

Package ‘etseed’

September 8, 2016

Title Client for 'etcd', a 'Key-value' Database

Description Client to interact with the 'etcd' 'key-value' data store
<<https://github.com/coreos/etcd>>. Functions included for managing
directories, keys, nodes, and getting statistics.

Version 0.1.0

License MIT + file LICENSE

URL <https://github.com/ropensci/etseed>

BugReports <https://github.com/ropensci/etseed/issues>

LazyData true

VignetteBuilder knitr

Imports httr (>= 1.2.0), jsonlite (>= 1.0), R6 (>= 2.1.3)

Suggests roxygen2 (>= 5.0.1), knitr, testthat

RoxygenNote 5.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2016-09-08 01:16:55

R topics documented:

etseed-package	2
auth	2
etcd	3
keys	5
members	8
metrics	9
roles	10
stats	11
users	12

Index	14
--------------	-----------

etseed-package	<i>etcd client for R</i>
----------------	--------------------------

Description

etcd client for R

etcd keys

Note that all keys have to be prefixed by a forward slash (/)

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

auth	<i>etcd authentication - auth control</i>
------	---

Description

etcd authentication - auth control

Arguments

auth_user, auth_pwd
 (character) Username and password for the authenticated user, the root user
 ... Further args passed on to [GET](#), [PUT](#), or [DELETE](#)

Value

see return for each method

Methods

- `auth_status`: get authentication status, returns single boolean, TRUE indicates authentication enabled, and FALSE authentication disabled
- `auth_enable`: enable authentication, returns single boolean, TRUE indicates authentication enable request worked
- `auth_disable`: disable authentication, returns single boolean, TRUE indicates authentication disable request worked

Examples

```
## Not run:
# make a client
cli <- etcd()

# check authentication status
cli$auth_status()

# enable authentication
cli$auth_enable("root", "pickbetterpwd")

# woops, if you got error about root user, craeate one first
cli$user_add(user = "root", password = "pickbetterpwd")

# disable authentication
cli$auth_disable("root", "pickbetterpwd")

# check again, now disabled
cli$auth_status()

# Users
# Add user
cli$user_add("jane", "janepwd", "root", "pickbetterpwd")
# List users
cli$user_list()
# Get a single user
cli$user_get("root")
cli$user_get("jane")
# Delete user
cli$user_delete("jane", "root", "pickbetterpwd")

## End(Not run)
```

etcd

etseed client

Description

etseed client

Usage

```
etcd(host = "127.0.0.1", port = 2379, api_version = "v2",
      allow_redirect = TRUE, scheme = "http")
```

Arguments

host	(character) Host url. Deafault: 127.0.0.1
port	(character/numeric) Port. Default: 2379

api_version (character) etcd API version. Default: 'v2'
allow_redirect (logical) Allow redirects? Default: TRUE
scheme (character) http scheme, one of http or https. Default: http

Details

etcd creates a R6 class object. The object is not cloneable and is portable, so it can be inherited across packages without complication.

etcd is used to initialize a client that knows about your etcd instance, with options for setting host, port, etcd api version, whether to allow redirects, and the http scheme.

Value

Various output, see help files for each grouping of methods.

etcd methods

Methods

ping() ping the etcd server
version() check the etcd version
keys() list keys, see also [keys](#)
key() get a key, see also [keys](#)
create() create a key, see also [keys](#)
delete() delete a key, see also [keys](#)
update() update a key, see also [keys](#)
create_inorder() create a key in order, see also [keys](#)
metrics() see metrics, see also [metrics](#)
stats() see stats, see also [stats](#)
user_add() add a user, see also [users](#)
user_list() list users, see also [users](#)
user_get() get a user, see also [users](#)
user_delete() delete a user, see also [users](#)
member_list() list members, see also [members](#)
member_add() add a member, see also [members](#)
member_change() change a member, see also [members](#)
member_delete() delete a member, see also [members](#)
role_add() add a role, see also [roles](#)
role_list() list roles, see also [roles](#)
role_get() get a role, see also [roles](#)
role_delete() delete a role, see also [roles](#)
auth_status() authentication status, see also [auth](#)
auth_enable() enable authentication, see also [auth](#)
auth_disable() disable authentication, see also [auth](#)

Examples

```
## Not run:
# make a client
cli <- etcd()

# ping
## ping to make sure it's up
cli$ping()

# version
## get etcd version information
cli$version()

# etcd variables
cli$host
cli$port
cli$api_version
cli$allow_redirect
cli$scheme

# set a different host
etcd(host = 'stuff.com')

# set a different port
etcd(host = 3456)

# set a different etcd API version
etcd(host = 'v3')

# set a different http scheme
etcd(scheme = 'https')

# don't allow redirects
etcd(allow_redirect = FALSE)

## End(Not run)
```

keys

List a key or all keys

Description

List a key or all keys

Arguments

key	(character) A key name. Optional.
value	Any object to store. Required for create, update, and delete functions
recursive	Whether to do recursive something???

