# Redis

## 概要

spring 封装了 RedisTemplate 对象来进行对redis的各种操作，它支持所有的 redis 原生的 api。

**RedisTemplate中定义了对5种数据结构操作**

redisTemplate.opsForValue();//操作字符串

redisTemplate.opsForHash();//操作hash

redisTemplate.opsForList();//操作list

redisTemplate.opsForSet();//操作set

redisTemplate.opsForZSet();//操作有序set

**序列话实体类**：当你想把的内存中的对象写入到硬盘的时候。当你想用套接字在网络上传送对象的时候。 当你想通过RMI传输对象的时候。

## 数据同步问题

redis和mysql数据的同步，代码级别大致可以这样做：

读: 读redis->没有，读mysql->把mysql数据写回redis

写: 写mysql->成功，写redis

## Redis的持久化

[Redis](http://lib.csdn.net/base/redis)强大的功能很大部分是由于他把数据缓存在内存中，为了使[redis](http://lib.csdn.net/base/redis)在重启的时候，数据不丢失，就需要已某种方式把数据持久化到磁盘中。Redis持久化的方式有俩种，RDB和AOF。

<https://my.oschina.net/u/2286631/blog/1057221>

### RDB快照方式

允许你每隔一段时间对内存数据做一次快照然后存储到硬盘中。该方式是Redis默认的持久化方式。

RDB可以通过在配置文件中配置时间或者改动键的个数来定义快照条件，编辑配置文件redis.conf，找到

save 900 1 #15分钟之内至少有一个建被更改则进行快照

save 300 10 #5分钟之内至少有10个建被更改则进行快照

save 60 10000 #1分钟之内至少有1000个建被更改则进行快照

他们之间是或的关系，RDB持久化到磁盘的文件默认路径是在当前目录，文件名为dump.rdb，你可以通过配置文件配置dir和dbfilename来指定文件目录和文件名称，RDB文件还可以进行压缩，你可以通过配置rdbcompression参数来进行压缩。

### AOF持久化

默认是关闭的，默认是打开RDB持久化

AOF：通过将发送到服务器的写操作命令记录下来，形成AOF文件，文件默认名称是appendonly.aof，可以通过appendfilename来指定文件名称。你可以通过配置文件打开AOF功能

appendonly yes

AOF的原理是直接把用户插入到服务器的命令追加到结尾，那么文件会原来越大，一些重复的写命令也会越来越多，这时，我们可以利用BGREWRITEAOF 命令来重写AOF，重写的配置如下：

auto-aof-rewrite-percentage 100 #aof文件大小超过上次重写时文件大小的百分之几开始重写，如果之前没有写过，则根据启动时文件大小。

auto-aof-rewrite-min-size 64mb #限制允许重写时的最小文件大小。

AOF在同步内存数据到磁盘上时，并不是马上把文件写如到磁盘中，而是先把文件缓存到系统，然后每隔30秒将文件写入到磁盘中，我们可以在配置文件中配置同步的策略

appendfsync always #每次都同步，保证数据不会丢失，但会慢

appendfsync everysec #每秒同步，系统默认同步策略

appendfsync no #不主动同步，由操作系统决定，快，但数据容易丢失

AOF rewrite

redis中的数据其实有限的，很多数据可能会自动过期，可能会被用户删除，可能会被redis用缓存清除的算法清理掉。redis中的数据会不断淘汰掉旧的，就一部分常用的数据会被自动保留在redis内存中，所以可能很多之前的已经被清理掉的数据，对应的写日志还停留在AOF中，AOF日志文件就一个，会不断的膨胀，到很大很大，所以AOF会自动在后台每隔一定时间做rewrite操作，比如日志里已经存放了针对100w数据的写日志了; redis内存只剩下10万; 基于内存中当前的10万数据构建一套最新的日志，到AOF中; 覆盖之前的老日志; 确保AOF日志文件不会过大，保持跟redis内存数据量一致。redis 2.4之前，还需要手动，开发一些脚本，crontab，通过BGREWRITEAOF命令去执行AOF rewrite，但是redis 2.4之后，会自动进行rewrite操作在redis.conf中，可以配置rewrite策略

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

比如说上一次AOF rewrite之后，是128mb。然后就会接着128mb继续写AOF的日志，如果发现增长的比例，超过了之前的100%，256mb，就可能会去触发一次rewrite但是此时还要去跟min-size，64mb去比较，256mb > 64mb，才会去触发rewrite

### AOF破损文件的修复

     如果redis在append数据到AOF文件时，机器宕机了，可能会导致AOF文件破损用redis-check-aof --fix命令来修复破损的AOF文件。

### AOF和RDB同时工作

* 1）如果RDB在执行snapshotting操作，那么redis不会执行AOF rewrite; 如果redis再执行AOF rewrite，那么就不会执行RDB snapshotting
* （2）如果RDB在执行snapshotting，此时用户执行BGREWRITEAOF命令，那么等RDB快照生成之后，才会去执行AOF rewrite
* （3）同时有RDB snapshot文件和AOF日志文件，那么redis重启的时候，会优先使用AOF进行数据恢复，因为其中的日志更完整

