Python etcd Client Documentation

Release 1.1.3

Dustin Oprea

August 19, 2014

Contents

Contents:

CHAPTER 1

etcd package

1.1 Subpackages

1.1.1 etcd.modules package

Submodules

etcd.modules.leader module

class etcd.modules.leader.**LeaderMod**(*client*)
Bases: etcd.common_ops.CommonOps

'Leader' functionality for consensus-based assignment. If multiple processes try to assign different simple strings to the given key, the first will succeed and block the others until the TTL expires. The same process repeats for all subsequent assignments.

delete(*key*, *value*)

get(*key*)

set_or_renew(*key*, *value*, *ttl*)

etcd.modules.lock module

class etcd.modules.lock.**LockMod**(*client*)
Bases: etcd.common_ops.CommonOps

get_lock(*lock_name*, *ttl*)

get_rlock(*lock_name*, *instance_value*, *ttl*)

Module contents

1.2 Submodules

1.2.1 etcd.client module

class etcd.client.**Client**(*host='127.0.0.1'*, *port=4001*, *is_ssl=False*, *ssl_do_verify=True*, *ssl_ca_bundle_filepath=None*, *ssl_client_cert_filepath=None*, *ssl_client_key_filepath=None*)

Bases: object

The main channel of functionality for the client. Connects to the server, and provides functions via properties.

Parameters

- **host** (*string*) – Hostname or IP of server
- **port** (*int*) – Port of server
- **is_ssl** (*bool*) – Whether to use 'http://' or 'https://'.
- **ssl_do_verify** (*bool or None*) – Whether to verify the certificate hostname.
- **ssl_ca_bundle_filepath** (*string or None*) – A bundle of rootCAs for verifications.
- **ssl_client_cert_filepath** (*string or None*) – A client certificate, for authentication.
- **ssl_client_key_filepath** (*string or None*) – A client key, for authentication.

Raises ValueError

directory

Return an instance of the class having the directory functionality.

Return type etcd.directory_ops.DirectoryOps

inorder

Return an instance of the class having the "in-order keys" functionality.

Return type etcd.inorder_ops.InOrderOps

module

Return an instance of the class that hosts the functionality provided by individual modules.

Return type etcd.client._Modules

node

Return an instance of the class having the general node functionality.

Return type etcd.node_ops.NodeOps

prefix

Return the URL prefix for the server.

Return type string

send(*version*, *verb*, *path*, *value=None*, *parameters=None*, *data=None*, *module=None*, *return_raw=False*, *allow_reconnect=True*)

Build and execute a request.

Parameters

- **version** (*int*) – Version of API
- **verb** (*string*) – Verb of request ('get', 'post', etc..)

- **path** (*string*) – URL path
- **value** (*scalar or None*) – Value to be converted to string and passed as "value" in the POST data.
- **parameters** (*dictionary or None*) – Dictionary of values to be passed via URL query.
- **data** (*dictionary or None*) – Dictionary of values to be passed via POST data.
- **module** (*string or None*) – Name of the etcd module that hosts the functionality.
- **return_raw** (*bool*) – Whether to return a `etcd.response.ResponseV2` object or the raw Requests response.
- **allow_reconnect** (*bool*) – Allow the client to consider alternate hosts if the current host fails connection.

Returns Response object

Return type `etcd.response.ResponseV2`

server
Return an instance of the class having the server functionality.

Return type `etcd.server_ops.ServerOps`

session

stat
Return an instance of the class having the stat functionality.

Return type `etcd.stat_ops.StatOps`

1.2.2 etcd.common_ops module

class `etcd.common_ops.`**CommonOps**(*client*)
Bases: `object`

Base-class of 'ops' modules.

Parameters **client** (`etcd.client.Client`) – Client instance.

compare_and_delete(*path*, *is_dir*, *current_value=None*, *current_index=None*, *is_recursive=None*)
The base compare-and-delete function for atomic deletes. A combination of criteria may be used if necessary.

Parameters

- **path** (*string*) – Key
- **is_dir** (*bool*) – If the node is a directory
- **current_value** (*string or None*) – Current value to check
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type `etcd.response.ResponseV2`

get_fq_node_path(*path*)
Return the full path of the given key.

Parameters **path** (*string*) – Key

get_text (*reason*, *path*, *version=2*)
Execute a request that will return flat text.

