
smoke-zephyr Documentation

Release 2.0.1

Spencer McIntyre

Dec 13, 2020

Contents

1	smoke_zephyr — Miscellaneous Python classes and functions	1
1.1	Data	1
1.2	Modules	1
2	Indices and tables	5
	Python Module Index	7
	Index	9

smoke_zephyr — Miscellaneous Python classes and functions

1.1 Data

distutils_version = '2.0.1'

A string suitable for being parsed by `distutils.version` classes.

version = '2.0.1'

A string representing the full version information.

version_info = version_info(major=2, minor=0, micro=1)

A tuple representing the version information in the format ('major', 'minor', 'micro')

version_label = ''

A version label such as alpha or beta.

1.2 Modules

1.2.1 `argparse_types` — Extra `argparse` types

Functions

Classes

1.2.2 `configuration` — General configuration manager

Data

Classes

1.2.3 job — Asynchronous job manager

The *JobManager* provides a way to schedule jobs and run tasks asynchronously from within python on the local system. In this case jobs are callback functions defined by the user.

Warning: The timing and scheduling functions within this module are not designed to be precise to the second.

Functions

normalize_job_id (*job_id*)

Convert a value to a job id.

Parameters **job_id** (*int*, *str*) – Value to convert.

Returns The job id.

Return type `uuid.UUID`

Classes

class JobManager (*use_utc=True*, *logger_name=None*)

This class provides a threaded job manager for periodically executing arbitrary functions in an asynchronous fashion.

__init__ (*use_utc=True*, *logger_name=None*)

Parameters

- **use_utc** (*bool*) – Whether or not to use UTC time internally.
- **logger_name** (*str*) – A specific name to use for the logger.

job_add (*callback*, *parameters=None*, *hours=0*, *minutes=0*, *seconds=0*, *tolerate_exceptions=True*, *expiration=None*)

Add a job to the job manager.

Parameters

- **callback** (*function*) – The function to run asynchronously.
- **parameters** (*list*, *tuple*) – The parameters to be provided to the callback.
- **hours** (*int*) – Number of hours to sleep between running the callback.
- **minutes** (*int*) – Number of minutes to sleep between running the callback.
- **seconds** (*int*) – Number of seconds to sleep between running the callback.
- **tolerate_exceptions** (*bool*) – Whether to continue running a job after it has thrown an exception.
- **expiration** (*int*, *datetime.timedelta*, *datetime.datetime*) – When to expire and remove the job. If an integer is provided, the job will be executed that many times. If a datetime or timedelta instance is provided, then the job will be removed after the specified time.

Returns The job id.

Return type `uuid.UUID`

job_count ()

Return the number of jobs.

Returns The number of jobs.

Return type `int`

job_count_enabled ()

Return the number of enabled jobs.

Returns The number of jobs that are enabled.

Return type `int`

job_delete (*job_id*, *wait=True*)

Delete a job.

Parameters

- **job_id** (`uuid.UUID`) – Job identifier to delete.
- **wait** (`bool`) – If the job is currently running, wait for it to complete before deleting it.

job_disable (*job_id*)

Disable a job. Disabled jobs will not be executed.

Parameters **job_id** (`uuid.UUID`) – Job identifier to disable.

job_enable (*job_id*)

Enable a job.

Parameters **job_id** (`uuid.UUID`) – Job identifier to enable.

job_exists (*job_id*)

Check if a job identifier exists.

Parameters **job_id** (`uuid.UUID`) – Job identifier to check.

Return type `bool`

job_is_enabled (*job_id*)

Check if a job is enabled.

Parameters **job_id** (`uuid.UUID`) – Job identifier to check the status of.

Return type `bool`

job_is_running (*job_id*)

Check if a job is currently running. False is returned if the job does not exist.

Parameters **job_id** (`uuid.UUID`) – Job identifier to check the status of.

Return type `bool`

job_run (*callback*, *parameters=None*)

Add a job and run it once immediately.

Parameters

- **callback** (*function*) – The function to run asynchronously.
- **parameters** (*list*, *tuple*) – The parameters to be provided to the callback.

Returns The job id.

Return type `uuid.UUID`

now ()

Return a `datetime.datetime` instance representing the current time.

Return type `datetime.datetime`

now_is_after (dt)

Check whether the datetime instance described in `dt` is after the current time.

Parameters `dt` (`datetime.datetime`) – Value to compare.

Return type `bool`

now_is_before (dt)

Check whether the datetime instance described in `dt` is before the current time.

Parameters `dt` (`datetime.datetime`) – Value to compare.

Return type `bool`

start ()

Start the JobManager thread.

stop ()

Stop the JobManager thread.

class JobRequestDelete

An instance of this class can be returned by a job callback to request that the job be deleted and not executed again.

1.2.4 utilities — Miscellaneous Python classes and functions

Functions

Classes

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

S

smoke_zephyr, 1
smoke_zephyr.argmaxparse_types, 1
smoke_zephyr.configuration, 1
smoke_zephyr.job, 2
smoke_zephyr.utilities, 4

Symbols

`__init__()` (*JobManager method*), 2

D

`distutils_version` (*in module smoke_zephyr*), 1

J

`job_add()` (*JobManager method*), 2

`job_count()` (*JobManager method*), 3

`job_count_enabled()` (*JobManager method*), 3

`job_delete()` (*JobManager method*), 3

`job_disable()` (*JobManager method*), 3

`job_enable()` (*JobManager method*), 3

`job_exists()` (*JobManager method*), 3

`job_is_enabled()` (*JobManager method*), 3

`job_is_running()` (*JobManager method*), 3

`job_run()` (*JobManager method*), 3

`JobManager` (*class in smoke_zephyr.job*), 2

`JobRequestDelete` (*class in smoke_zephyr.job*), 4

N

`normalize_job_id()` (*in module smoke_zephyr.job*), 2

`now()` (*JobManager method*), 3

`now_is_after()` (*JobManager method*), 4

`now_is_before()` (*JobManager method*), 4

S

`smoke_zephyr` (*module*), 1

`smoke_zephyr.argparse_types` (*module*), 1

`smoke_zephyr.configuration` (*module*), 1

`smoke_zephyr.job` (*module*), 2

`smoke_zephyr.utilities` (*module*), 4

`start()` (*JobManager method*), 4

`stop()` (*JobManager method*), 4

V

`version` (*in module smoke_zephyr*), 1

`version_info` (*in module smoke_zephyr*), 1

`version_label` (*in module smoke_zephyr*), 1