rodario Documentation

Release 1.0.0a8

haliphax

Dec 14, 2017

Contents

1 Connection 3

2 The Registry 5

3 Actors and Proxies 7

4 Decorators 11

5 Exceptions 13

6 Utilities 15

7 Indices and tables 17

Python Module Index 19

A simple, redis-backed Python actor framework

- Homepage

CHAPTER 1

Connection

By default, rodario will use a `StrictRedis` connection to localhost on the default port. If you wish to override this behavior, then replace the `rodario.get_redis_connection` method after import:

```
from redis import StrictRedis
import rodario
rodario.get_redis_connection = lambda: StrictRedis(host='1.2.3.4')
```

CHAPTER 2

The Registry

class rodario.registry.**Registry**
Actor registry class (singleton wrapper)

static __new__(*prefix=None*)
Retrieve the singleton instance for Registry.

Parameters **prefix** (*str*) – Optional prefix for redis key names

Return type *rodario.registry._RegistrySingleton*

class rodario.registry.**_RegistrySingleton**(*prefix=None*)
Singleton for actor registry

__init__(*prefix=None*)
Initialize the registry.

Parameters **prefix** (*str*) – Optional prefix for redis key names

actors
Retrieve a list of registered actors.

Return type set

exists(*uuid*)
Test whether an actor exists in the registry.

Parameters **uuid** (*str*) – UUID of the actor to check for

Return type bool

get_proxy(*uuid*)
Return an ActorProxy for the given UUID.

Parameters **uuid** (*str*) – The UUID to return a proxy object for

Return type *rodario.actors.ActorProxy*

register(*uuid*)
Register a new actor.

Parameters **uuid** (*str*) – The UUID of the actor to register

unregister(*uuid*)
Unregister an existing actor.

Parameters **uuid** (*str*) – The UUID of the actor to unregister

CHAPTER 3

Actors and Proxies

class rodario.actors.**Actor**(*uuid=None*)
Base Actor class

__init__(*uuid=None*)
Initialize the Actor object.

Parameters **uuid** (*str*) – Optionally-provided UUID

is_alive
Return True if this Actor is still alive.

Return type bool

join(*channel*, *func=None*)
Join this Actor to a pubsub cluster channel.

Parameters

- **channel** (*str*) – The channel to join
- **func** (*callable*) – The message handler function

part(*channel*)
Remove this Actor from a pubsub cluster channel.

Parameters **channel** (*str*) – The channel to part

proxy()
Wrap this Actor in an ActorProxy object.

Return type *rodario.actors.ActorProxy*

start()
Fire up the message handler thread.

stop()
Kill the message handler thread.

class rodario.actors.**ActorProxy**(*actor=None*, *uuid=None*)
Proxy object that fires calls to an actor over redis pubsub

__init__(*actor=None*, *uuid=None*)
Initialize instance of ActorProxy.

Accepts either an Actor object to clone or a UUID, but not both.

Parameters

- **actor** (rodario.actors.Actor) – Actor to clone
- **uuid** (*str*) – UUID of Actor to clone

_proxy(*method_name*, **args*, ***kwargs*)
Proxy a method call to redis pubsub.

This method is not meant to be called directly. Instead, it is used by the proxy's self-generated methods to provide the proxy with the same public API as the actor it represents.

Parameters

- **method_name** (*str*) – The method to proxy
- **args** (*tuple*) – The arguments to pass
- **kwargs** (*dict*) – The keyword arguments to pass

Return type multiprocessing.Queue

proxyid = None
This proxy object's UUID for creating unique channels

class rodario.actors.**ClusterProxy**(*channel*)
Proxy object responsible for multiple actors

This class is meant to be inherited by child objects which can provide their own API methods for coordinating the Actors in their channel.

__init__(*channel*)
Initialize instance of ClusterProxy.

Parameters **channel** (*str*) – The cluster channel to use

_proxy(*method_name*, **args*, ***kwargs*)
Proxy a method call to redis pubsub.

Use this method in your child objects which inherit from ClusterProxy to provide the proxy with some representation of the public API for the Actors it represents.

Paramstr method_name The method to proxy

Parameters

- **args** (*tuple*) – The arguments to pass
- **kwargs** (*dict*) – The keyword arguments to pass

Return type *rodario.future.Future*

Returns A Future whose first value is the number of expected responses

channel = None
Cluster channel

proxyid = None
This proxy object's UUID for creating unique channels

class rodario.future.**Future**(*queue*)
Custom response type for proxied method calls

__init__(*queue*)
Initialize the Future by saving a reference to the Queue

Parameters **queue** (*multiprocessing.Queue*) – The response queue to wrap

get(**args*, ***kwargs*)
Resolve and return the proxied method call's value.

Return type mixed

ready
Return True if the response value is available.