Parameters

- **reason** (*string*) – Brief phrase describing the request
- **path** (*string*) – URL path
- **version** (*int*) – API version

Returns Response text

Return type string

validate_path (*path*)
Validate the key that we were given.

Parameters **path** (*string*) – Key

Raises ValueError

1.2.3 etcd.config module

etcd.config.**HOST_FAIL_WAIT_S = 5**
Number of seconds that must elapse before we're allowed to retry a host.

1.2.4 etcd.directory_ops module

class etcd.directory_ops.**DirectoryOps** (*client*)
Bases: etcd.common_ops.CommonOps

Functions specific to directory management.

create (*path*, *ttl=None*)
A normal node-set will implicitly create directories on the way to setting a value. This call exists for when you'd like to -explicitly- create one.

We implicitly fail if the directory already exists.

Parameters

- **path** (*string*) – Key
- **ttl** (*int or None*) – Time until removed

Returns Response object

Return type etcd.response.ResponseV2

Raises EtcdAlreadyExistsException

delete (*path*, *current_value=None*, *current_index=None*)
Delete the given directory. It must be empty.

Parameters

- **path** (*string*) – Key
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

delete_if_index (*path*, *current_index*)
Only delete the given directory if the node is at the given index. It must be empty.

Parameters

- **path** (*string*) – Key
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

delete_recursive (*path*, *current_index=None*)
Delete the given directory, along with any children.

Parameters

- **path** (*string*) – Key
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

delete_recursive_if_index (*path*, *current_index*)
Only delete the given directory (and its children) if the node is at the given index.

Parameters

- **path** (*string*) – Key
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

1.2.5 etcd.exceptions module

exception etcd.exceptions.**EtcdAlreadyExistsException**
Bases: etcd.exceptions.EtcdException

Raised when a directory can't be created because it already exists.

exception etcd.exceptions.**EtcdError**
Bases: etcd.exceptions.EtcdException

The base error for the client.

exception etcd.exceptions.**EtcdException**
Bases: exceptions.Exception

The base exception for the client.

exception etcd.exceptions.**EtcdPreconditionException**
Bases: etcd.exceptions.EtcdException

Raised when a CAS condition fails.

1.2.6 etcd.inorder_ops module

class etcd.inorder_ops.**InOrderOps**(*client*)
Bases: etcd.common_ops.CommonOps

The functions having to do with in-order keys.

get_inorder(*path*)
Get an instance of the in-order directory class for a specific key.

Parameters **path** (*string*) – Key

Returns A _InOrder instance for the given path.

Return type etcd.inorder_ops._InOrder

1.2.7 etcd.node_ops module

class etcd.node_ops.**NodeOps**(*client*)
Bases: etcd.common_ops.CommonOps

Common key-value functions.

compare_and_swap(*path*, *value*, *current_value=None*, *current_index=None*, *prev_exists=None*, *ttl=None*)
The base compare-and-swap function for atomic comparisons. A combination of criteria may be used if necessary.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **current_value** (*scalar or None*) – Current value to check
- **current_index** (*int or None*) – Current index to check
- **prev_exists** (*bool or None*) – Whether the node should exist or not
- **ttl** (*int or None*) – The number of seconds until the node expires

Returns Response object

Return type etcd.response.ResponseV2

Raises etcd.exceptions.EtcdPreconditionException

create_only(*path*, *value*, *ttl=None*)
A convenience function that will only set a node if it doesn't already exist.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **ttl** (*int or None*) – The number of seconds until the node expires

Returns Response object

Return type etcd.response.ResponseV2

delete(*path*, *current_value=None*, *current_index=None*)
Delete the given node.

Parameters

- **path** (*string*) – Node key
- **current_value** (*string or None*) – Current value to check
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

delete_if_index (*path*, *current_index*)
Only delete the given node if it's at the given index.

Parameters

- **path** (*string*) – Key
- **current_index** (*int or None*) – Current index to check

Returns Response object

Return type etcd.response.ResponseV2

delete_if_value (*path*, *current_value*)
Only delete the given node if it's at the given value.

Parameters

- **path** (*string*) – Key
- **current_value** (*string*) – Current value to check

Returns Response object

Return type etcd.response.ResponseV2

get (*path*, *recursive=False*)
Get the given node.

Parameters

- **path** (*string*) – Node key
- **recursive** (*bool*) – Node is a directory, and we want to read it recursively.

