Python 操作redis模块

2017-10-19 [每日一Python](https://mp.weixin.qq.com/s?__biz=MzI2Mjg5NDA4Ng==&mid=2247483712&idx=1&sn=dd3df8dc974781c3695bc6b734ac693b&chksm=ea456629dd32ef3fe7c22468340bfcba70ffe75be3ecf2a0ba82d28dccea20cb0a3bada226fc&scene=27&key=c5e6aaaa5d1e242560363ce8b41e54ce87c4efe9b75c078bd7f3a8fdf06d9b2bcbe5461bed66301121b2c1299f8ceb6167e4861900e0a8bcd9e46df0d68aad84b3917cae358c20f0e6a954da48e106e2&ascene=7&uin=Nzk3NjIyNQ%3D%3D&devicetype=Windows+7&version=6205051a&pass_ticket=pVlB793F3uK4YJsVaPHBeSBPYMcGIImT6eadIav5PlM%3D&winzoom=1##)

1

redis

redis是一个key-value存储系统 ，通常被称为数据结构服务器，因为值（value）可以是字符串、哈希、列表、集合和有序集合等类型;

这些数据类型都支持push/pop、add/remove以及取交集并集和差集等更丰富的操作，且这些操作都是原子性的。

Python操作redis模块时，要先启动redis；

2

基本操作：

首先，实例化redis所在主机的IP和端口号，然后就可以.set()设置key和value进行存储,通过.get()读取value值：

import redis

rr=redis.Redis(host='127.0.0.1', port=6379)

rr.set('key1','value1')

rr.get('key')

redis-py通过连接池（connection pool）来管理对一个redis server的所有连接；为了避免每次建立、释放连接的开销，默认为每个Redis实例都会维护一个自己的连接池；可以直接建立一个连接池，然后作为Redis的参数，这样就可以实现多个Redis实例共享一个连接池：

import redis

pool=redis.ConnectionPool(host='127.0.0.1', port=6379)

rr = redis.Redis(connection\_pool=pool)

rr.set('key2','value2')

rr.get('key2')

redis-py默认在执行每次请求时都会创建和断开一次连接，如果想要在一次请求中指定多个命令，可以使用pipline（管道）来实现一次请求指定多个命令；

import redis

pool=redis.ConnectionPool(host='127.0.0.1', port=6379)

rr=redis.Redis(connection\_pool=pool)

pipe=rr.pipeline(transaction=True)

rr.set('key1', 'value1')

rr.set('key2', 'value2')

pip.execute()

3

发布与订阅

发布与订阅：

---创建redisdemo.py文件：

import redis

class Redispub:

    def \_\_init\_\_(self):

        self.\_\_conn = redis.Redis(host='127.0.0.1')

        self.pub\_name = 'fm12'

        self.sub\_name = 'fm12'

    def public(self,mes):

        self.\_\_conn.publish(self.sub\_name, mes)

        return True

    def subscribe(self):

        pub = self.\_\_conn.pubsub()

        pub.subscribe(self.sub\_name)

        pub.parse\_response()

        return pub

----创建pub.py文件（发布者）

from redisdemo import Redispub

pubobj =  Redispub()

pubobj.public('hello,欢迎关注每日一Python')

-----创建sub.py文件（订阅者）

from redisdemo import Redispub

subobj = Redispub()

redis\_sub = subobj.subscribe()

while True:

    mes = redis\_sub.parse\_response()

    print(mes)

    print(type(mes))

先执行redisdemo.py文件，然后执行sub.py文件进行订阅，最后执行pub.py文件发布消息，订阅者就会接收到消息；