sorted	Whether to return sorted or not.
prevValue	(character) Previous value to match against.
prevIndex	(integer) Previous index to match against.
tTl	(integer) Seconds after which the key will be removed.
dir	(logical) Whether to create or delete a directory or not
wait	(logical) Whether to wait or not for a key change. Default: FALSE
wait_index	(integer) Index to wait until
...	Further args passed on to GET

Value

All return a list, with named slots 'action' and 'node'. If the key previously existed, or is deleted, or updated, you get an addition slot 'prevNode'. The action slot tells you what action you took. the 'node' slot gives the key name, its value, a 'modifiedIndex' and a 'createdIndex'

Methods

- key: get a single key
- keys: get all keys
- create: create a key
- create_in_order: create a key in order
- update: update a key
- delete: delete a key

Headers

You can get header info on requests via curl options like `key("/mykey", config = verbose())`, but make sure to load `httr` first. Headers include in particular three useful ones that provide global information about the etcd cluster that serviced a request:

- X-Etcd-Index: the current etcd index as explained above. When request is a watch on key space, X-Etcd-Index is the current etcd index when the watch starts, which means that the watched event may happen after X-Etcd-Index.
- X-Raft-Index: similar to the etcd index but is for the underlying raft protocol
- X-Raft-Term is an integer that will increase whenever an etcd master election happens in the cluster. If this number is increasing rapidly, you may need to tune the election timeout. See the tuning (<https://github.com/coreos/etcd/blob/master/Documentation/tuning.md>) section for details.

Setting a key from file

Note that setting a key from a file is not working yet.

Examples

```
## Not run:
# make a client
cli <- etcd()

# Make a key
cli$create(key="/mykey", value="this is awesome")
cli$create(key="/things", value="and stuff!")
## use ttl (expires after ttl seconds)
cli$create(key="/stuff", value="tables", ttl=10)

# Make a directory
cli$create(key="/mydir", dir = TRUE)
# List a directory
cli$key("/mydir")
# Make a key inside a directory
cli$create("/mydir/key1", value = "foo")
cli$create("/mydir/key2", value = "bar")
# List again, now with two keys
cli$key("/mydir")
# Delete a directory
cli$delete(key="/mydir", dir = TRUE)

# Update a key
cli$update(key="/things", value="and stuff! and more things")

# Create an in-order key
cli$create_inorder("/queue", "thing1")
cli$create_inorder("/queue", "thing2")
cli$create_inorder("/queue", "thing3")
cli$key("/queue", sorted = TRUE, recursive = TRUE)

# List all keys
cli$keys()
cli$keys(sorted = TRUE)
cli$keys(recursive = TRUE)
cli$keys(sorted = TRUE, recursive = TRUE)

# List a single key
cli$key("/mykey")
cli$key("/things")

# Waiting
## Wait for a change via long-polling
## in another R session, load etseed, then run the 2nd line of code
# cli$key("/anewkey", wait = TRUE)
# cli$create("/anewkey", "hey from another R session")
## Wait for change from cleared event index
# cli$key("/anewkey", wait = TRUE, wait_index = 7)

# Delete a key
cli$create("/hello", "world")
```

```

cli$delete("/hello")
## Delete only if matches previous value, fails
cli$delete("/things", prevValue="two")
## Delete only if matches previous index
### Fails
cli$delete("/things", prevIndex=1)
### Works
cli$delete("/things", prevIndex=13)

# curl options
library("httr")
cli$keys(config = verbose())

# Hidden keys
## Create a hidden key using "_" at beginning
cli$create("/_message", "my hidden key")
## A key that's not hidden
cli$create("/message", "my un-hidden key")
## Call to root directory doesn't show the hidden key
cli$keys()

# Set a key from a file
# cat("hello\nworld", file = "myfile.txt")
# cli$create("/myfile", file = file)

## End(Not run)

```

members

Manage etcd members

Description

Manage etcd members

Arguments

id	(character) A member id
newid	(logical) new member id
...	Further args passed on to GET

Details

Be careful with these commands

Value

Logical or a list, see Methods for what each returns

Methods

- `member_list`: list all members, a list of members
- `member_add`: add a member, a list of info just added
- `member_change`: change a member, may not be working quite yet
- `member_delete`: delete a member, returns nothing on success

