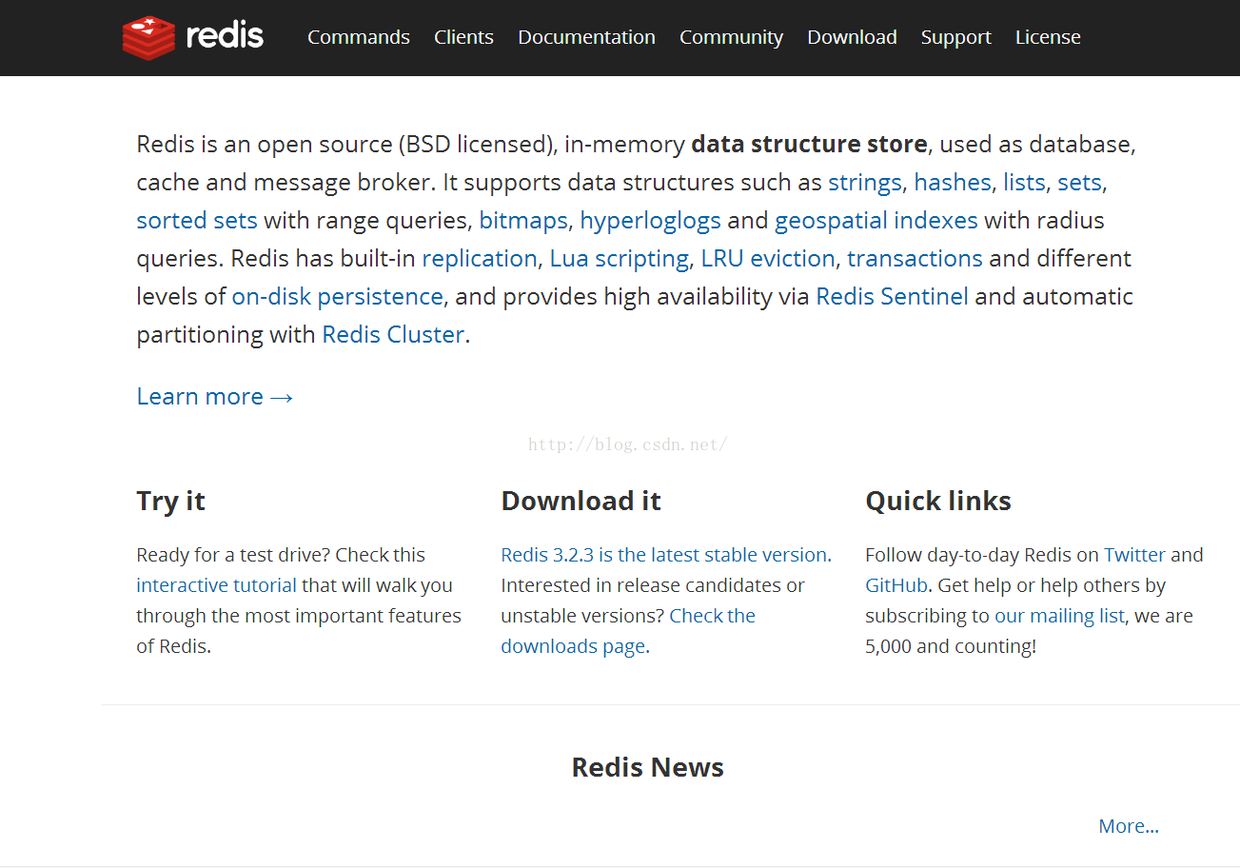
Windows 64位下安装Redis超详细教程

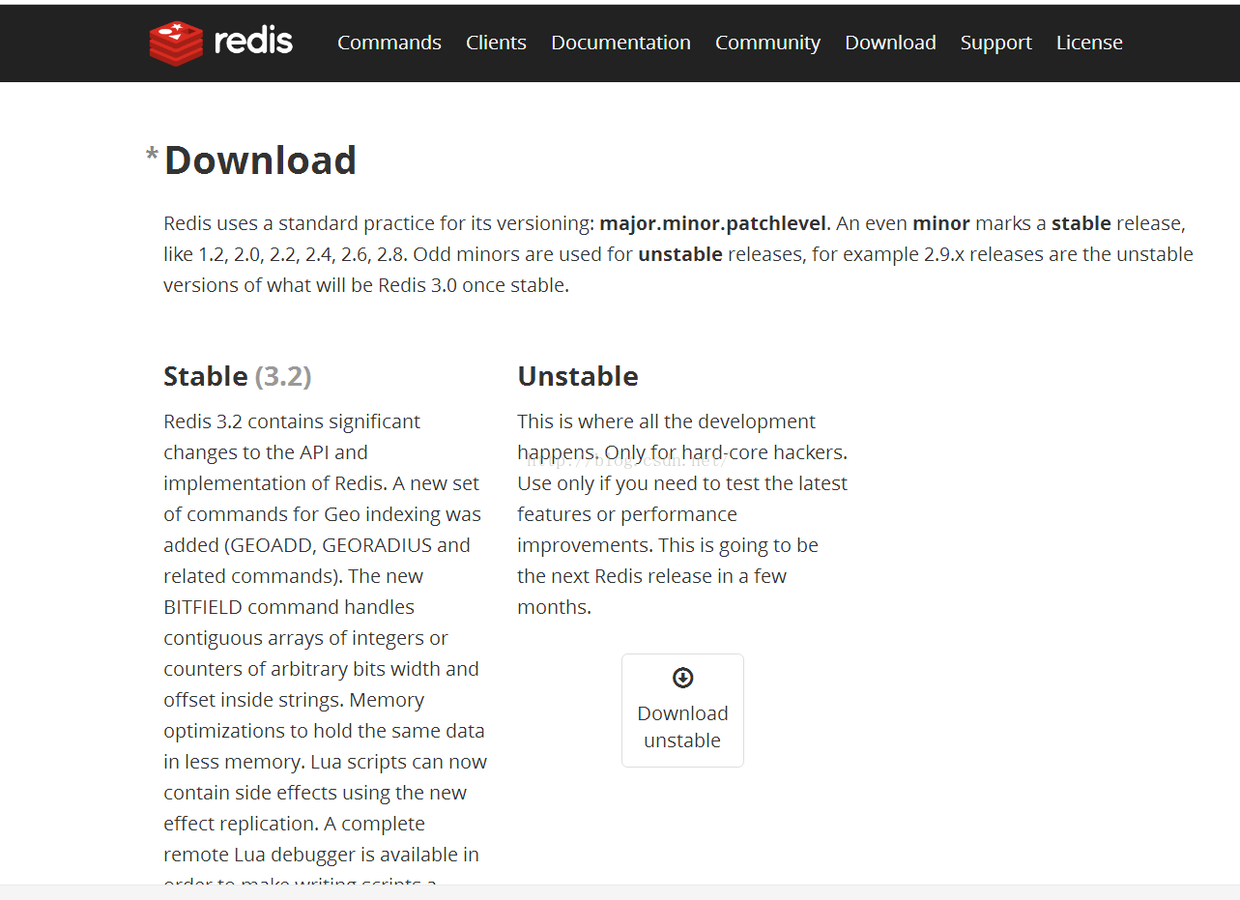
　　Redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash(哈希类型)。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave。

