# 安装

官网地址：<https://redis.io/>

下载地址：<https://redis.io/download>

## Linux安装Redis服务端

## Windows安装Redis服务端

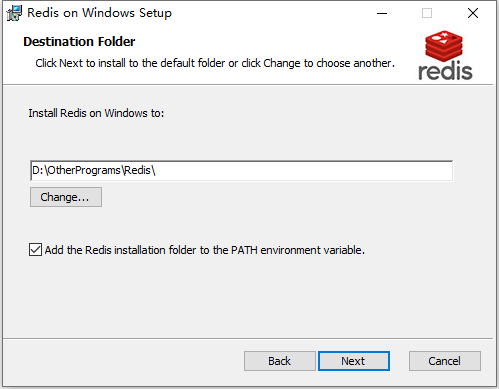
Windows版本的Redis的下载地址为：

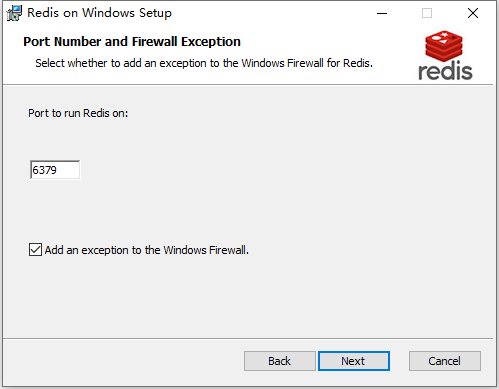
<https://github.com/tporadowski/redis>

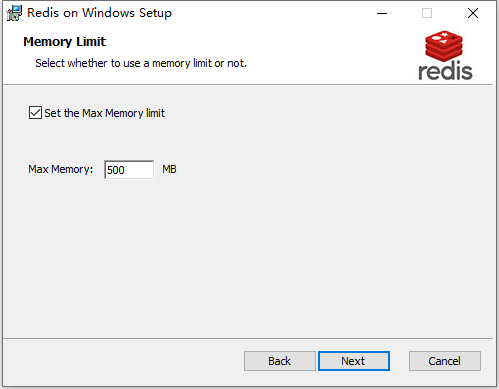
<https://github.com/tporadowski/redis/releases>

这里，我们下载最新版本：**Redis-x64-5.0.9.msi**

安装过程如下：







安装完成之后，自动加入了Path变量并注册为系统服务，如下图所示：



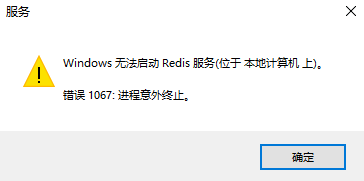


查看Redis服务的属性，可以看到可执行文件路径格式为：

|  |
| --- |
| …\redis-server.exe --service-run "…\redis.windows-service.conf" |

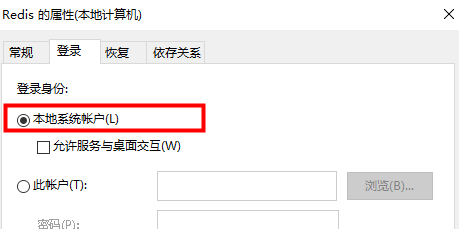
### 启动错误问题

我们启动服务，提示如下错误信息：



**第一步：把Redis网络服务修改为本地服务**

双击Redis服务，把网络服务修改为本地服务：



重新启动发现还是启动不了，查看Redis安装根目录下的server\_log.txt日志文件，显示如下错误信息：

|  |
| --- |
| oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo  Redis version=5.0.9, bits=64, commit=9414ab9b, modified=0, pid=13000, just started  Configuration loaded  Could not create server TCP listening socket 127.0.0.1:6379: bind: 存储控制块地址无效。 |

这个问题应该是我们安装之后，已经自动启动了，只是Redis服务没有显示已经启动，所以我们在Redis点击启动时，一直都是显示“存储控制块地址无效”的错误信息。

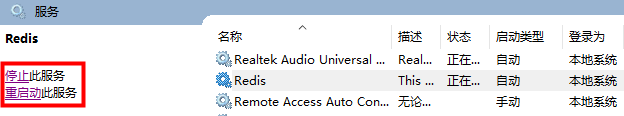
**第二步：尝试关闭Redis服务再重启**

我们用管理员身份打开命令行，输入如下信息：

|  |
| --- |
| C:\Users\Administrator\Desktop  λ redis-cli.exe  127.0.0.1:6379> shutdown  not connected> |

