# Redis介绍

Redis（REmote DIctionary Server，远程字典服务器） 是一个开源的key-value存储系统。Redis可基于内存也可持久化的数据库，支持分布式集群部署，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持，当前Redis的应用已经非常广泛，国内像新浪、微信、淘宝等均在使用Redis。Redis具有以下特征：

速度快：Redis使用标准C编写实现，而且将所有数据存储到内存中，所以速度非常快。官方提供的数据表明，在一个普通的机器上，Redis读写速度每秒10万次左右。

持久化：由于所有数据保持在内存中，所以对数据的更新将异步地保存到磁盘上，Redis提供了一些策略来保存数据，比如根据时间间隔或更新次数。Redis支持两种持久化方式，一种是 Snapshotting（快照）也是默认方式，另一种是Append-only file（缩写aof）的方式。

数据结构：可以将Redis看做“数据结构服务器”。目前，Redis支持5种数据结构，包括字符串、哈希表、链表、集合和有序集合。

自动操作：Redis对不同数据类型的操作是自动的，因此设置或增加key值，从一个集合中增加或删除一个元素都能安全的操作。

支持多种语言：Redis支持多种语言，诸如C#,C,C++,Object-C,Ruby, Python, PHP, Perl, Java等。

主-从复制：Redis支持简单而快速的主-从复制，具有非常快速的非阻塞首次同步，Redis会通过周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，实现master-slave(主-从)同步。官方提供了一个数据，Slave在21秒即完成了对Amazon网站10G key set的复制。

事务处理：Redis支持简单事务处理，Redis能保证一个client发起的事务中的命令可以连续的执行，而中间不会插入其他client的命令。

数据分区：将数据分布到多个Redis实例中，但这主要看客户的是否支持，目前主流编程语言都支持该功能，主要采用一致性Hash算法实现。

其它特征：Redis消息传递队列中使用（消息发布/订阅）、支持虚拟内存（vm）、管道（pipeline）。

# Haproxy介绍

HAProxy是一个使用C语言编写的自由及开放源代码软件[1]，其提供[高可用性](https://baike.baidu.com/item/%E9%AB%98%E5%8F%AF%E7%94%A8%E6%80%A7)、[负载均衡](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1)，以及基于[TCP](https://baike.baidu.com/item/TCP)和[HTTP](https://baike.baidu.com/item/HTTP)的应用程序[代理](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86)。

HAProxy实现了一种[事件驱动](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8), 单一进程模型，此模型支持非常大的并发连接数。[多进程](https://baike.baidu.com/item/%E5%A4%9A%E8%BF%9B%E7%A8%8B)或多线程模型受内存限制 、系统调度器限制以及无处不在的锁限制，很少能处理数千并发连接。[事件驱动模型](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8%E6%A8%A1%E5%9E%8B)因为在有更好的资源和时间管理的用户空间(User-Space) 实现所有这些任务，所以没有这些问题。此模型的弊端是，在多核系统上，这些程序通常扩展性较差。这就是为什么他们必须进行优化以 使每个CPU时间片(Cycle)做更多的工作。

包括 [GitHub](https://baike.baidu.com/item/GitHub)、[Bitbucket](https://baike.baidu.com/item/Bitbucket)[3]、Stack Overflow[4]、[Reddit](https://baike.baidu.com/item/Reddit)、[Tumblr](https://baike.baidu.com/item/Tumblr)、[Twitter](https://baike.baidu.com/item/Twitter)[5][6]和 [Tuenti](https://baike.baidu.com/item/Tuenti)[7]在内的知名网站，及亚马逊网络服务系统都使用了HAProxy。

Haproxy有如下特点：

1、支持两种代理模式：TCP（四层）和HTTP（七层），支持虚拟主机；  
2、能够补充Nginx的一些缺点比如Session的保持，Cookie的引导等工作。  
3、支持url检测后端的服务器出问题的检测会有很好的帮助。  
4、更多的负载均衡策略比如：动态加权轮循(Dynamic Round Robin)，加权源地址哈希(Weighted Source Hash)，加权URL哈希和加权参数哈希(Weighted Parameter Hash)已经实现。  
5、单纯从效率上来讲HAProxy更会比Nginx有更出色的负载均衡速度。  
6、HAProxy可以对Mysql进行负载均衡，对后端的DB节点进行检测和负载均衡。  
9、支持负载均衡算法：Round-robin（轮循）、Weight-round-robin（带权轮循）、source（原地址保持）、RI（请求URL）、rdp-cookie（根据cookie）。  
10、不能做Web服务器即Cache。