## Redis Windows版本的使用

<http://www.cnblogs.com/Luwak90/p/9412492.html>

### 启动redis

切换到redis目录下运行redis-server.exe redis.windows.conf

这时候另启一个cmd窗口，原来的不要关闭，不然就无法访问服务端了。

切换到redis目录下运行 redis-cli.exe 。

C:\redis>redis-cli.exe

127.0.0.1:6379> ping

PONG

以上说明我们已经成功安装了redis。

## redis的过期策略以及内存淘汰机制

maxmemory-policy allkeys-lru

当内存不足以容纳新写入数据时，在键空间中，移除最近最少使用的key。推荐使用，目前项目在用这种。

## Redis适用场景

### String

String数据结构是简单的key-value类型，value其实不仅可以是String，也可以是数字。   
常规key-value缓存应用；   
**使用场景：**常规key-value缓存应用。常规计数: 微博数, 粉丝数。

### Hash

**常用命令：**hget,hset,hgetall 等

Redis hash是一个string类型的field和value的映射表，hash特别适合用于存储对象。 通过 key(用户ID) + field(属性标签) 就可以操作对应属性数据了，既不需要重复存储数据，也不会带来序列化和并发修改控制的问题  
**使用场景：**存储部分变更数据，如用户信息等。

### List

**常用命令：**lpush,rpush,lpop,rpop,lrange等。

List即可以作为栈，也可以作为队列。

**应用场景：**

Redis list的应用场景非常多，也是Redis最重要的数据结构之一，比如twitter的关注列表，粉丝列表等都可以用Redis的list结构来实现。

list就是链表，略有数据结构知识的人都应该能理解其结构。使用Lists结构，我们可以轻松地实现最新消息排行等功能。List的另一个应用就是消息队列，可以利用List的PUSH操作，将任务存在List中，然后工作线程再用POP操作将任务取出进行执行。Redis还提供了操作List中某一段的api，你可以直接查询，删除List中某一段的元素。   
Redis的list是每个子元素都是String类型的双向链表，可以通过push和pop操作从列表的头部或者尾部添加或者删除元素，这样List即可以作为栈，也可以作为队列。

### Set

**常用命令：**

sadd,spop,smembers,sunion 等

**应用场景：**

     Redis set对外提供的功能与list类似是一个列表的功能，特殊之处在于set是可以自动排重的，当你需要存储一个列表数据，又不希望出现重复数据时，set是一个很好的选择，并且set提供了判断某个成员是否在一个set集合内的重要接口，这个也是list所不能提供的。

Set 就是一个集合，集合的概念就是一堆不重复值的组合。利用Redis提供的Set数据结构，可以存储一些集合性的数据。

Set是集合，是String类型的无序集合，set是通过hashtable实现的，概念和数学中个的集合基本类似，可以交集，并集，差集等等，set中的元素是没有顺序的获取某段时间所有数据去重值   
使用场景：

交集，并集，差集：(Set)

获取某段时间所有数据去重值

### Sorcted Set

**常用命令：**

zadd,zrange,zrem,zcard等

  Redis sorted set的使用场景与set类似，区别是set不是自动有序的，而sorted set可以通过用户额外提供一个优先级(score)的参数来为成员排序，并且是插入有序的，即自动排序。当你需要一个有序的并且不重复的集合列表，那么可以选择sorted set数据结构，比如twitter 的public timeline可以以发表时间作为score来存储，这样获取时就是自动按时间排好序的。

**Redis实际应用场景**

* **显示最新的项目列表**
* **排行榜应用，取TOP N操作**

### Pub/sub

ub/Sub 从字面上理解就是发布（Publish）与订阅（Subscribe），在Redis中，你可以设定对某一个key值进行消息发布及消息订阅，当一个key值上进行了消息发布后，所有订阅它的客户端都会收到相应的消息。这一功能最明显的用法就是用作实时消息系统，比如普通的即时聊天，群聊等功能。

使用场景

Pub/Sub构建实时消息系统

Redis的Pub/Sub系统可以构建实时的消息系统   
比如很多用Pub/Sub构建的实时聊天系统的例子。

## Redis远程连接

<https://blog.csdn.net/pingweicheng/article/details/81086466>

* **步骤一：注释掉redis.window.conf文件中的bind属性设置。**
* **步骤二：把protected-mode属性设置no**

redis-cli -h 47.105.102.121 -p 6379

-h 选择指定服务端ip

-p 选择指定端口。

## 密码设置

requirepass actorYKTRedis（密码）

requirepass yourpassword //此处注意，行前不能有空格

重新设置密码后，重新登录才能获取操作权限

redis-cli -h 47.105.102.121 -p 6379 -a actorYKTRedis //需添加密码参数

## ****注意事项：****

不能直接双击redis安装目录下bin目录下的redis-server.exe启动redis服务端，否则会导致无法通过远程客户端进行操作。

ActiveMQ