



# Redis 内存性数据库



# 1. Redis 简介：完全开源免费的高性能的 key-value 数据库。

- 支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用；
- 不仅仅支持简单的 key-value 类型的数据，同时还提供 list，set，zset，hash 等数据结构的存储；
- 支持数据的备份，即 master-slave 模式的数据备份；
- 性能极高，Redis 能读的速度是 110000 次 /s，写的速度是 81000 次 /s；
- Redis 的所有操作都是原子性的，意思就是要么成功执行要么失败完全不执行。单个操作是原子性的。多个操作也支持事务，即原子性，通过 MULTI 和 EXEC 指令包起来；
- 支持 publish/subscribe, 通知, key 过期等等特性。



## 2. 进入下载页

<https://github.com/microsoftarchive/redis/releases> ,

The screenshot shows the GitHub repository page for `microsoftarchive/redis`. The repository is a fork of `antirez/redis`. The page displays the release page for version `3.2.100`, which is marked as a pre-release. The release was created by `enricogior` on July 1, 2016. The release notes state that this is the first release of Redis on Windows 3.2, based on `antirez/redis/3.2.1` with some Windows-specific fixes. It is noted that the release has passed standard tests but has not been tested in a production environment. The release includes four assets: `Redis-x64-3.2.100.msi` (5.8 MB), `Redis-x64-3.2.100.zip` (4.98 MB), `Source code (zip)`, and `Source code (tar.gz)`.

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

🔗 microsoftarchive / redis  
forked from antirez/redis

👁 Watch 1.4k ★ Star 16.2k 🍴 Fork 16.1k

🔗 Code 📄 Issues 182 🔄 Pull requests 8 🏠 Actions 📁 Projects 0 📖 Wiki 🛡 Security 📊 Insights

Be notified of new releases  
Create your free GitHub account today to subscribe to this repository for new releases and build software alongside 40 million developers.  
Sign up Dismiss

Releases Tags

Pre-release

📁 win-3.2.100  
🔗 def0757  
Compare

### 3.2.100

👤 enricogior released this on 1 Jul 2016 · 1210 commits to 3.0 since this release

This is the first release of Redis on Windows 3.2.

This release is based on `antirez/redis/3.2.1` plus some Windows specific fixes. It has passed all the standard tests but it hasn't been tested in a production environment.

Therefore, **before considering using this release in production, make sure to test it thoroughly in your own test environment.**

See the [release notes](#) for details.

Assets 4

📦 <a href="#">Redis-x64-3.2.100.msi</a>	5.8 MB
📦 <a href="#">Redis-x64-3.2.100.zip</a>	4.98 MB
📄 <a href="#">Source code (zip)</a>	
📄 <a href="#">Source code (tar.gz)</a>	



### 3. 将下载好的文件解压，将文件夹重新命名为 Redis，此文件只需解压无需安装

电脑 > 本地磁盘 (F:) > soft > worksoft

名称	修改日期	类型	大小
Android Studio	2019/12/10 16:07	文件夹	
curl	2019/12/10 18:54	文件夹	
cwRsync	2020/2/2 11:19	文件夹	
DeskPins-chs 1.30	2019/4/11 17:13	文件夹	
Dict	2020/1/14 15:50	文件夹	
Docker Toolbox	2020/2/8 14:48	文件夹	
etc	2019/12/10 18:54	文件夹	
Microsoft VS Code	2020/2/14 16:17	文件夹	
mysql-8.0.18-winx64	2020/2/24 13:09	文件夹	
Navicat Premium 12	2020/1/11 10:05	文件夹	
prince	2019/12/23 12:44	文件夹	
Redis	2020/2/24 16:34	文件夹	
tmp	2020/2/2 10:41	文件夹	
txkt	2020/1/4 9:47	文件夹	
var	2019/12/10 18:48	文件夹	
vnc	2019/12/26 13:58	文件夹	
DeskPins.zip	2020/2/17 11:56	好压 ZIP 压缩文件	74 KB
my.ini	2020/2/21 10:02	配置设置	1 KB
mysql-8.0.18-winx64.zip	2020/2/11 10:11	好压 ZIP 压缩文件	278,804 KB
Redis-x64-3.2.100.zip	2020/2/24 15:43	好压 ZIP 压缩文件	5,102 KB



