# Redis

## redis的数据类型

### redis的通用命令

1. keys \* 查询所有的键
2. type key 查询键的类型
3. del key 删除键

### redis的持久化

1. redis是一个内存数据库，当redis服务器重启，或者电脑重启，数据会丢失，我们可以将redis内存中的数据持久化保存到硬盘的文件中。
2. Redis持久化机制：
3. RDB：默认方式，不需要进行配置，默认就使用这种机制

\* 在一定的间隔时间中，检测key的变化情况，然后持久化数据，持久化文件 dump.rdb

1. 编辑redis.conf文件

save 900 1

# after 900 sec (15 min) if at least 1 key changed

save 300 10

# after 300 sec (5 min) if at least 10 keys changed

save 60 10000

# after 60 sec if at least 10000 keys changed

1. 重新启动redis，并指定配置文件名称

命令行打开，命令：redis-server.exe redis.conf

1. 触发rdb的操作

4）rdb文件的写入原理

1. AOF：日志记录的方式，可以记录每一条命令的操作。可以咋IMEI一次命令后，持久化数据。持久化文件 appendonly.aof
2. 编辑redis.conf配置文件

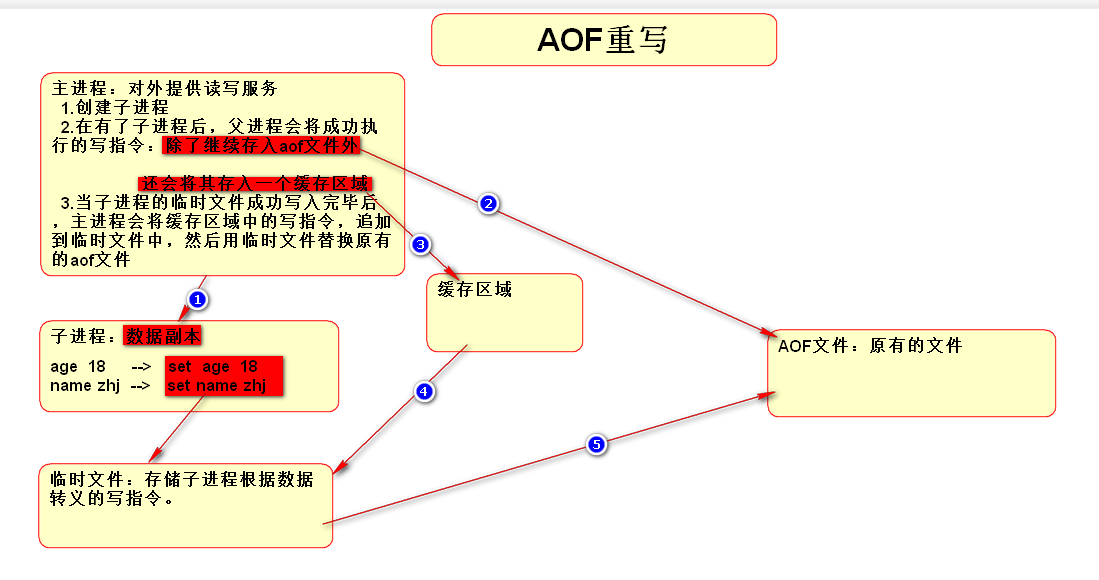
appendonly no（关闭aof） -----》appendonly yes

# appendfsync always 每一次操作都进行持久化

appendfsync everysec 每隔一秒进行一次持久化

# appendfsync no 不进行持久化

1. 重新启动redis，并指定配置文件名称命令行打开，命令：redis-server.exe redis.conf
2. 触发aof重写的操作
3. Aof文件的写入原理



### Java客户端 Jedis

\*Jedis：一款java操作redis数据库的工具

\*使用步骤：

\*下载Jedis的jar包

\*使用java API操作redis

// 1.获取连接

Jedis jedis = new Jedis(“ip”,”port”); // 空参默认“localhost”“6379”

//2.操作

jedis.set(“key”,”value”);

//3.关闭连接

jedis.close();

\*Jedis操作各种数据结构

1. 字符串类型 String

set ：jedis.set("username","zhnagsan");

get ：jedis.get("username");

jedis.setex("activecode",20,"hehe"); //将activecode：hehe键值对存入redis，并且20秒后自动删除该键值对；

1. 哈希类型 Hash ： map格式

hset: jedis.hset(“user”,”name”,”lisi”);

jedis.hset(“user”,”age”,”18”);

jedis.hset(“user”,”gender”,”male”);

hget: String name = jedis.hget(“user”,”name”);

hgetAll: Map<String ,String> user = jedis.hgetAll(“user”);

