link.etcd Documentation

Release 0.4

David Delassus

September 02, 2016

Contents

link.etcd is a wrapper library to etcd, providing a middleware and a configuration driver using that middleware..

Check out the source code on Github.

Contents:

CHAPTER 1

Tutorial

This tutorial covers how to use the `Etcd` middleware and the provided configuration driver.

A functional instance of **etcd** must be running.

1.1 Configure the middleware

Configuration file for the middleware is stored in:

```
$B3J0F_CONF_DIR/link/etcd/middleware.conf
```

Here is the default configuration if nothing is specified:

```
[ETCD]
host = localhost
port = 4001
# srv_domain is not defined
version_prefix = /v2
read_timeout = 60
allow_redirect = True
protocol = http
# cert is not defined
# ca_cert is not defined
# username is not defined
# password is not defined
allow_reconnect = False
use_proxies = False
# expected_cluster_id is not defined
per_host_pool_size = 10
```

Then you can instantiate the middleware:

```
from link.etcd.middleware import EtcdMiddleware

client = EtcdMiddleware()
```

The dict protocol has been (partially) implemented to access data:

```
client['/collection'] = {
    'subcollection': [
        'item1',
        'item2'
    ]
```

```
}

# This will write:
#   - item1 in /collection/subcollection/1
#   - item2 in /collection/subcollection/2

tree = client['/collection']

# tree will contains the dict set above

del client['/collection']

# this will erase the whole tree
```

1.2 Using the configuration driver

When creating a configurable, just use the provided configuration driver:

```
from b3j0f.conf import Configurable, category, Parameter
from link.etcd.driver import EtcdConfDriver


@Configurable(
    paths='myproject/myconfigurable.conf',
    conf=category(
        'MYCONF',
        Parameter(name='myparam')
    ),
    drivers=[EtcdConfDriver()]
)
class MyConfigurable(object):
    pass
```

Then, your configuration will be stored in **etcd** at the following path:

```
/myproject/myconfigurable.conf/
/myproject/myconfigurable.conf/MYCONF
/myproject/myconfigurable.conf/MYCONF/myparam
```

CHAPTER 2

API documentation

2.1 link.etcd package

2.1.1 Submodules

2.1.2 link.etcd.driver module

class link.etcd.driver.**EtcdConfDriver**(**args*, ***kwargs*)
Bases: b3j0f.conf.driver.base.ConfDriver

Driver that reads configuration from **etcd**.

resource()

rscpaths(*path*)

2.1.3 link.etcd.middleware module

class link.etcd.middleware.**EtcdMiddleware**(**args*, ***kwargs*)
Bases: link.middleware.connectable.ConnectableMiddleware

Middleware that connects to **etcd**.

The following operations are available:

```
client = EtcdMiddleware()
client['/path'] = value
value = client['/path']
del client['/path']
'/path' in client
```

If value is a dict, then paths are created recursively. If value is a list, then items are appended to the directory.

2.1.4 Module contents

l

E

L

R