## 4. 打开一个 cmd 窗口使用 cd 命令切换目录到 F:\soft\worksoft\Redis

电脑 > 本地磁盘 (F:) > soft > worksoft

名称	修改日期	类型	大小
Android Studio	2019/12/10 16:07	文件夹	
curl	2019/12/10 18:54	文件夹	
cwRsync	2020/2/2 11:19	文件夹	
DeskPins-chs 1.30	2019/4/11 17:13	文件夹	
Dict	2020/1/14 15:50	文件夹	
Docker Toolbox	2020/2/8 14:48	文件夹	
etc	2019/12/10 18:54	文件夹	
Microsoft VS Code	2020/2/14 16:17	文件夹	
mysql-8.0.18-winx64	2020/2/24 13:09	文件夹	
Navicat Premium 12	2020/1/11 10:05	文件夹	
prince	2019/12/23 12:44	文件夹	
Redis	2020/2/24 16:34	文件夹	
tmp	2020/2/2 10:41	文件夹	
txkt	2020/1/4 9:47	文件夹	
var	2019/12/10 18:48	文件夹	
vnc	2019/12/26 13:58	文件夹	
DeskPins.zip	2020/2/17 11:56	好压 ZIP 压缩文件	74 KB
my.ini	2020/2/21 10:02	配置设置	1 KB
mysql-8.0.18-winx64.zip	2020/2/11 10:11	好压 ZIP 压缩文件	278,804 KB
Redis-x64-3.2.100.zip	2020/2/24 15:43	好压 ZIP 压缩文件	5,102 KB



## 5. 运行 redis-server.exe redis.windows.conf

```
F:\soft\worksoft\Redis>redis-server.exe redis.windows.conf

Redis 3.2.100 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 7760

http://redis.io

[7760] 24 Feb 16:47:51.051 # Server started, Redis version 3.2.100
[7760] 24 Feb 16:47:51.051 * The server is now ready to accept connections on port 6379
```



## 6. 运行 redis-server.exe redis.windows.conf ，报错

管理员: 命令提示符

```
Microsoft Windows [版本 10.0.18362.657]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>f:

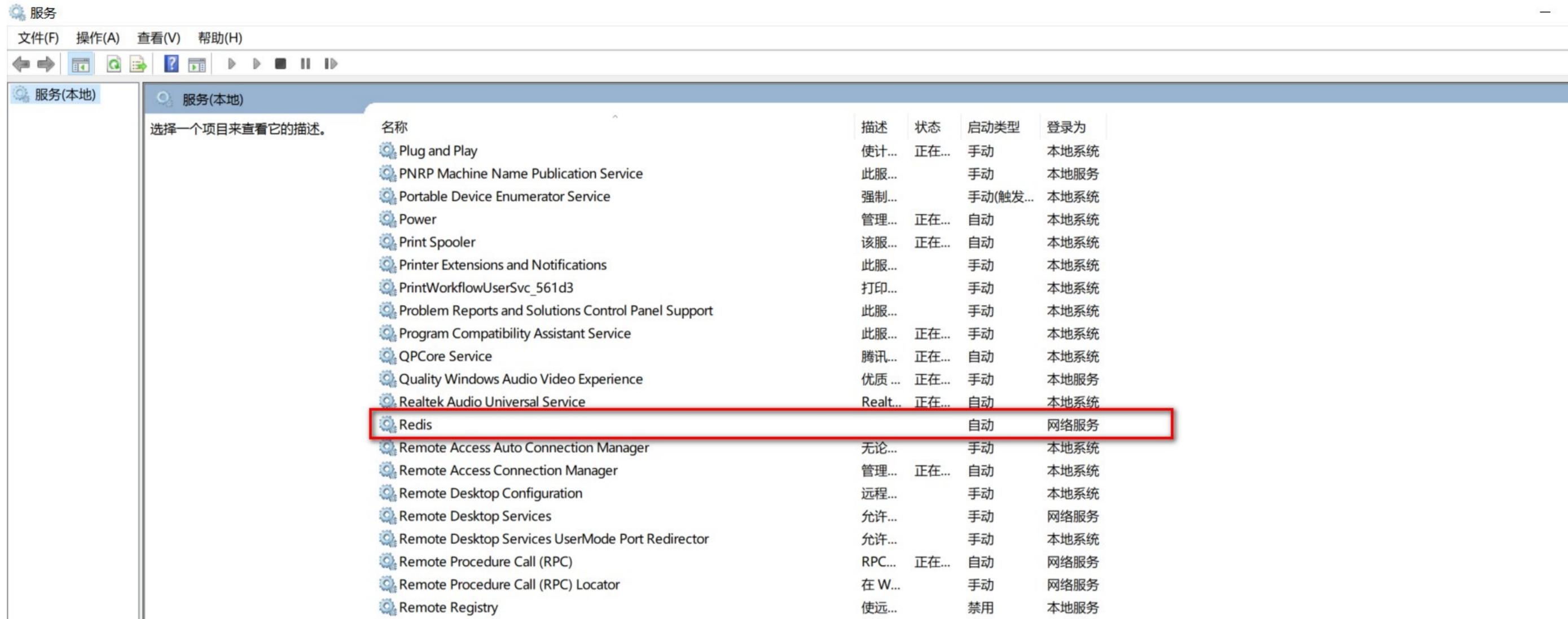
F:\>cd soft\worksoft\Redis

F:\soft\worksoft\Redis>redis-server.exe redis.windows.conf
[11400] 24 Feb 16:43:11.092 # Creating Server TCP listening socket 127.0.0.1:6379: bind: No error

F:\soft\worksoft\Redis>_
```



