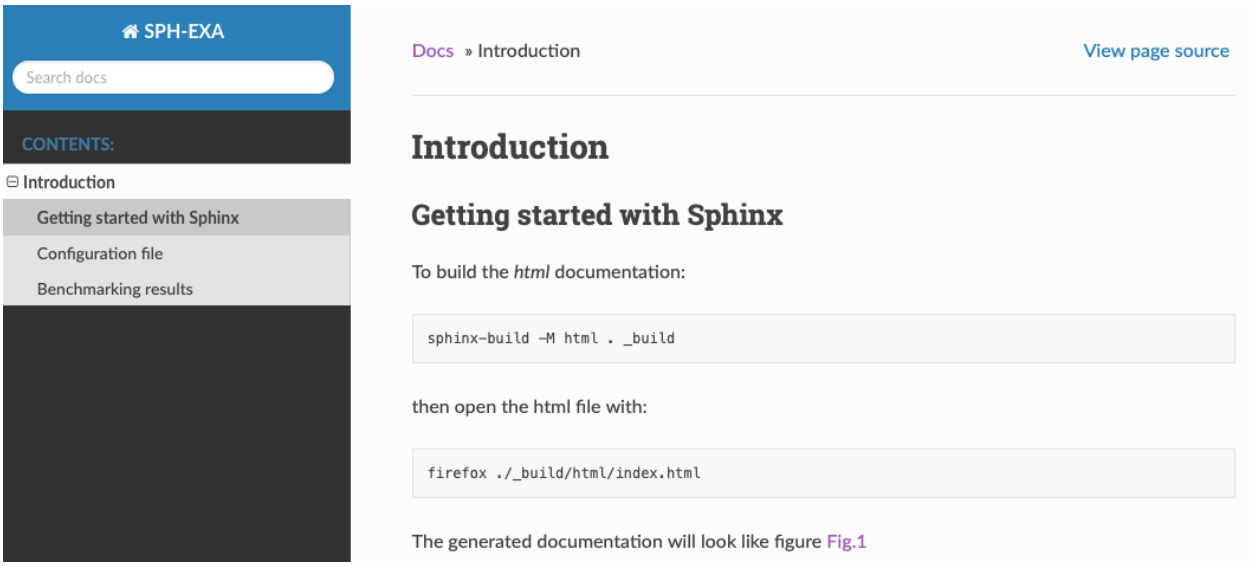

Fig. 1: Title: typical sphinx html documentation

A typical example configuration file is `conf.py`:

```
import re
import sys
import sphinx_rtd_theme
```

CHAPTER 1

Link to Reference Guide

This page tests links to *foo.py* (reference2.rst), see documentation.

1.1 link to module

- `.. automodule::  reframechecks.common.sphexa.foo`
- Link to *foo.py* module
- Link to *foo.py/square* module
- `:mod:`Foo <reframechecks.common.sphexa.foo>``

1.2 link to class

- `.. autoclass::  reframechecks.common.sphexa.foo.Foo`
- Link to *Foo* class
- `:class:`Foo <reframechecks.common.sphexa.foo.Foo>``

1.3 link to method

- `.. automethod::  reframechecks.common.sphexa.foo.hello`
- Inside Class works:
  - Link to *reframechecks.common.sphexa.foo.Foo.bye()* method.
  - Link to *bye* method.
  - `:meth:`bye <reframechecks.common.sphexa.foo.Foo.bye>``

- Not inside Class works:
  - Link to *reframechecks.common.sphexa.foo.hello()* method.
  - Link to *hello* method.
  - Link to *square* method.
  - `:meth:`hello <reframechecks.common.sphexa.foo.hello>`

1.4 link to function

- `.. autofunction:: square`
- Link to *reframechecks.common.sphexa.foo.square()* function.
- Link to *square* function.
- Link to `square (ko)` function.

CHAPTER 2

Reference Guide

This page tests links, see `started.rst`.

2.1 Foo Documentation

2.2 automodule

- reframechecks.common.sphexa/foo.py

foo module.

