gsocketpool Documentation

Release 0.1.6

Studio Ousia

Apr 11, 2017

Contents

CHAPTER 1

Introduction

gsocketpool is a simple connection pool for gevent.

Basic Usage

The following is an example to create a connection pool that communicates an echo server running on *localhost 2000*.

```
>>> from gsocketpool import Pool
>>> from gsocketpool import TcpConnection
>>>
>>> options = dict(host='localhost', port=2000)
>>> pool = Pool(TcpConnection, options)
>>>
>>> with pool.connection() as conn:
...     conn.send('hello')
...     print conn.recv()
hello
```

Implementing Protocol

Arbitrary protocols can be easily implemented by extending *Connection* class. You have to override at least three functions such as *open()*, *close()* and *is_connected()*.

TcpConnection used in the above example is also implemented as a subclass of *Connection*.

```
class TcpConnection(Connection):

    def __init__(self, host, port, lifetime=600, timeout=None):
        self._sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self._host = host
        self._port = port
```

```
        self._lifetime = lifetime
        self._timeout = timeout
        self._connected = False
        self._created = None

    @property
    def socket(self):
        return self._sock

    def open(self):
        self._sock.connect((self._host, self._port))
        if self._timeout:
            self._sock.settimeout(self._timeout)

        self._connected = True
        self._created = time.time()

    def close(self):
        if self._connected:
            self._sock.close()
            self._connected = False

    def is_connected(self):
        return self._connected

    def is_expired(self):
        if time.time() - self._created > self._lifetime:
            return True
        else:
            return False

    def send(self, data):
        assert self._connected

        self._sock.send(data)

    def recv(self, size=1024):
        assert self._connected

        return self._sock.recv(size)
```

For detailed usage, please refer to the *API reference*.

CHAPTER 2

API Reference

class gsocketpool.connection.**Connection**
A base connection class.

Arbitrary connections can be defined by extending this class.

close()
Closes the connection.

get()
Returns the raw connection.

is_connected()
Returns whether the connection has been established.

Return type bool

is_expired()
Returns whether the connection is expired.

Return type bool

open()
Opens a connection.

reconnect()
Attempts to reconnect the connection.

class gsocketpool.connection.**TcpConnection**(*host*, *port*, *lifetime=600*, *timeout=None*)
A TCP connection.

Parameters

- **host** (*str*) – Hostname.
- **port** (*int*) – Port.
- **lifetime** (*int*) – Maximum lifetime (in seconds) of the connection.
- **timeout** (*int*) – Socket timeout.

class gsocketpool.pool.**Pool**(*factory*, *options={}*, *initial_connections=0*, *max_connections=200*, *reap_expired_connections=True*, *reap_interval=180*)

Connection pool.

Usage: Communicating echo server running on localhost 2000:

```
>>>
>>> from gsocketpool import Pool
>>> from gsocketpool import TcpConnection
>>> options = dict(host='localhost', port=2000)
>>> pool = Pool(TcpConnection, options)
>>>
>>> with pool.connection() as conn:
...     conn.send('hello')
...     print conn.recv()
hello
```

Parameters

- **factory** – *Connection* class or a callable that creates *Connection* instance.
- **options** (*dict*) – (optional) Options to pass to the factory.
- **initial_connections** (*int*) – (optional) The number of connections that are initially established.
- **max_connections** (*int*) – (optional) The maximum number of connections.
- **reap_expired_connections** (*bool*) – (optional) If set to True, a background thread (greenlet) that periodically kills expired connections will be launched.
- **reap_interval** (*int*) – (optional) The interval to run to kill expired connections.

acquire(*retry=10*, *retried=0*)

Acquires a connection from the pool.

Parameters **retry** (*int*) – (optional) The maximum number of times to retry.

Returns *Connection* instance.

Raises *PoolExhaustedError*

drop(*conn*)

Removes the connection from the pool.

Parameters **conn** (Connection) – *Connection* instance.

Raises *ConnectionNotFoundError*

drop_expired()

Removes all expired connections from the pool.

Parameters **conn** (Connection) – *Connection* instance.

release(*conn*)

Releases the connection.

Parameters **conn** (Connection) – *Connection* instance.

Raises *ConnectionNotFoundError*

size

Returns the pool size.

class gsocketpool.exceptions.**PoolExhaustedError**

class gsocketpool.exceptions.**ConnectionNotFoundError**

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index