11、免费开源，稳定性也是非常好，单Haproxy也跑得不错，稳定性可以与LVS相媲美。

12、根据官方文档，HAProxy可以跑满10Gbps-New benchmark of HAProxy at 10 Gbps using Myricom's 10GbE NICs (Myri-10G PCI-Express)，这个作为软件级负载均衡，也是比较惊人的。

13、HAProxy可以作为MySQL、邮件或其它的非web的负载均衡。

14、自带强大的监控服务器状态的页面，这个也是我非常喜欢它的原因之一

15、HAProxy支持虚拟主机。

# Redis部署图



# 部署步骤

## Redis安装及配置

### 服务器列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **操作系统** | **IP** | **Redis访问端口** | **Sentinel访问端口** |
| 1 | CentOS 6.6 | 192.168.18.159 | 7004(master) | 7001 |
| 2 | CentOS 6.6 | 192.168.18.160 | 7005(slave) | 7002 |
| 3 | CentOS 6.6 | 192.168.18.161 | 7006(slave) | 7003 |

### Redis安装文件

redis-4.0.8.tar.gz

### Redis安装配置步骤

192.168.18.159服务器安装配置redis。

1.用root用户登录服务器

2.新建/home/software/redis目录，存放redis安装包

mkdir –p /home/software/redis

3.把安装包拷贝到/home/software/redis目录下，并进入到该目录

cd /home/software/redis

4.解压安装包

tar zxf redis-4.0.8.tar.gz

5.新建/data目录，作为redis安装目录

mkdir /data

6.把安装包解压后的目录拷贝到/data目录下

cp -a /home/software/redis/redis-4.0.8 /data/

7.进入到/data/redis-4.0.8目录

cd /data/redis-4.0.8

8.开始安装

make

9.安装完毕后，新建/data/redis-4.0.8/slave/redis\_01/data目录，存放redis节点文件

mkdir -p /data/redis-4.0.8/slave/redis\_01/data

10.拷贝/data/redis-4.0.8/redis.conf到/data/redis-4.0.8/slave/redis\_01/目录下

cp -a /data/redis-4.0.8/redis.conf /data/redis-4.0.8/slave/redis\_01/

11.修改/data/redis-4.0.8/slave/redis\_01/redis.conf文件，改成主节点，主要配置点见下

#外部访问地址

bind 192.168.18.159

#外部访问端口

port 7004

#后台运行

daemonize yes

#pid文件存储位置

pidfile "/data/redis-4.0.8/slave/redis\_01/redis\_7004.pid"

#redis节点日志文件存储位置

logfile "/data/redis-4.0.8/slave/redis\_01/redis\_7004.log"

#redis节点数据文件存储位置

dir "/data/redis-4.0.8/slave/redis\_01/data"

#访问密码

requirepass "1234.com"

12.新建/data/redis-4.0.8/sentinel目录，存放sentinel节点文件

mkdir -p /data/redis-4.0.8/sentinel

13.拷贝/data/redis-4.0.8/sentinel.conf到/data/redis-4.0.8/sentinel/目录下

cp -a /data/redis-4.0.8/sentinel.conf /data/redis-4.0.8/sentinel/

14.修改/data/redis-4.0.8/sentinel/sentinel.conf文件，监控redis节点，主要配置见下

#关闭保护模式

protected-mode no

#访问端口

port 7001

#后台运行

daemonize yes

#日志存放位置

logfile "/data/redis-4.0.8/sentinel/sentinel\_7001.log"

#设置redis master名称及IP、端口，有2个sentinel认为master不可用，则redis主节点不可用

sentinel monitor mymaster 192.168.18.159 7004 2

#redis主节点访问密码

sentinel auth-pass mymaster 1234.com

192.168.18.160、161服务器安装配置redis

192.168.18.160、161两台服务器安装redis的方式和192.168.18.159一致，只是配置redis slave节点和sentinel时稍有区别，区别见下。

redis.conf区别

修改redis.conf中bind、port、pidfile、logfile、dir参数，按照所在服务器填写

新增

#master节点密码

masterauth 1234.com

#绑定master节点的IP和端口

slaveof 192.168.18.159 7004

sentinel.conf区别：

修改port参数

## HAProxy安装及配置

### 服务器列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **操作系统** | **IP** | **tcp端口** | **备注** |
| 1 | CentOS 6.6 | 192.168.18.162 | 9999 |  |