Return type bool

CHAPTER 4

Decorators

class rodario.decorators.**DecoratedMethod**(*func*, *decorations=None*, *before=None*, *after=None*)

Generic decorated method

__init__(*func*, *decorations=None*, *before=None*, *after=None*)

Wrap the given function.

Parameters

- **func** (*function*) – The function to wrap
- **decorations** (*set*) – The decorator tags to attach
- **before** (*list*) – The list of before-hook functions
- **after** (*list*) – The list of after-hook functions

after = []

List of after-hook functions

before = []

List of before-hook functions

static decorate(*func*, *decorations=None*, *before=None*, *after=None*)

Decorate the given function. If it is already a *DecoratedMethod*, it will be appended to rather than overwritten.

Parameters

- **decorations** (*set*) – The decorator tags to attach
- **before** (*list*) – The list of before-hook functions
- **after** (*list*) – The list of after-hook functions

Return type *rodario.decorators.DecoratedMethod*

Returns A `DecoratedMethod` wrapper around `func`

decorations = set([])

Set of decorator tags

rodario.decorators.**singular**(*func*)

First-come, first-served cluster channel call. Needs some work; should accept parameters for context and expiry.

Parameters **func** (*function*) – The function to wrap

Return type *rodario.decorators.DecoratedMethod*

CHAPTER 5

Exceptions

class rodario.exceptions.**InvalidActorException**
Raised when a referenced actor does not exist

class rodario.exceptions.**InvalidProxyException**
Raised when a proxy is not given a valid object to wrap

class rodario.exceptions.**UUIDInUseException**
Raised during UUID registration if the UUID is already taken

class rodario.exceptions.**RegistrationException**
Raised when actor registration fails

class rodario.exceptions.**EmptyClusterException**
Raised when a message is passed to an empty cluster channel

CHAPTER 6

Utilities

Utility module for rodario framework

rodario.util.**acquire_lock**(*name*, *expiry=None*, *context=None*, *conn=None*)
Acquire a lock in redis.

Parameters

- **name** (*str*) – Name of the lock to acquire
- **expiry** (*int*) – The duration of the lock (in seconds)
- **context** (*str*) – The context to apply to the lock name
- **conn** (*redis.Connection*) – The redis connection to use

Return type bool

Returns Whether or not the lock was acquired

CHAPTER 7

Indices and tables

- genindex
- search

Python Module Index

r

rodario.util, 15

Index

Symbols

_RegistrySingleton (class in rodario.registry), 5
__init__() (rodario.actors.Actor method), 7
__init__() (rodario.actors.ActorProxy method), 7
__init__() (rodario.actors.ClusterProxy method), 8
__init__() (rodario.decorators.DecoratedMethod method), 11
__init__() (rodario.future.Future method), 8
__init__() (rodario.registry._RegistrySingleton method), 5
__new__() (rodario.registry.Registry static method), 5
_proxy() (rodario.actors.ActorProxy method), 8
_proxy() (rodario.actors.ClusterProxy method), 8

A

acquire_lock() (in module rodario.util), 15
Actor (class in rodario.actors), 7
ActorProxy (class in rodario.actors), 7
actors (rodario.registry._RegistrySingleton attribute), 5
after (rodario.decorators.DecoratedMethod attribute), 11

B

before (rodario.decorators.DecoratedMethod attribute), 11

C

channel (rodario.actors.ClusterProxy attribute), 8
ClusterProxy (class in rodario.actors), 8

D

decorate() (rodario.decorators.DecoratedMethod static method), 11
DecoratedMethod (class in rodario.decorators), 11
decorations (rodario.decorators.DecoratedMethod attribute), 11

E

EmptyClusterException (class in rodario.exceptions), 13
exists() (rodario.registry._RegistrySingleton method), 5

F

Future (class in rodario.future), 8

G

get() (rodario.future.Future method), 9
get_proxy() (rodario.registry._RegistrySingleton method), 5

I

InvalidActorException (class in rodario.exceptions), 13
InvalidProxyException (class in rodario.exceptions), 13
is_alive (rodario.actors.Actor attribute), 7

J

join() (rodario.actors.Actor method), 7

P

part() (rodario.actors.Actor method), 7
proxy() (rodario.actors.Actor method), 7
proxyid (rodario.actors.ActorProxy attribute), 8
proxyid (rodario.actors.ClusterProxy attribute), 8

R

ready (rodario.future.Future attribute), 9
register() (rodario.registry._RegistrySingleton method), 5
RegistrationException (class in rodario.exceptions), 13
Registry (class in rodario.registry), 5
rodario.util (module), 15

S

singular() (in module rodario.decorators), 11
start() (rodario.actors.Actor method), 7
stop() (rodario.actors.Actor method), 7

U

unregister() (rodario.registry._RegistrySingleton method), 6
UUIDInUseException (class in rodario.exceptions), 13