　　1、安装Redis

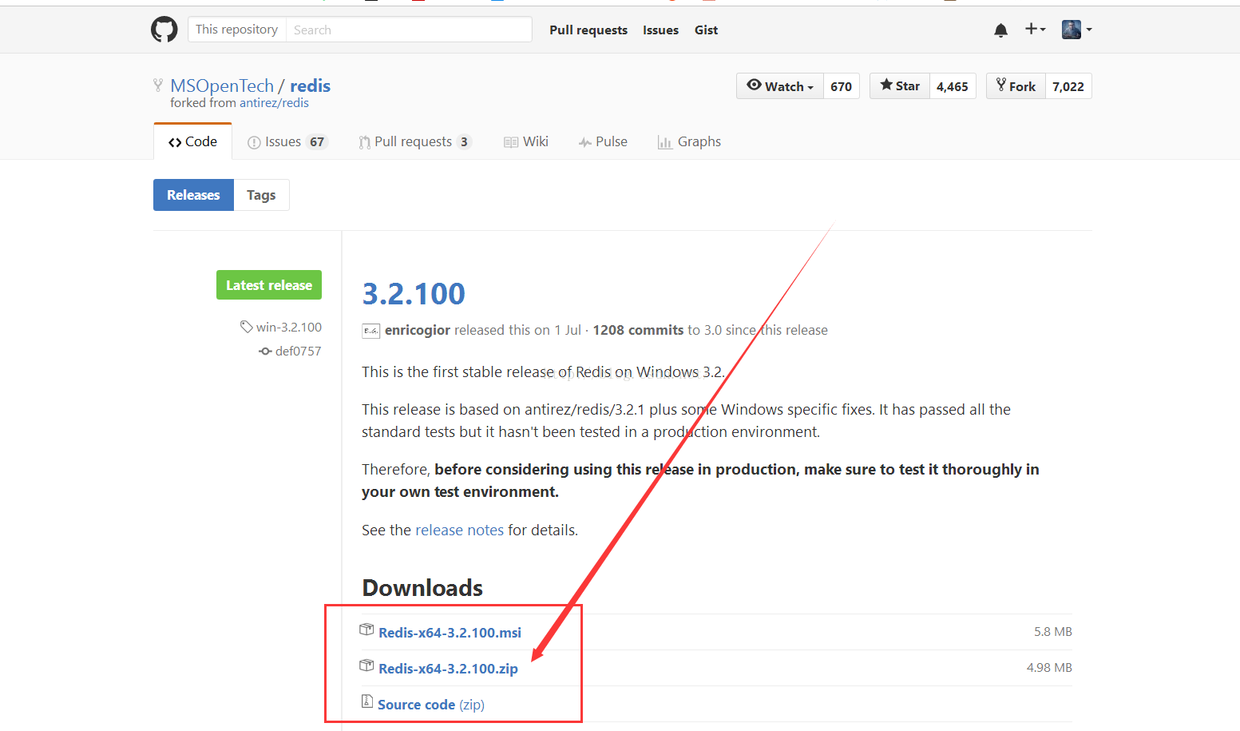
　　官方网站：http://redis.io/



　　官方下载：http://redis.io/download 可以根据需要下载不同版本

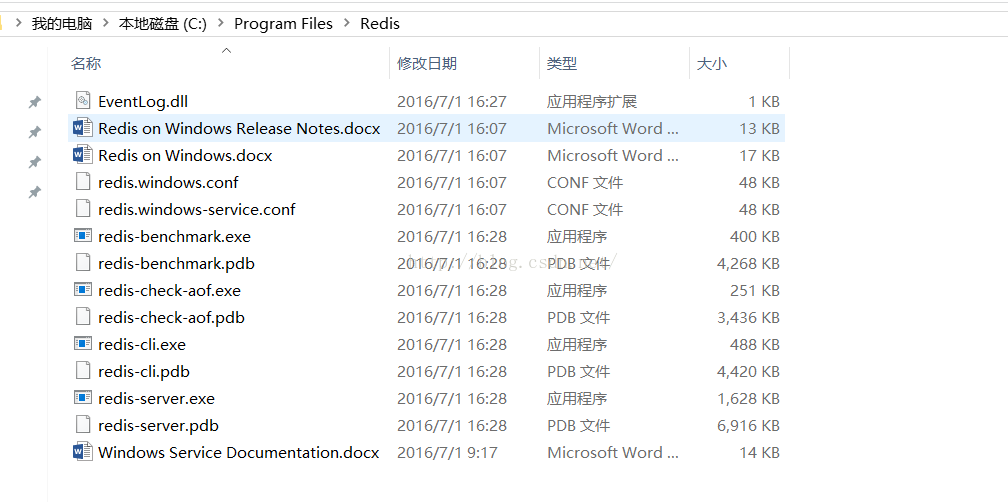


　　windows版：https://github.com/MSOpenTech/redis/releases



　　我选择的是第三种，在github上下载的zip压缩文件。

　　下载完成后，在C盘【Program Files】目录下新建文件夹【Redis】，右键解压Redis ZIP包，把所有文件解压到redis文件夹中。



　　文件介绍：

　　redis-benchmark.exe #基准测试

　　redis-check-aof.exe # aof

　　redischeck-dump.exe # dump

　　redis-cli.exe # 客户端

　　redis-server.exe # 服务器

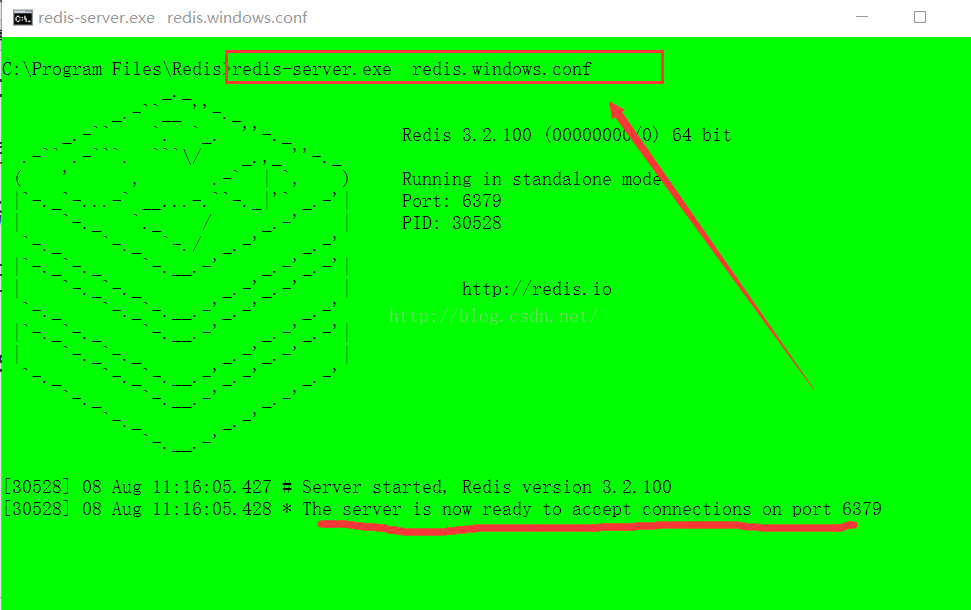
　　redis.windows.conf # 配置文件

　　2、启动Redis

　　windows 运行(快捷键：按住shift + 鼠标右键)，选择在此处打开命令窗口，进入cmd操作系统窗口。

　　使用命令【redis-server.exe redis.windows.conf】，启动redis 服务。

　　服务启动成功状态



　　不要问我为啥颜色是绿色的，自己去设置，个人感觉护眼(心理作用，哈哈 ^\_^)。

　　启动redis服务的doc窗口，不用关闭，因为服务需要一直执行，关闭服务，直接关闭窗口就行。

　　3、测试使用

　　新打开一个doc窗口，用自带的客户端工具进行测试 命令【redis-\*\*cli.exe\*\*】,详细操作如下。。



　　4.Java开发包Jedis

　　Jedis ：http://www.oschina.net/p/jedis (Redis的官方首选Java开发包)

<!--Redis -->

<dependency>

<groupId>redis.clients</groupId>

<artifactId>jedis</artifactId>

<version>2.0.0</version>

<type>jar</type>

<scope>compile</scope>

</dependency>

　　测试例子原帖：http://flychao88.iteye.com/blog/1527163

package com.lujianing.utils;

import org.junit.Before;