### HAProxy安装文件

haproxy-1.7.0.tar.gz

### HAProxy安装置步骤

192.168.18.162服务器安装配置HAProxy。

1.使用root用户登录服务器

2.新建/home/software/haproxy目录，存放HAProxy安装包

mkdir -p /home/software/haproxy

3.把HAProxy安装包传到服务器/home/software/haproxy目录下，并进入该目录

cd /home/software/haproxy

4.解压安装包

tar zxf haproxy-1.7.0.tar.gz

5.进入到/home/software/haproxy/haproxy-1.7.0目录

cd /home/software/haproxy/haproxy-1.7.0

6.开始第一步安装

make TARGET=linux2628 ARCH=x86\_64 PREFIX=/usr/local/haproxy

说明：通过使用 uname -r 命令查看系统内核，例如：2.6.32-504.el6.x86\_64，如果内核是2.6.28以上的，配置TARGET=linux2628；ARCH=x86\_64为系统位数

7.开始第二步安装

make install PREFIX=/usr/local/haproxy

8.创建/usr/local/haproxy/haproxy.cfg

9.haproxy.cfg中，内容及重要配置说明见下

global

daemon

#group haproxy

log /dev/log local0

maxconn 16000

pidfile /var/run/haproxy.pid

#stats socket /var/lib/haproxy/stats

tune.bufsize 32768

#user haproxy

defaults

log global

maxconn 8000

mode http

option redispatch

retries 3

stats enable

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout check 10s

listen stats

bind 0.0.0.0:10000

mode http

stats enable

stats uri /

stats refresh 5s

stats show-node

stats show-legends

stats hide-version

listen awredis

#访问地址

bind 192.168.18.162:9999

#连接数最低的服务器优先接收

balance leastconn

#模式

mode tcp

#允许访问白名单，其中white\_ip\_list为白名单文件

tcp-request connection accept if { src -f /usr/local/haproxy/white\_ip\_list }

#白名单之外的源IP不允许访问

tcp-request connection reject

#redis 健康检查，确保只有master提供连接

option tcp-check

tcp-check connect

tcp-check send AUTH\ 1234.com\r\n

tcp-check send PING\r\n

tcp-check expect string +PONG

tcp-check send info\ replication\r\n

tcp-check expect string role:master

tcp-check send QUIT\r\n

tcp-check expect string +OK

#绑定redis节点IP和端口

server redis01 192.168.18.159:7004 check port 7004 inter 5s fastinter 2s downinter 5s rise 3 fall 3

server redis02 192.168.18.160:7005 check port 7005 inter 5s fastinter 2s downinter 5s rise 3 fall 3

server redis03 192.168.18.161:7006 check port 7006 inter 5s fastinter 2s downinter 5s rise 3 fall 3

10.创建白名单

cd /usr/local/haproxy && vi white\_ip\_list

说明，白名单中的IP，按照每行一个的方式填写。

## 启动步骤

1.使用root登录192.168.18.159服务器

2.进入到/data/redis-4.0.8/src目录

cd /data/redis-4.0.8/src

3.启动节点

./redis-server /data/redis-4.0.8/slave/redis\_01/redis.conf

4.启动sentinel

./redis-sentinel /data/redis-4.0.8/sentinel/sentinel.conf

5. 使用root登录192.168.18.160服务器

6. 进入到/data/redis-4.0.8/src目录

cd /data/redis-4.0.8/src

7. 启动节点

./redis-server /data/redis-4.0.8/slave/redis\_02/redis.conf

8.启动sentinel

./redis-sentinel /data/redis-4.0.8/sentinel/sentinel.conf

9. 使用root登录192.168.18.161服务器

10. 进入到/data/redis-4.0.8/src目录

cd /data/redis-4.0.8/src

11. 启动节点

./redis-server /data/redis-4.0.8/slave/redis\_03/redis.conf

12. 启动sentinel

./redis-sentinel /data/redis-4.0.8/sentinel/sentinel.conf

13.启动HAProxy

/usr/local/haproxy/sbin/haproxy -f /usr/local/haproxy/haproxy.cfg

# 代码示例

## Java代码示例

### 主要jar包目录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **Jar包** | **版本** | **作用** | **备注** |
| 1 | spring-data-redis | 1.8.4.RELEASE | 操作redis数据库 |  |
| 2 | jedis | 2.9.0 | 封装redis的java客户端 |  |
| 3 | spring-core | 4.3.14.RELEASE | Spring核心模块 |  |
| 4 | spring-beans | 4.3.14.RELEASE | 管理对象 |  |
| 5 | spring-context | 4.3.14.RELEASE | Spring核心扩展 |  |
| 6 | spring-aop | 4.3.14.RELEASE | AOP特性 |  |

