更多信息参考：<https://github.com/memcached/memcached/wiki>

memcache的客户端操作一般都只提供了get和set等简单的操作，这些操作都是非常高效的

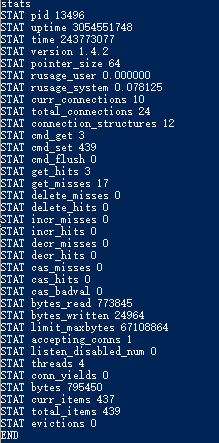
memcache是个key-value存储系统，但是在某些时候我们需要遍历memcache的所有key；

遍历主要使用stats相关命令：

客户端命令简介：

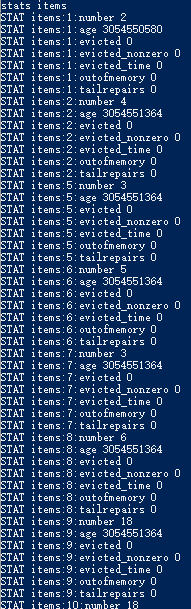
stats命令：

列出当前memcache的状态



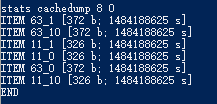
stats items 命令：列出

列出不同的items存储的状态（memcache会有优化机制，因此他会将不同size的数据存储在不同的区块中，因此每个items后面的num可能不同），后面的num代表该item存储的key的数量。



stats cachedump 3 0命令：

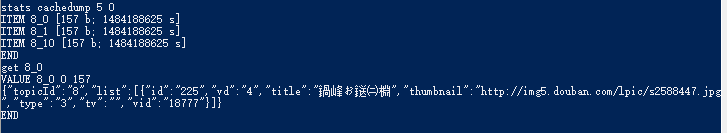
3指的是上面stats iems结果中items后面的数字，0表示列出所有的key，如果是1则列出第一个key（该值应该在0与该item的number之间）



这里执行stats cachedump 8 0列出8这个item下的所有的key

ITEM后面的字符串时存储的key，之后的中括号中的第一个数字代表value的长度（单位b代表字节），再后面的参数代表此key的创建时间（以1970年为开始）

知道了key，然后就可以使用get命令获得value的结果了。



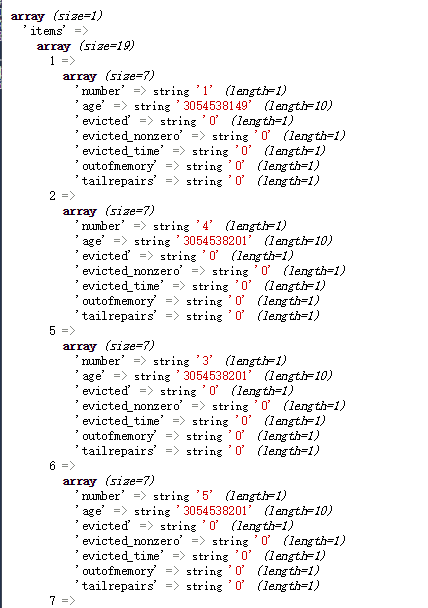
后面的VALUE，之后的数字0不知什么意思，157代表此值的长度。

写程序遍历memcache中的key：

程序：$items=$mem->getExtendedStats();对应上面的stats命令，返回的结果数组与命令行的一致。

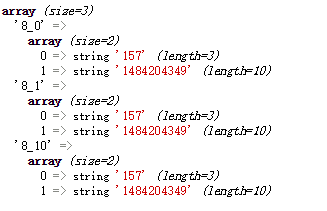


程序： $items=$mem->getExtendedStats("items")['localhost:11211'];这句对应上面的stats items命令，结果安全一样



数组里的key：1、2、3等数字代表的就是stats items 里面的标号（可能是memcache的存储分区标记），根据这些标号执行cachedump命令，可以打印出所有的key，然后get即可得到左右的值

程序：$item=$mem->getExtendedStats('cachedump',5,1)['localhost:11211'];这句的效果如同命令stats cachedump 5 0；得到的结果数组如下：



与stats cachedump 5 0 结果一致，0代表得到的此存储区域的所有key。

结果数组的元素中：0索引值代表的是此key的长度，1索引值代表的是此key的创建时间？（有疑问）。

完整程序：

$mem=new memcache();

$conn=@$mem->connect('localhost',11211); //不加@抑制符会提示

if($conn){

$items=$mem->getExtendedStats("items")['localhost:11211']['items'];

foreach ($items as $key => $value) {

$number=$key;

$keys=$mem->getExtendedStats('cachedump',$number,0)['localhost:11211'];

foreach ($keys as $k => $v) {

var\_dump($mem->get($k));

}

}

}else{

exit("連接memcache失敗");

}