也就是说我们先登录客户端，然后关闭Redis服务。

现在我们打开Redis系统服务，进行启动，发现启动成功，如下图所示：



这样就可以了。

回顾一下之前的“存储控制块地址无效”的错误信息，很明显应该是已经启动了占用了配置的存储块地址了，我们在Redis系统服务点击启动，自然无法在使用相同的存储块了，所以才显示“存储块地址无效”的错误。

### 命令行启动和关闭服务

|  |  |
| --- | --- |
| 启动服务 | redis-server --service-start |
| 停止服务 | redis-server --service-stop |

### 配置文件地址

Redis的配置文件为安装根目录下的**redis.windows-service.conf**文件。

由于Redis启动时使用的是该配置文件，所以我们可以在这里修改我们的Redis启动时的IP、端口等配置信息。

### Redis配置项

|  |  |
| --- | --- |
| requirepass | 配置认证密码。 |
| database | 修改默认数据库的数量，必须 ＞10 个以上。 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 数据类型

## 字符串类型

## 散列类型

## 列表类型

## 集合类型

## 有序集合类型

# 缓存

## 文件缓存

假如我们有几千万个tag列表分页需要缓存，怎么办？

我们可以给每个tag列表内容生成一个缓存文件，然后把该缓存文件的路径和键保存到Redis缓存中。

或者是Redis的tag键来对应几千万个tag列表页的数组，在数组中的每个元素也是一个键值对，而这个键值对才是各个tag标签对应的分页数据。只是这样处理的话，几千万个tag如果同时过期，那查询的时候估计数据库会崩掉。

php读写txt文件与读写数据库哪个更快？解决办法：

<https://blog.csdn.net/forest_fire/article/details/50944166>

另外一种更好的做法是：

把缓存索引写在硬盘上，当我们需要读取某个tag时，我们可以先把缓存索引文件读取出来，然后再从缓存索引文件中找到该tag信息的缓存过期时间以及对应的缓存文件，判断是否缓存过期再读取数据库重新生成内容。另外，在tag数量特别大的时候，我们可以根据tag的ID来进行求余运算，把tag分配到不同的缓存索引文件中去。但是仍然是有个问题，就是缓存文件同时过期的问题。

**最保险和安全的做法是**：

一个tag对应一个 redis的key，并设置不同的过期时间，value则是对应的文件路径。

这样的话Redis需要几千万个key来保存对应的tag列表数据。

就算Redis宕机了，也不影响读取tag缓存数据，因为缓存的目的是用来更新数据而已。

另外一个要考虑用户和爬虫，同时访问tag的频率的高低问题。

# 分布式锁

# 消息队列

# 过滤器

10分钟彻底理解Redis的持久化机制：RDB和AOF

<https://juejin.im/post/5d09a9ff51882577eb133aa9>

看完这篇Redis缓存三大问题，保你面试能造火箭，工作能拧螺丝。

<https://juejin.im/post/5edceb206fb9a047a644684f>

redis 实现搜索热词统计

<https://juejin.im/post/5ed736dce51d45784f800dda>

【Redis面试题】如何使用Redis实现微信步数排行榜？

<https://juejin.im/post/5ed9af276fb9a047ec3f9bd3>

基于Redis的分布式锁实现

<https://juejin.im/post/5cc165816fb9a03202221dd5>

如何用Redis实现分布式锁？

<https://juejin.im/post/5e9473f5e51d454702460323>

Redis系列(一)：Redis简介及环境安装

<https://juejin.im/post/5e1bc399f265da3e505f0ec4>

Redis系列(二)：Redis的5种数据结构及其常用命令

<https://juejin.im/post/5e24f57de51d4531037176db>

Redis系列(三)：Redis的持久化机制(RDB、AOF)

<https://juejin.im/post/5e426906f265da571c4c8a49>

Redis系列(四)：Redis的复制机制(主从复制)

<https://juejin.im/post/5e8bd91df265da47fa28b8f4>