
snakescale

Release 0.8.0

Dec 28, 2018

Contents

1	Features	3
1.1	API Reference	3
1.2	How to Contribute	5
	Python Module Index	7


```
pip install 'snakescale[full]'
```


- Non-strict wrappers for the data pipelining language Snakemake

1.1 API Reference

`snakescale.available_tools` (*version: str = 'latest'*) → List[str]

`snakescale.available_subtools` (*tool: str, version: str = 'latest'*) → Optional[List[str]]

`snakescale.scale` (*tool: str, subtool: Optional[str] = None, version: str = 'latest', as_uri: bool = True*)
→ str

`snakescale.formatters.clean_picard_style_value` (*value: Union[List[str], str]*) →
Union[List[str], str]

Clean a dictionary of Picard key-value pairs.

`snakescale.formatters.snakecase_to_kebab_case` (*key: str*) → str

Convert snake_case to kebab-case.

`snakescale.formatters.clean_picard_style_key` (*key: str*) → str

Clean a Picard parameter key.

`snakescale.formatters.format_bedtools_params` (*params: Mapping*) → str

Clean a dictionary of bedtools key-value pairs.

`snakescale.formatters.format_bwa_params` (*params: Mapping*) → str

Clean a dictionary of bwa key-value pairs.

`snakescale.formatters.format_dwgsim_params` (*params: Mapping*) → str

Clean a dictionary of dwgsim key-value pairs.

`snakescale.formatters.format_fgbio_params` (*params: Mapping*) → str

Clean a dictionary of fgbio key-value pairs.

`snakescale.formatters.format_kraken_params` (*params: Mapping*) → str

Clean a dictionary of kraken key-value pairs.

`snakescale.formatters.format_picard_params (params: Mapping) → str`

Clean a dictionary of picard key-value pairs.

`snakescale.resources.collect_jvm_resources () → str`

Collect JVM resources allocations the resource dictionary.

`snakescale.resources.collect_picard_style_jvm_resources () → str`

Collect Picard specific JVM resources allocations the resource dictionary.

class `snakescale.utils.ExitCode`

Bases: `int`

The code returned to a parent process by an executable.

Examples

```
>>> from subprocess import call
>>> exit_code = ExitCode(call('ls'))
>>> exit_code.is_ok()
True
```

`is_ok () → bool`

Is this code zero.

class `snakescale.utils.ProcessReturn`

Bases: `tuple`

A collection of STDOUT, STDERR, and exit code from a terminated process.

stdout

Alias for field number 0

stderr

Alias for field number 1

exit_code

Alias for field number 2

__asdict ()

Return a new OrderedDict which maps field names to their values.

classmethod `__make (iterable, new=<built-in method __new__ of type object at 0x886c00>, len=<built-in function len>)`

Make a new ProcessReturn object from a sequence or iterable

__replace (kws)**

Return a new ProcessReturn object replacing specified fields with new values

`snakescale.utils.call_snakemake (arguments: str) → snakescale.utils.ProcessReturn`

Call Snakemake in this thread as if we are using the CLI.

Parameters `arguments` – A string of arguments to pass to Snakemake.

Examples

```
>>> process_return = call_snakemake('')
>>> process_return.exit_code.is_ok()
False
```

(continues on next page)

(continued from previous page)

```
>>> process_return.stderr.strip()
'Error: Snakefile "Snakefile" not present.'
```

`snakescale.testing.run_tool_test` (*tool: str, subtool: Optional[str] = None, version: str = 'latest'*) → `snakescale.utils.ProcessReturn`

Execute a tool and optional subtool test against a specific wrapper version.

1.2 How to Contribute

Pull requests, feature requests, and issues welcome!

S

- `snakescale`, 3
- `snakescale.formatters`, 3
- `snakescale.resources`, 4
- `snakescale.testing`, 5
- `snakescale.utils`, 4

Symbols

`_asdict()` (snakescale.utils.ProcessReturn method), 4
`_make()` (snakescale.utils.ProcessReturn class method), 4
`_replace()` (snakescale.utils.ProcessReturn method), 4

A

`available_subtools()` (in module snakescale), 3
`available_tools()` (in module snakescale), 3

C

`call_snakemake()` (in module snakescale.utils), 4
`clean_picard_style_key()` (in module snakescale.formatters), 3
`clean_picard_style_value()` (in module snakescale.formatters), 3
`collect_jvm_resources()` (in module snakescale.resources), 4
`collect_picard_style_jvm_resources()` (in module snakescale.resources), 4

E

`exit_code` (snakescale.utils.ProcessReturn attribute), 4
`ExitCode` (class in snakescale.utils), 4

F

`format_bedtools_params()` (in module snakescale.formatters), 3
`format_bwa_params()` (in module snakescale.formatters), 3
`format_dwgsim_params()` (in module snakescale.formatters), 3
`format_fgbio_params()` (in module snakescale.formatters), 3
`format_kraken_params()` (in module snakescale.formatters), 3
`format_picard_params()` (in module snakescale.formatters), 3

I

`is_ok()` (snakescale.utils.ExitCode method), 4

P

`ProcessReturn` (class in snakescale.utils), 4

R

`run_tool_test()` (in module snakescale.testing), 5

S

`scale()` (in module snakescale), 3
`snakcase_to_kebab_case()` (in module snakescale.formatters), 3
`snakescale` (module), 3
`snakescale.formatters` (module), 3
`snakescale.resources` (module), 4
`snakescale.testing` (module), 5
`snakescale.utils` (module), 4
`stderr` (snakescale.utils.ProcessReturn attribute), 4
`stdout` (snakescale.utils.ProcessReturn attribute), 4