import org.junit.Test;

import redis.clients.jedis.Jedis;

import redis.clients.jedis.JedisPool;

import redis.clients.jedis.JedisPoolConfig;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

/\*\*

\* Created by lujianing on 14-2-28.

\*/

public class JedisUtilTest {

JedisPool pool;

Jedis jedis;

@Before

public void setUp() {

pool = new JedisPool(new JedisPoolConfig(), "192.168.10.61");

jedis = pool.getResource();

// jedis.auth("password");

}

@Test

public void testGet(){

System.out.println(jedis.get("lu"));

}

/\*\*

\* Redis存储初级的字符串

\* CRUD

\*/

@Test

public void testBasicString(){

//-----添加数据----------

jedis.set("name","minxr");//向key-->name中放入了value-->minxr

System.out.println(jedis.get("name"));//执行结果：minxr

//-----修改数据-----------

//1、在原来基础上修改

jedis.append("name","jarorwar"); //很直观，类似map 将jarorwar append到已经有的value之后

System.out.println(jedis.get("name"));//执行结果:minxrjarorwar

//2、直接覆盖原来的数据

jedis.set("name","闵晓荣");

System.out.println(jedis.get("name"));//执行结果：闵晓荣

//删除key对应的记录

jedis.del("name");

System.out.println(jedis.get("name"));//执行结果：null

/\*\*

\* mset相当于

\* jedis.set("name","minxr");

\* jedis.set("jarorwar","闵晓荣");

\*/

jedis.mset("name","minxr","jarorwar","闵晓荣");