Returns Response object

Return type etcd.response.ResponseV2

Raises KeyError

set (*path*, *value*, *ttl=None*)
Set the given node.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **ttl** (*int or None*) – Number of seconds until expiration

Returns Response object

Return type etcd.response.ResponseV2

update_if_index (*path*, *value*, *current_index*, *ttl=None*)
A convenience function that will only set a node if its existing "modified index" matches.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **current_index** (*int*) – Current index to check
- **ttl** (*int or None*) – The number of seconds until the node expires

Returns Response object

Return type etcd.response.ResponseV2

update_if_value (*path*, *value*, *current_value*, *ttl=None*)
A convenience function that will only set a node if its existing value matches.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **current_value** (*scalar or None*) – Current value to check
- **ttl** (*int or None*) – The number of seconds until the node expires

Returns Response object

Return type etcd.response.ResponseV2

update_only (*path*, *value*, *ttl=None*)
A convenience function that will only set a node if it already exists.

Parameters

- **path** (*string*) – Node key
- **value** (*scalar*) – Value to assign
- **ttl** (*int or None*) – The number of seconds until the node expires

Returns Response object

Return type etcd.response.ResponseV2

wait (*path*, *recursive=False*)
Long-poll on the given path until it changes.

Parameters

- **path** (*string*) – Node key
- **recursive** (*bool*) – Wait on any change in the given directory or any of its descendants.

Returns Response object

Return type etcd.response.ResponseV2

Raises KeyError

1.2.8 etcd.response module

class etcd.response.**ResponseV2** (*response*, *request_verb*, *request_path*)
Bases: object

An object that describes a response for every V2 request.

Parameters

- **response** (*requests.models.Response*) – Raw Requests response object
- **request_verb** (*string*) – Request verb ('get', post', 'put', etc..)
- **request_path** (*string*) – Node key

Returns Response object

Return type etcd.response.ResponseV2

class etcd.response.**ResponseV2AliveDirectoryNode**(*action*, *node*)
Bases: etcd.response.ResponseV2DirectoryNode

Represents a directory node, which may also be accompanied by children that can be enumerated.

children

initialize(*node*)

is_collection

class etcd.response.**ResponseV2AliveNode**(*action*, *node*)
Bases: etcd.response.ResponseV2BasicNode

Base-class representing a single, non-deleted node.

initialize(*node*)

class etcd.response.**ResponseV2BasicNode**(*action*, *node*)
Bases: object

Base-class representing all nodes: deleted, alive, or a collection.

Parameters

- **action** (*string*) – Action type
- **node** (*dictionary*) – Node dictionary

Returns Response object

Return type etcd.response.ResponseV2

initialize(*node*)
This function acts as the constructor for subclasses.

Parameters **node** (*dictionary*) – Node dictionary

is_collection
If the node is a directory, do we have the collection of children nodes?

Return type bool

is_deleted
Is the node deleted?

Return type bool

is_directory
Is the node a directory?

Return type bool

class etcd.response.**ResponseV2DeletedDirectoryNode**(*action*, *node*)
Bases: etcd.response.ResponseV2DirectoryNode

Represents a single DIRECTORY node either appearing in isolation or among siblings.

is_deleted

class etcd.response.**ResponseV2DeletedNode**(*action*, *node*)
Bases: etcd.response.ResponseV2BasicNode

Represents a single, deleted node.

is_deleted

class etcd.response.**ResponseV2DirectoryNode**(*action*, *node*)
Bases: etcd.response.ResponseV2BasicNode

A base-class representing a single directory node.

is_directory

1.2.9 etcd.server_ops module

class etcd.server_ops.**ServerOps**(*client*)
Bases: etcd.common_ops.CommonOps

Functions that query the server for cluster-level information.

get_dashboard_url()
Return the URL for the dashboard on the server currently connected- to.

Returns URL

Return type string

get_leader_url_prefix()
Return the URL prefix of the leader host.

Returns URL prefix

Return type string

get_machines()
Return the list of servers in the cluster represented as nodes.

Returns Response object

Return type etcd.response.ResponseV2

get_version()
Return a string representing the version of the server that we're connected to.

Returns Version

Return type string

1.3 Module contents

CHAPTER 2

Indices and tables

- *genindex*
- *modindex*
- *search*

e