1. 列表类型list ：linkedList格式，支持重复元素

Lpush/rpush: jedis.lpush("mylist","a","b","c"); // 从左边存 cba

jedis.rpush("mylist","a","b","c");// 从右边存 abc

Rpop/lpop: String element1 = jedis.lpop(“mylist”); // c

String element2 = jedis.rpop(“mylist”);// c

Lrange start end:(范围获取)

jedis.lrange("mylist",0,-1);

1. 集合类型：set： 不允许重复元素

sadd : jedis.sadd(“myset”,”java”,”php”,”c”);

smembers:Set<String> myset = jedis.smembers(“myset”);

1. 有序集合类型 sortedset： 不允许重复元素，且元素有顺序

zadd: jedis.zadd(“mysortedset”,3,”yasyuo”);

jedis.zadd(“mysortedset”,7,”liqin”);

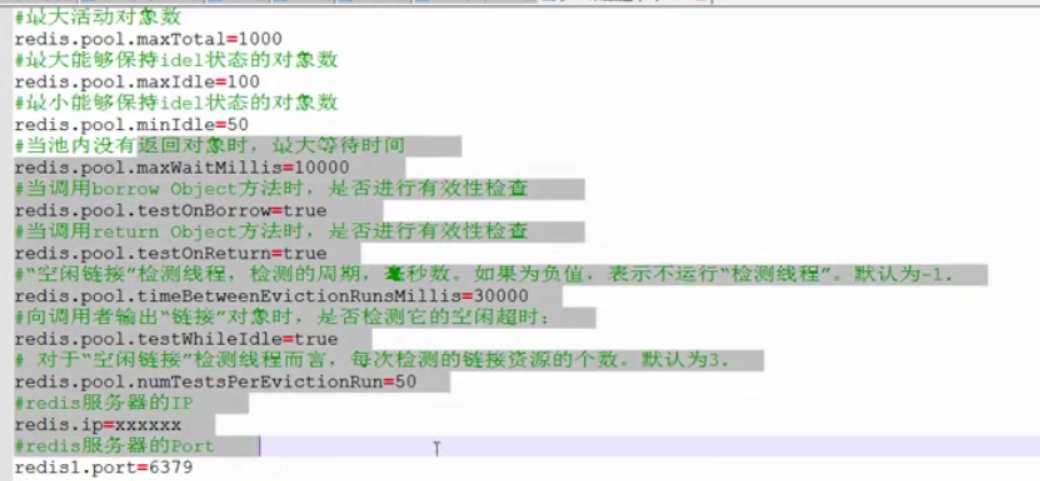
jedis.zadd(“mysortedset”,5,”vn”);

Set<String> mysortedset = jedis.zrange(“mysortedset”,0,-1); // 默认按分数升序

### Jedis连接池：JedisPool

\*使用：

1. 创建JedisPool连接池对象
2. 调用方法 个体Resource()方法获取Jedis连接



//0.创建一个配置对象

JedisPoolConfig config = new JedisPoolConfig （）；

config.setMaxTotal(50);

config.setMaxIdle(10);

// 1.创建JedisPool对象

JedisPool jedisPool = new JedisPool（）； // 默认 localhost 6379

JedisPool jedisPool = new JedisPool（config，”localhost”,6379）；

// 2.获取连接

Jedis jedis = jedisPool.getResource();

//3.使用

//4.关闭（归还到连接池中）

jedis.close();

\*JedisPool工具类 JedisPoolUtils

加载配置文件，配置连接池的参数，提供获取连接的方法

|  |  |
| --- | --- |
| |  | | --- | | public class JedisPoolUtils {  private static JedisPool *jedisPool*;   static {  // 读取配置文件  InputStream is = JedisPoolUtils.class.getClassLoader()  .getResourceAsStream("jedis.properties");  // 创建Properties对象  Properties properties new Properties():  // 关联文件  try{  properties.load(is);  }catch (IOException exception){  exception.printStackTrace();  }  // 获取数据，设置到JedisPoolConfig中  JedisPoolConfig config = new JedisPoolConfig();  config.setMaxTotal(Integer.parseInt(paproperties.getProperties("maxTotal")));  config.setMaxIdle(Integer.parseInt(paproperties.getProperties("maxTotal")));   // 初始化JedisPool  *jedisPool* = new JedisPool(config,properties.getProperty("host"),Integer.parseInt(properties.getProperty("port")));    }  */\*\*  \* 获取连接方法  \*/* public static Jedis getJedis(){  return *jedisPool*.getResource():  } | |

### 案例

利用redis作为缓存

\*注意：使用redis缓存一些不经常发生变化的数据。

\*数据库的数据一旦发生改变，则需要更新缓存；

\*数据库的表执行 增删改的相关操作，需要将redis缓存数据情况，再次存入

\*在service对应的增删改方法中，将redis数据删除。

\*Redis作为mabatis的二级缓存原理：