### 框架配置

redis.properties文件，配置redis连接池，内容见下：

#HAProxy访问IP

redis.host=192.168.18.162

# HAProxy访问端口

redis.port=9999

# redis密码

redis.pass=1234.com

#最多可空闲连接

redis.maxIdle=300

#连接池支持的最大连接数

redis.maxActive=600

#请求等待时间

redis.maxWait=1000

#在borrow一个jedis实例时，是否提前进行alidate操作；如果为true，则得到的jedis实例均是可用的

redis.testOnBorrow=true

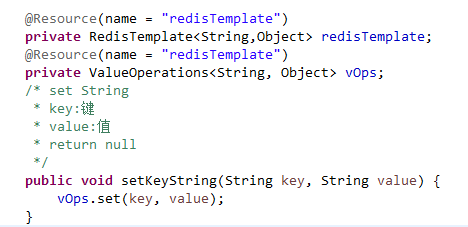
spring.xml文件，加载spring、redis配置，主要内容见下：



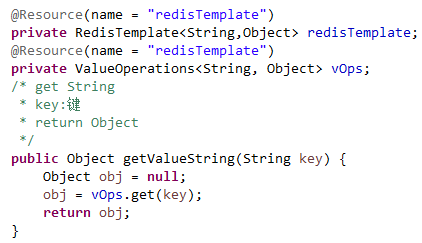
### 代码示例

#### 操作字符串

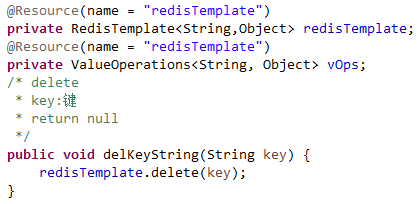
新增key-value键值对



获取key对应的value值

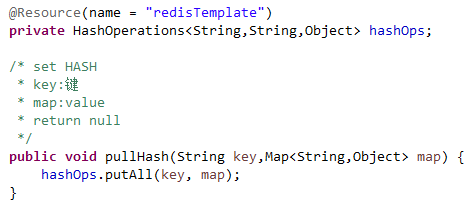


删除key-value键值对

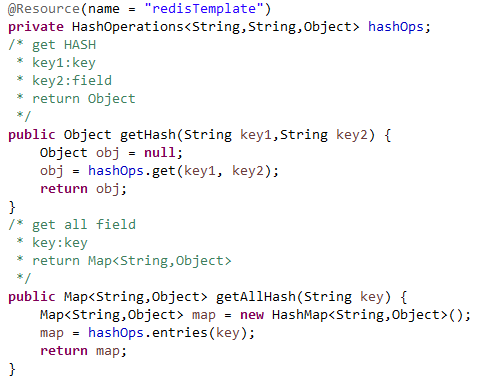


#### 操作HASH

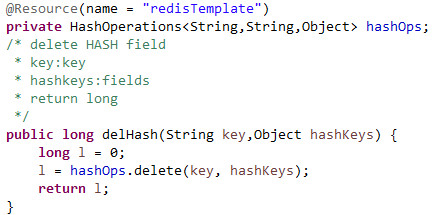
存储key-value



获取key对应的value

