
pymemcache-client Documentation

Release 0.3.0

TUNE Inc.

Jan 28, 2018

Contents:

1	Introduction	1
2	Badges	3
3	Install	5
4	Install Memcached	7
5	Architecture	9
6	Functionality	11
7	Configuration	13
8	Dependencies	15

CHAPTER 1

Introduction

Extension of Python package [pymemcache](#) providing client configuration.

CHAPTER 2

Badges

docs	
info	
tests	
package	
other	

CHAPTER 3

Install

```
pip install pymemcache-client
```


CHAPTER 4

Install Memcached

To run examples and tests locally upon “localhost”, memcached must be installed.

Install memcached using brew

```
$ brew install memcached

==> Installing memcached
##### 100.0%
==> Pouring memcached-1.5.3.sierra.bottle.tar.gz
==> Caveats
==> Summary
```

Start memcached using brew

```
$ brew services start memcached

==> Successfully started `memcached` (label: homebrew.mxcl.memcached)
```


CHAPTER 5

Architecture

Extension of Python package `pymemcache` returning from `pymemcache.client.hash` import `HashClient` and configured using either `~/ .pymemcache.json` or `pymemcache.json`.

`class PymemcacheClient:` Prepare a Pymemcache Client using available configuration.

Configuration can be provided when creating an instance of this class by:

- User-level file: `~/.pymemcache.json`
- Application-level file: `./pymemcache.json`
- Provided as *file path* value in JSON-format to parameter `config_file`
- Provided as `dict` value in JSON-format to parameter `config`

CHAPTER 7

Configuration

This is the expected configuration that can be provided to class `PymemcacheClient` in JSON format:

```
{
  "client_type": "[OPTIONAL str, DEFAULT 'basic']",
  "servers": [
    {
      "host": "[str]",
      "port": [int]
    }
  ],
  "server_type": "[OPTIONAL str, DEFAULT 'standard']",
  "connect_timeout": [OPTIONAL float, Defaults to forever],
  "timeout": [OPTIONAL float, Defaults to forever],
  "no_delay": [OPTIONAL bool, DEFAULT false],
  "ignore_exc": [OPTIONAL bool, DEFAULT false],
  "default_noreply": [OPTIONAL bool, DEFAULT true],
  "allow_unicode_keys": [OPTIONAL bool, DEFAULT false]
}
```

- **"client_type": [OPTIONAL] Type of memcached client, DEFAULT "basic"**
 - "basic": Memcached client using `from pymemcache.client.base import Client`. A client for a single memcached server.
 - "hash": Memcached client using `from pymemcache.client.hash import HashClient`. A client for communicating with a cluster of memcached servers.
- **"servers": A list of memcached servers and requiring at least one.**
 - if "client_type" is "basic", then 1 server and only 1 is required.
 - if "client_type" is "hash", then 1 or more servers is required.
- **"server_type": Defining special needs for connect to memcached servers, DEFAULT "standard".**
 - "standard": Pooling server connections using standard HTTP.

- "elasticache": Pooling server connections using Auto Discovery through [AWS Elasticache](#).
- "connect_timeout": OPTIONAL float, seconds to wait for a connection to the memcached server. Defaults to "forever" (uses the underlying default socket timeout, which can be very long), i.e., the socket is put in blocking mode waiting to connect to memcached server.
- "timeout": OPTIONAL float, OPTIONAL float, seconds to wait for send or recv calls on the socket connected to memcached. Defaults to "forever" (uses the underlying default socket timeout, which can be very long), i.e., the socket is put in blocking mode.
- "no_delay": OPTIONAL bool, set the TCP_NODELAY flag, which may help with performance in some cases. DEFAULT false.
- "ignore_exc": OPTIONAL bool, True to cause the "get", "gets", "get_many" and "gets_many" calls to treat any errors as cache misses. DEFAULT false.
- "default_noreply": OPTIONAL bool, the DEFAULT value for 'noreply' as passed to store commands. DEFAULT true.
- "allow_unicode_keys": OPTIONAL bool, support unicode (utf8) keys. DEFAULT false.

CHAPTER 8

Dependencies

`pymemcache-client` module is built upon Python 3 and has dependencies upon several Python modules available within [Python Package Index PyPI](#).

- `pymemcache`
- `telnetlib3`
- `ujson`