## 7. 解决方案：win+R 然后输入 services.msc ，调出 Windows 服务，然后找到 Redis ，右键手动停止就好





## 8. 基本数据结构

- String: 是二进制安全的，意思是 redis 的 string 可以包含任何数据。比如 jpg 图片或者序列化的对象；最大能存储 512MB。
- Hash: 一个 string 类型的 key 和 value 的映射表，hash 特别适合用于存储对象；存储  $2^{32} - 1$  键值对（40 多亿）。
- List: 按照插入顺序排序，你可以添加一个元素到列表的头部（左边）或者尾部（右边）。可存储  $2^{32} - 1$  元素（每个列表可存储 40 多亿）
- Set: 无序集合，通过哈希表实现的，所以添加，删除，查找的复杂度都是  $O(1)$ 。
- Sorted Set: 有序集合每个元素都会关联一个 double 类型的分数，redis 正是通过分数来为集合中的成员进行从小到大的排序；zset 的成员是唯一的，但分数 (score) 却可以重复。



## 9. 各个数据类型应用场景

类型	简介	特性	场景
String(字符串)	二进制安全	可以包含任何数据,比如jpg图片或者序列化的对象,一个键最大能存储512M	---
Hash(字典)	键值对集合,即编程语言中的Map类型	适合存储对象,并且可以像数据库中update一个属性一样只修改某一项属性值(Memcached中需要取出整个字符串反序列化成对象修改完再序列化存回去)	存储、读取、修改用户属性
List(列表)	链表(双向链表)	增删快,提供了操作某一段元素的API	1,最新消息排行等功能(比如朋友圈的时间线) 2,消息队列
Set(集合)	哈希表实现,元素不重复	1、添加、删除,查找的复杂度都是O(1) 2、为集合提供了求交集、并集、差集等操作	1、共同好友 2、利用唯一性,统计访问网站的所有独立ip 3、好友推荐时,根据tag求交集,大于某个阈值就可以推荐
Sorted Set(有序集合)	将Set中的元素增加一个权重参数score,元素按score有序排列	数据插入集合时,已经进行天然排序	1、排行榜 2、带权重的消息队列



## 10. 基本使用

- String: SET key member/GET key
- Hash: HMSET key field1 "Hello" field2 "World"/HGET/HGETALL key field1
- List: lpush key member/lrange key 0 10
- Set: sadd key member/smembers key
- Sorted Set: zadd key score member/ZRANGEBYSCORE key 0 1000



10. 另启一个 cmd 窗口，原来的不要关闭，不然就无法访问服务端，切换到 redis 目录下运行：`redis-cli.exe -h 127.0.0.1 -p 6379`

```
C:\WINDOWS\system32>f:  
  
F:\>cd F:\soft\worksoft\Redis  
  
F:\soft\worksoft\Redis>redis-cli.exe -h 127.0.0.1 -p 6379  
127.0.0.1:6379>
```



## 11. Java 使用 Redis- 引入依赖

```
<!-- https://mvnrepository.com/artifact/redis.clients/jedis -->
```

```
<dependency>
```

```
  <groupId>redis.clients</groupId>
```

```
  <artifactId>jedis</artifactId>
```

```
  <version>3.1.0</version>
```

```
</dependency>
```



## 12. Java 使用 Redis- 创建 Jedis 实例

```
// 连接本地的 Redis 服务
```

```
Jedis jedis = new Jedis("localhost");
```

```
jedis.auth("hogwarts");// 此处为你设置的密码
```

```
System.out.println(" 连接成功 ");
```

```
// 查看服务是否运行
```

```
System.out.println(" 服务正在运行 : "+jedis.ping());
```



## 13.Java 使用 Redis- 常用方法

- `set(key, value)` : 给数据库中名称为 `key` 的 `string` 赋予值 `value`
- `get(key)` : 返回数据库中名称为 `key` 的 `string` 的 `value`
- `lpush(key, value)` : 在名称为 `key` 的 `list` 头添加一个值为 `value` 的元素
- `lrange(key, start, end)` : 返回名称为 `key` 的 `list` 中 `start` 至 `end` 之间的元素 ( 下标从 0 开始, 下同 )
- `keys(pattern)` : 返回满足给定 `pattern` 的所有 `key`



扫码添加小助手，回复『公开课』即可进群  
课程视频领取请联系课程顾问