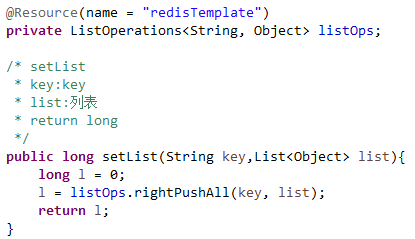


删除HASH filed

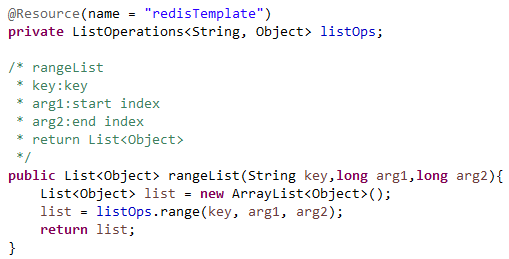


#### 操作列表

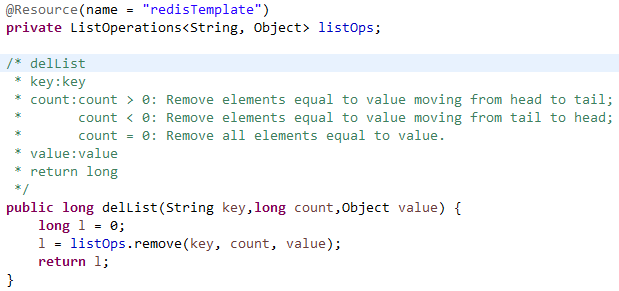
存储key-value



获取list

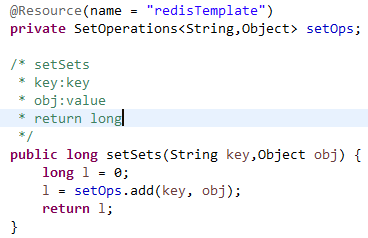


删除list

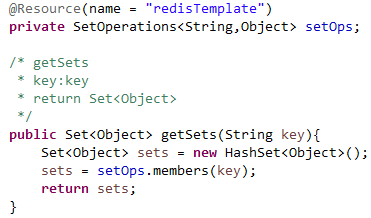


#### 操作集合

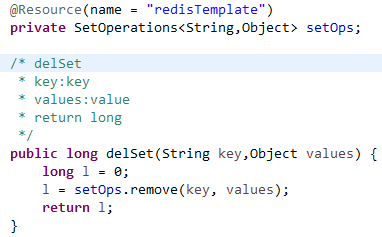
存储key-value



获取集合

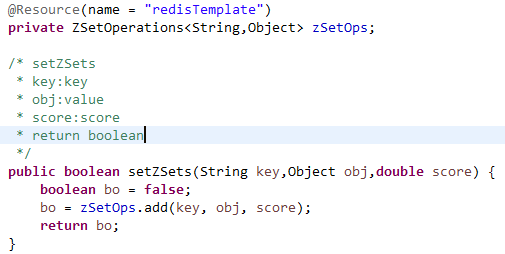


删除集合

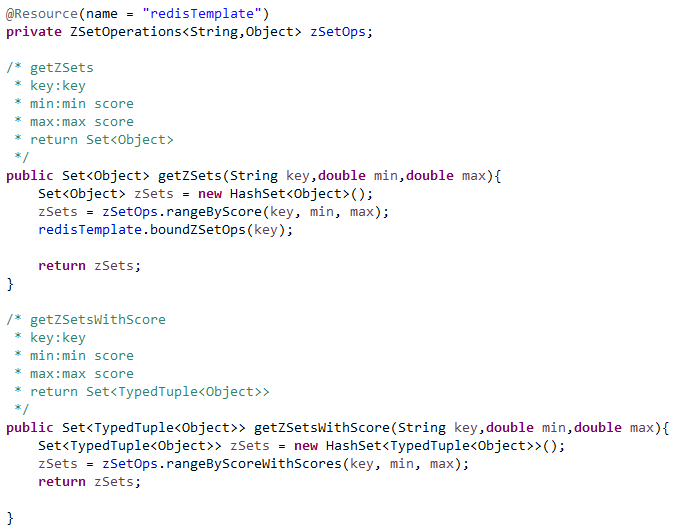


#### 操作有序集合

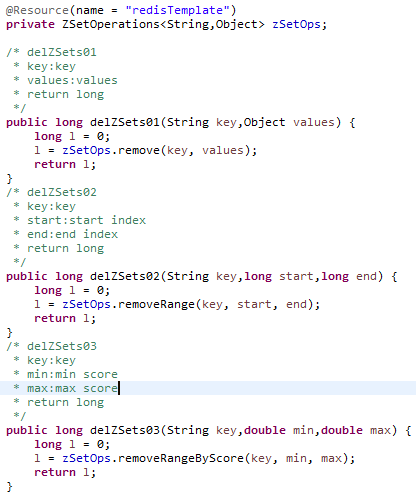
新增有序集合



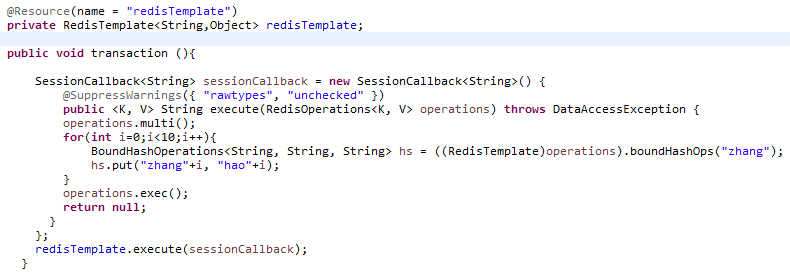
获取有序集合



删除有序集合



#### 事务



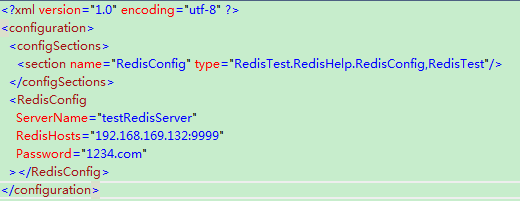
## C#代码示例

### 主要类库目录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | 类库 | **版本** | **作用** | **备注** |
| 1 | StackExchange.Redis | 1.2.6 | 封装redis的.net客户端 |  |

