
task_processing Documentation

Release LATEST_RELEASE

Yelp Inc

Jun 11, 2018

Contents

1	Generated Docs	3
	Python Module Index	7

Interfaces and shared infrastructure for generic task processing @ Yelp

1.1 task_processing

1.1.1 task_processing package

Subpackages

task_processing.interfaces package

Submodules

task_processing.interfaces.event module

```
class task_processing.interfaces.event.Event
    Bases: pyrsistent._precord.PRecord
```

```
    static validate_task_event (r)
```

```
task_processing.interfaces.event.control_event (**kwargs)
```

```
task_processing.interfaces.event.json_deserializer (dct)
```

```
task_processing.interfaces.event.json_serializer (o)
```

```
task_processing.interfaces.event.task_event (**kwargs)
```

task_processing.interfaces.runner module

```
class task_processing.interfaces.runner.Runner
    Bases: object
```

```
    TASK_CONFIG_INTERFACE
```

```
    alias of task_processing.interfaces.task_executor.DefaultTaskConfigInterface
```

```
kill (task_id)
run (task_config)
```

task_processing.interfaces.task_executor module

```
class task_processing.interfaces.task_executor.DefaultTaskConfigInterface
    Bases: pyrsistent._precord.PRecord
```

```
class task_processing.interfaces.task_executor.TaskExecutor
    Bases: object
```

The core interface for Task Processing This is the class you want to implement to add a new TaskExecutor

TASK_CONFIG_INTERFACE

The interface, specified as a PRecord of objects that you will be passing as task_configs to run
alias of *DefaultTaskConfigInterface*

```
get_event_queue ()
    Get queue of events
```

Returns TBD

```
kill (task_id)
    Kill the specified task
```

Parameters **task_id** (*str*) – The task that you want to kill

```
run (task_config)
    Run the supplied task
```

Parameters **task_config** – An object satisfying the TASK_CONFIG_INTERFACE

The executor should start running the provided task and return the task id.

Returns **str task_id** Callers get the id of the task that was run
to check status or kill it later

```
stop ()
    Stop the executor stack
```

Module contents

`task_processing.plugins` package

Subpackages

`task_processing.plugins.mesos` package

Submodules

`task_processing.plugins.mesos.execution_framework` module

`task_processing.plugins.mesos.mesos_executor` module

`task_processing.plugins.mesos.translator` module

Module contents

Module contents

`task_processing.runners` package

Submodules

`task_processing.runners.async` module

```
class task_processing.runners.async.Async (executor, callbacks=None)
```

```
    Bases: task_processing.interfaces.runner.Runner
```

```
    callback_loop ()
```

```
    kill (task_id)
```

```
    run (task_config)
```

```
    stop ()
```

```
exception task_processing.runners.async.AsyncError
```

```
    Bases: Exception
```

```
class task_processing.runners.async.EventHandler (predicate, cb)
```

```
    Bases: tuple
```

```
    cb
```

```
        Alias for field number 1
```

```
    predicate
```

```
        Alias for field number 0
```

`task_processing.runners.promise` module

```
class task_processing.runners.promise.Promise (executor, futures_executor)
```

```
    Bases: task_processing.interfaces.runner.Runner
```

kill (*task_id*)

run (*task_config*)

Schedules execution of the supplied task

Parameters **task_config** – An object satisfying the TASK_CONFIG_INTERFACE

Returns A Future object representing the execution of the task.

stop ()

task_processing.runners.subscription module

class task_processing.runners.subscription.**Subscription** (*executor, queue*)

Bases: *task_processing.interfaces.runner.Runner*

event_producer ()

kill (*task_id*)

run (*task_config*)

stop ()

task_processing.runners.sync module

class task_processing.runners.sync.**Sync** (*executor*)

Bases: *task_processing.interfaces.runner.Runner*

kill (*task_id*)

run (*task_config*)

stop ()

Module contents

Module contents

1.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

t

- `task_processing`, 6
- `task_processing.interfaces`, 5
 - `task_processing.interfaces.event`, 3
 - `task_processing.interfaces.runner`, 3
 - `task_processing.interfaces.task_executor`, 4
- `task_processing.plugins`, 5
- `task_processing.runners`, 6
 - `task_processing.runners.async`, 5
 - `task_processing.runners.promise`, 5
 - `task_processing.runners.subscription`, 6
 - `task_processing.runners.sync`, 6

A

Async (class in task_processing.runners.async), 5
 AsyncError, 5

C

callback_loop() (task_processing.runners.async.Async method), 5
 cb (task_processing.runners.async.EventHandler attribute), 5
 control_event() (in module task_processing.interfaces.event), 3

D

DefaultTaskConfigInterface (class in task_processing.interfaces.task_executor), 4

E

Event (class in task_processing.interfaces.event), 3
 event_producer() (task_processing.runners.subscription.Subscription method), 6
 EventHandler (class in task_processing.runners.async), 5

G

get_event_queue() (task_processing.interfaces.task_executor.TaskExecutor method), 4

J

json_deserializer() (in module task_processing.interfaces.event), 3
 json_serializer() (in module task_processing.interfaces.event), 3

K

kill() (task_processing.interfaces.runner.Runner method), 3
 kill() (task_processing.interfaces.task_executor.TaskExecutor method), 4
 kill() (task_processing.runners.async.Async method), 5

kill() (task_processing.runners.promise.Promise method), 5
 kill() (task_processing.runners.subscription.Subscription method), 6
 kill() (task_processing.runners.sync.Sync method), 6

P

predicate (task_processing.runners.async.EventHandler attribute), 5
 Promise (class in task_processing.runners.promise), 5

R

run() (task_processing.interfaces.runner.Runner method), 4
 run() (task_processing.interfaces.task_executor.TaskExecutor method), 4
 run() (task_processing.runners.async.Async method), 5
 run() (task_processing.runners.promise.Promise method), 6
 run() (task_processing.runners.subscription.Subscription method), 6
 run() (task_processing.runners.sync.Sync method), 6
 Runner (class in task_processing.interfaces.runner), 3

S

stop() (task_processing.interfaces.task_executor.TaskExecutor method), 4
 stop() (task_processing.runners.async.Async method), 5
 stop() (task_processing.runners.promise.Promise method), 6
 stop() (task_processing.runners.subscription.Subscription method), 6
 stop() (task_processing.runners.sync.Sync method), 6
 Subscription (class in task_processing.runners.subscription), 6
 Sync (class in task_processing.runners.sync), 6

T

TASK_CONFIG_INTERFACE

(task_processing.interfaces.runner.Runner
attribute), 3
TASK_CONFIG_INTERFACE
(task_processing.interfaces.task_executor.TaskExecutor
attribute), 4
task_event() (in module
task_processing.interfaces.event), 3
task_processing (module), 6
task_processing.interfaces (module), 5
task_processing.interfaces.event (module), 3
task_processing.interfaces.runner (module), 3
task_processing.interfaces.task_executor (module), 4
task_processing.plugins (module), 5
task_processing.runners (module), 6
task_processing.runners.async (module), 5
task_processing.runners.promise (module), 5
task_processing.runners.subscription (module), 6
task_processing.runners.sync (module), 6
TaskExecutor (class in
task_processing.interfaces.task_executor),
4

V

validate_task_event() (task_processing.interfaces.event.Event
static method), 3