2.3 autoclass

- reframechecks.common.sphexa/foo.py -> class Foo:

class reframechecks.common.sphexa.foo.**Foo**
Bases: object

Foo class.

bye(*name*)
Print bye addressed to *name*.

Args: name (str): Name to address.

2.4 automethod

- reframechecks.common.sphexa/foo.py ->

foo.**hello**()
Print hello addressed to *name*.

Args: name (str): Name to address.

2.5 autofunction

reframechecks.common.sphexa.foo.**seconds_neigh**(*self*)
Reports *FindNeighbors* time in seconds using the internal timer from the code

reframechecks.common.sphexa.foo.**square**(*a*)
short description of the function square

longish explanation: returns the square of a: a^2

Parameters **a** – an input argument

Returns a*a

reframechecks.common.sphexa.foo.**hello**(*name*)
Print hello addressed to *name*.

Args: name (str): Name to address.

CHAPTER 3

TODO

3.1 link to parameter::

- Link to `prefix` parameter

3.2 Howto

```
class RegressionTest(metaclass=RegressionTestMeta):
    '''Base class for regression tests.'''
    #: The set of reference values for this test.
    #:
    #: The reference values are specified as a scoped dictionary keyed on the
    #: performance variables defined in :attr:`perf_patterns` and scoped under
    #: the system/partition combinations.
    #: The reference itself is a three- or four-tuple that contains the
    #: reference value, the lower and upper thresholds and, optionally, the
    #: measurement unit.
    #: An example follows:
    #:
    #: .. code:: python
    #:
    #:    self.reference = {
    #:        'sys0:part0': {
    #:            'perfvar0': (50, -0.1, 0.1, 'Gflop/s'),
    #:            'perfvar1': (20, -0.1, 0.1, 'GB/s')
    #:        },
    #:        'sys0:part1': {
    #:            'perfvar0': (100, -0.1, 0.1, 'Gflop/s'),
    #:            'perfvar1': (40, -0.1, 0.1, 'GB/s')
    #:        }
```

(continues on next page)

(continued from previous page)

```
    #:    }
    #:
    #: :type: A scoped dictionary with system names as scopes or :class:`None`
    #: :default: ``{}``
    reference = fields.ScopedDictField('reference', typ.Tuple[object])

    @property
    def current_environ(self):
        '''The programming environment that the regression test is currently
        executing with.

        This is set by the framework during the :func:`setup` phase.

        :type: :class:`reframechecks.common.sphexa.environments.Environment`.
        '''
        return self._current_environ
```

CHAPTER 4

JG Cheatsheet

```
# with overline, for parts
* with overline, for chapters
=, for sections
-, for subsections
^, for subsubsections
", for paragraphs
```

CHAPTER 5

Part title..

5.1 Chapter title..

5.1.1 Section title..

Subsection title..

Subsubsection title..

Paragrapth title..

Paragrapth title..

Subtitle: Titles

- All sections marked with the same adornment style are deemed to be at the same level:

5.1.2 Chapter 1 Title

Section 1.1

Section 1.2

5.1.3 Chapter 2 Title

etc. . .

italic / **bold** / *interpreted* / `inline` / `None`

A	B
a	b
aa	bb

- SPH-EXA website ...
- `download conf.py`
- pdf
- open a local file ?.
- how to go to the Images section ?.
- This works in reframe doc: here.

New in version 0.1.

Changed in version 0.1.

Caution: this is a caution

Warning: this is a warning

Tip: this is a tip

Note: this is a note

Notice space after : :

```
Command line: ./bin/reframe -C tutorial/config/settings.py \
 -c tutorial/example1.py -r
Reframe version: 2.x
```

```
> $EBROOTREFRAME/bin/reframe -r $SCRATCH/example1.py
```

```
import os
```

```
#!/bin/bash -l
#SBATCH --job-name="rfm_Example1Test_job"
export X=`date`
```

see 1

5.1.4 Indices and tables

- genindex
- modindex
- search

Python Module Index

r

Index