Examples

```
## Not run:
Sys.setenv(ETSEED_USER = "root")
Sys.setenv(ETSEED_PWD = "pickbetterpwd")

# make a client
cli <- etcd()

# list members
cli$member_list()

# add a member
# cli$member_add("http://10.0.0.10:2380")

# change a member - not sure this is working...
## mms <- cli$member_list()
## cli$member_change(mms$members[[1]]$id, "http://10.0.0.10:8380", config=verbose())

# delete a member
# mms <- cli$member_list()
# cli$member_delete(mms$members[[1]]$id)

## End(Not run)
```

 metrics

Get etcd metrics

Description

Get etcd metrics

Arguments

`pretty` (logical) Print easier to read with newlines, or as a single string. Default: TRUE
 ... Further args passed on to [GET](#)

Value

Prints a human readable text representation to console of the metrics of your etcd cluster

Examples

```
## Not run:
# make a client
cli <- etcd()

cli$metrics()
cli$metrics(pretty = FALSE)

## End(Not run)
```

roles

etcd authentication - roles

Description

etcd authentication - roles

Arguments

role	(character) Role to act upon
perm_read	Read permission to set
perm_write	Write permission to set
grant_read	Grant read permission to set
grant_write	Grant write permission to set
revoke_read	Revoke read permission to set
revoke_write	Revoke write permission to set
auth_user, auth_pwd	(character) Username and password for the authenticated user, the root user
...	Further args passed on to GET , PUT , or DELETE

Value

see return for each method

Methods

- `role_list`: list roles, a list of roles
- `role_add`: add a role, returns a list on success
- `role_get`: get a role by name, a list of details for the role
- `role_delete`: delete a role by name, nothing returned on success

Examples

```
## Not run:
# make a client
cli <- etcd()

# Add role
#perms <- list(fun = list(kv = list(read = "/message/*")))
#cli$role_add(role = "fun", perm_read = perms, auth_user="root",
# auth_pwd="pickbetterpwd")

# List roles
cli$role_list()

# Get a single role
cli$role_get("root")
cli$role_get("fun")

# Delete role
cli$role_delete("fun", "root", "pickbetterpwd")

# Update a role
### FIXME - still working on this
## First, create
# cli$role_add("stuff", perm_read = "/message/*", perm_write = "/message/*",
#   auth_user = "root", auth_pwd = "pickbetterpwd")
## update
# "xxx"
## get
# cli$role_get("stuff")

## End(Not run)
```

stats

Get etcd statistics

Description

Get etcd statistics

Arguments

which (character) one of leader (default), self, or store
pretty (logical) parse text to an R list, or not. Default: TRUE
... Further args passed on to [GET](#)

Value

Prints a human readable text representation to console. If pretty = TRUE (default), returns a list, but if pretty = FALSE returns JSON data.

Examples

```
## Not run:
# make a client
cli <- etcd()

# leader stats
cli$stats()

# self stats
cli$stats("self")

# store stats
cli$stats("store")

# Pretty or not
cli$stats(pretty = TRUE)
cli$stats(pretty = FALSE)

## End(Not run)
```

 users

etcd authentication - users

Description

etcd authentication - users

Arguments

user	(character) User name to create/delete
password	(character) Password to give the new user
roles	(list/character vector) Roles to give the new user
auth_user, auth_pwd	(character) Username and password for the authenticated user, the root user
...	Further args passed on to GET , PUT , or DELETE

Value

see return for each method

Methods

- `user_list`: list users, a list of users
- `user_add`: add a user, a list with slots `user` and `roles`
- `user_get`: get a user by name, a list with slots `user` and `roles` (if any exist)
- `user_delete`: delete a user by name, returns nothing on success

Examples

```
## Not run:
# make a client
cli <- etcd()

# Add user
cli$user_add("jane", "janepwd", "root", "pickbetterpwd")

# List users
cli$user_list()

# Get a single user
cli$user_get("root")
cli$user_get("jane")

# Delete user
cli$user_delete("jane", "root", "pickbetterpwd")

## End(Not run)
```

Index

*Topic **package**

etseed-package, 2

auth, 2, 4

auth_disable (auth), 2

auth_enable (auth), 2

auth_status (auth), 2

create (keys), 5

create_in_order (keys), 5

DELETE, 2, 10, 12

delete (keys), 5

etcd, 3

etseed (etseed-package), 2

etseed-package, 2

GET, 2, 6, 8–12

key (keys), 5

keys, 4, 5

member_add (members), 8

member_change (members), 8

member_delete (members), 8

member_list (members), 8

members, 4, 8

metrics, 4, 9

PUT, 2, 10, 12

role_add (roles), 10

role_delete (roles), 10

role_get (roles), 10

role_list (roles), 10

roles, 4, 10

stats, 4, 11

update (keys), 5

user_add (users), 12

user_delete (users), 12

user_get (users), 12

user_list (users), 12

users, 4, 12