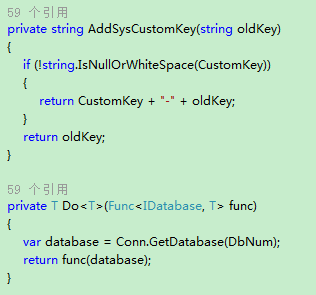
### 框架配置

Web.config文件：



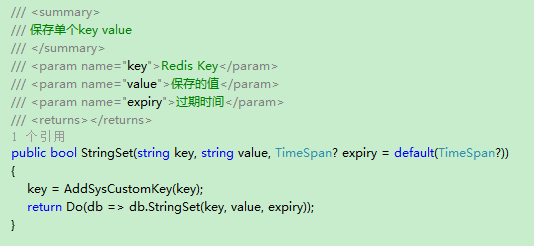
### 代码示例

#### 通用辅助方法

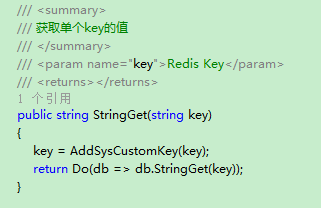


#### 操作字符串

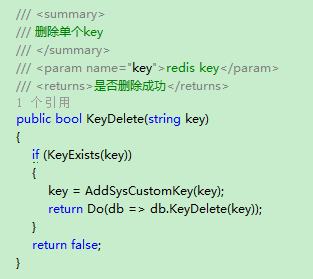
新增key-value键值对



获取key对应的value值



删除key-value键值对

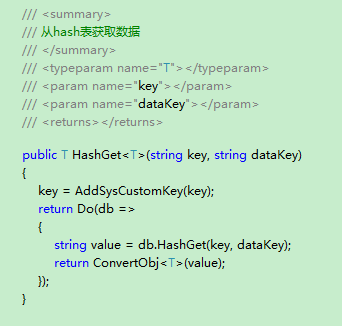


#### 操作HASH

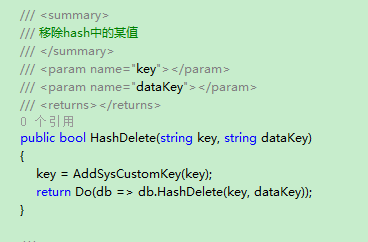
存储key-value



获取key对应的value

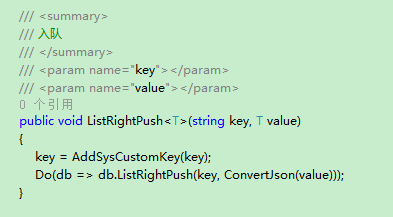


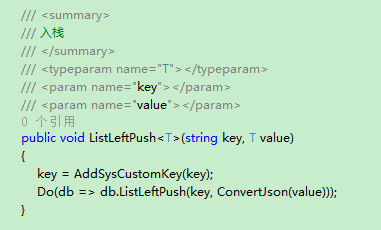
删除HASH filed



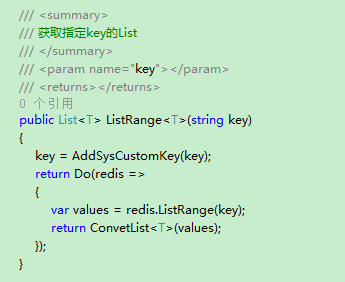
#### 操作集合

存储key-value

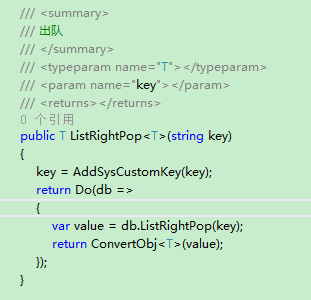


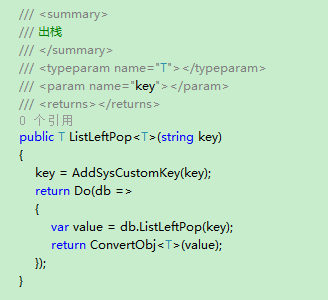


获取集合



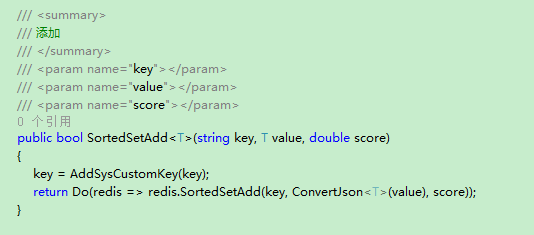
删除集合





#### 操作有序集合

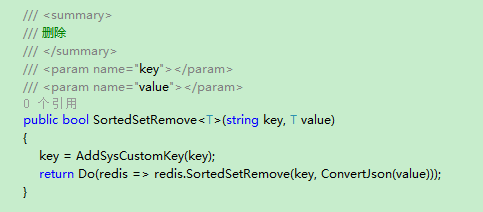
新增有序集合



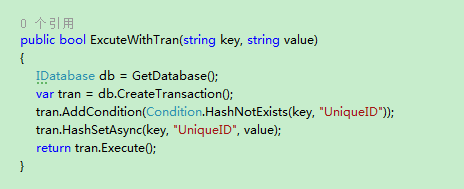
获取有序集合



删除有序集合



#### 事务



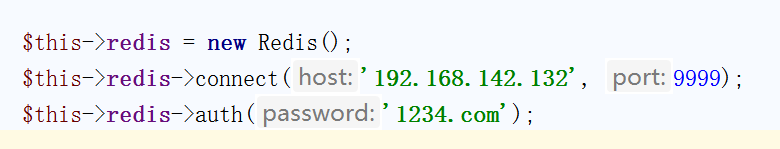
## PHP代码示例

### 主要php扩展包

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **扩展** | **版本** | **作用** | **备注** |
| 1 | php\_redis | 2.7-5.5 | 操作redis数据库 |  |
| 2 | php\_igbinary | 1.2.1-5,5 | 操作redis数据库 |  |