System.out.println(jedis.mget("name","jarorwar"));

}

/\*\*

\* jedis操作Map

\*/

@Test

public void testMap(){

Map<String,String> user=new HashMap<String,String>();

user.put("name","minxr");

user.put("pwd","password");

jedis.hmset("user",user);

//取出user中的name，执行结果:[minxr]-->注意结果是一个泛型的List

//第一个参数是存入redis中map对象的key，后面跟的是放入map中的对象的key，后面的key可以跟多个，是可变参数

List<String> rsmap = jedis.hmget("user", "name");

System.out.println(rsmap);

//删除map中的某个键值

// jedis.hdel("user","pwd");

System.out.println(jedis.hmget("user", "pwd")); //因为删除了，所以返回的是null

System.out.println(jedis.hlen("user")); //返回key为user的键中存放的值的个数1

System.out.println(jedis.exists("user"));//是否存在key为user的记录 返回true

System.out.println(jedis.hkeys("user"));//返回map对象中的所有key [pwd, name]

System.out.println(jedis.hvals("user"));//返回map对象中的所有value [minxr, password]

Iterator<String> iter=jedis.hkeys("user").iterator();

while (iter.hasNext()){

String key = iter.next(); System.out.println(key+":"+jedis.hmget("user",key));

}

}

/\*\*

\* jedis操作List

\*/

@Test

public void testList(){

//开始前，先移除所有的内容

jedis.del("java framework");

System.out.println(jedis.lrange("java framework",0,-1));

//先向key java framework中存放三条数据

jedis.lpush("java framework","spring");

jedis.lpush("java framework","struts");

jedis.lpush("java framework","hibernate");

//再取出所有数据jedis.lrange是按范围取出，

// 第一个是key，第二个是起始位置，第三个是结束位置，jedis.llen获取长度 -1表示取得所有

System.out.println(jedis.lrange("java framework",0,-1));

}

/\*\*

\* jedis操作Set

\*/

@Test

public void testSet(){

//添加

jedis.sadd("sname","minxr");

jedis.sadd("sname","jarorwar");

jedis.sadd("sname","闵晓荣");

jedis.sadd("sanme","noname");

//移除noname

jedis.srem("sname","noname");

System.out.println(jedis.smembers("sname"));//获取所有加入的value

System.out.println(jedis.sismember("sname", "minxr"));//判断 minxr 是否是sname集合的元素

System.out.println(jedis.srandmember("sname"));

System.out.println(jedis.scard("sname"));//返回集合的元素个数

}

@Test

public void test() throws InterruptedException {

//keys中传入的可以用通配符

System.out.println(jedis.keys("\*")); //返回当前库中所有的key [sose, sanme, name, jarorwar, foo, sname, java framework, user, braand]

System.out.println(jedis.keys("\*name"));//返回的sname [sname, name]

System.out.println(jedis.del("sanmdde"));//删除key为sanmdde的对象 删除成功返回1 删除失败（或者不存在）返回 0

System.out.println(jedis.ttl("sname"));//返回给定key的有效时间，如果是-1则表示永远有效

jedis.setex("timekey", 10, "min");//通过此方法，可以指定key的存活（有效时间） 时间为秒

Thread.sleep(5000);//睡眠5秒后，剩余时间将为<=5

System.out.println(jedis.ttl("timekey")); //输出结果为5

jedis.setex("timekey", 1, "min"); //设为1后，下面再看剩余时间就是1了

System.out.println(jedis.ttl("timekey")); //输出结果为1

System.out.println(jedis.exists("key"));//检查key是否存在 System.out.println(jedis.rename("timekey","time"));

System.out.println(jedis.get("timekey"));//因为移除，返回为null

System.out.println(jedis.get("time")); //因为将timekey 重命名为time 所以可以取得值 min

//jedis 排序

//注意，此处的rpush和lpush是List的操作。是一个双向链表（但从表现来看的）

jedis.del("a");//先清除数据，再加入数据进行测试

jedis.rpush("a", "1");

jedis.lpush("a","6");

jedis.lpush("a","3");

jedis.lpush("a","9");

System.out.println(jedis.lrange("a",0,-1));// [9, 3, 6, 1]

System.out.println(jedis.sort("a")); //[1, 3, 6, 9] //输入排序后结果

System.out.println(jedis.lrange("a",0,-1));

}

}

　　Redis会定时 保存数据到硬盘上。