### 框架配置

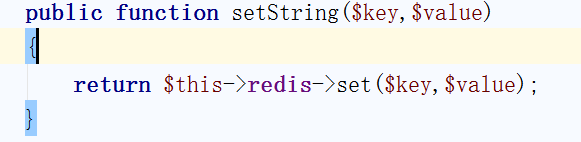
php连接haproxy



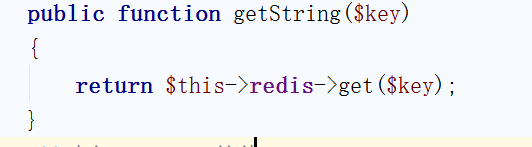
### 代码示例

#### 操作字符串

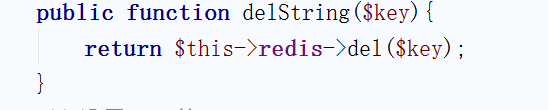
设置key-value



获取key的值

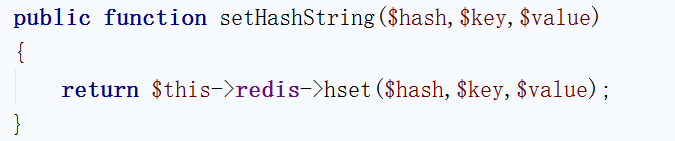


删除key-value

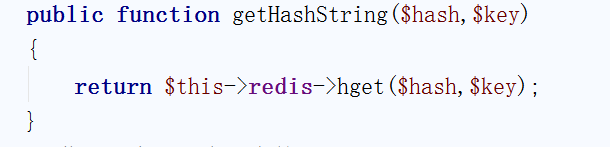


#### 操作Hash

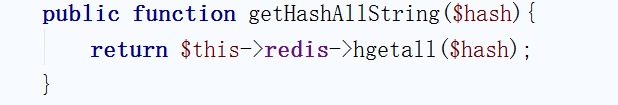
设置key-value



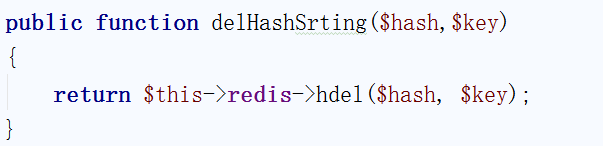
获取Hash中某个key的值



获取一个Hash中所有key的值

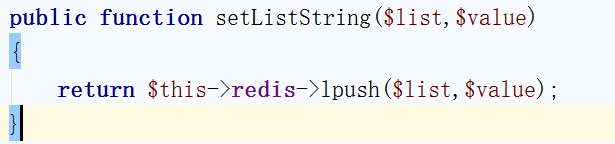


删除hash中一个key的值

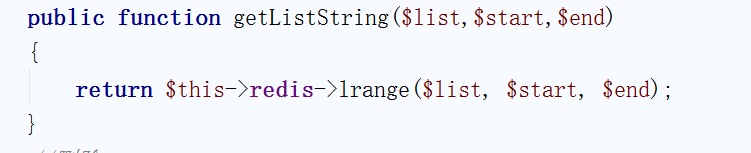


#### 操作List

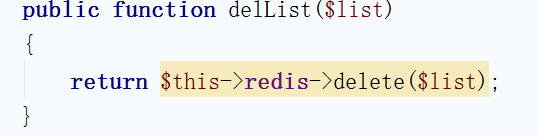
存储数据到list



返回key中从start到end位置间的元素,如果获取所有的元素，start=0,end=-1

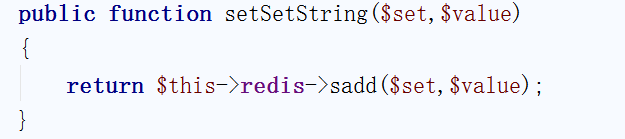


删除list

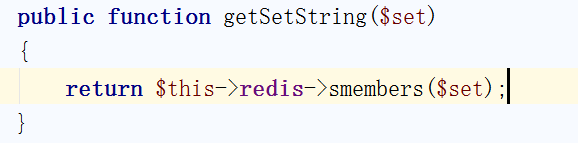


#### 操作集合

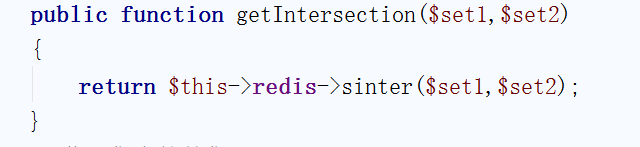
存储数据到集合中



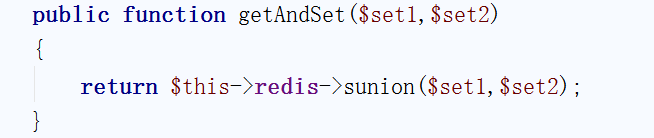
获取集合中的数据



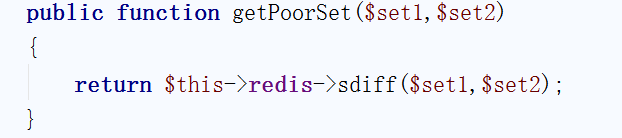
交集



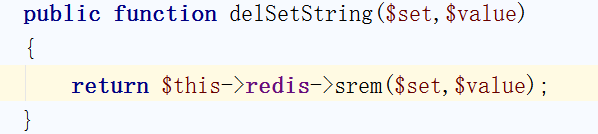
并集



差集

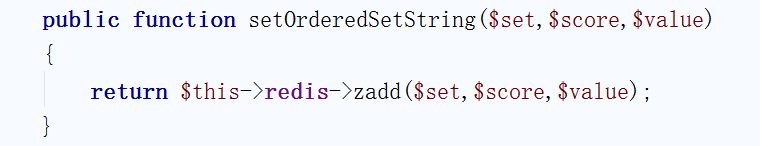


删除集合中指定元素

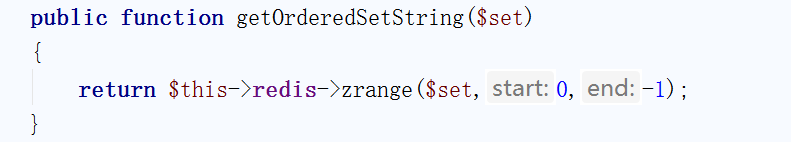


#### 操作有序集合

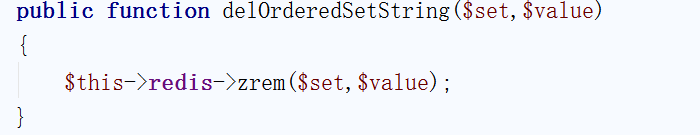
存储数据到有序集合中



返回有序集合中所有元素



删除有序集合中指定成员



#### 事务

