

hash

1, hset

描述: 将哈希表key中的域field的值设为value。如果key不存在, 一个新的哈希表被创建并进行HSET操作。如果域field已经存在于哈希表中, 旧值将被覆盖。

参数: key field value

返回值: 如果field是哈希表中的一个新建域, 并且值设置成功, 返回1。如果哈希表中域field已经存在且旧值已被新值覆盖, 返回0。

2, hsetnx

描述: 将哈希表key中的域field的值设置为value, 当且仅当域field不存在。若域field已经存在, 该操作无效。如果key不存在, 一个新哈希表被创建并执行HSETNX命令。

参数: key field value

返回值: 设置成功, 返回1。如果给定域已经存在且没有操作被执行, 返回0。

3, hget

描述: 返回哈希表key中给定域field的值。

参数: key field

返回值: 给定域的值。当给定域不存在或是给定key不存在时, 返回nil。

4, hmset

描述: 同时将多个field - value(域-值)对设置到哈希表key中。此命令会覆盖哈希表中已存在的域。如果key不存在, 一个空哈希表被创建并执行HMSET操作。

参数: key field value [field value ...]

返回值: 如果命令执行成功, 返回OK。当key不是哈希表(hash)类型时, 返回一个错误。

5, hmget

描述: 返回哈希表key中, 一个或多个给定域的值。如果给定的域不存在于哈希表, 那么返回一个nil值。因为不存在的key被当作一个空哈希表来处理, 所以对一个不存在的key进行HMGET操作将返回一个只带有nil值的表。

参数: key field [field ...]

返回值: 一个包含多个给定域的关联值的表, 表值的排列顺序和给定域参数的请求顺序一样。

6, hgetall

描述: 返回哈希表key中, 所有的域和值。在返回值里, 紧跟每个域名(field name)之后是域的值(value), 所以返回值的长度是哈希表大小的两倍。

参数: key

返回值: 以列表形式返回哈希表的域和域的值。 若key不存在, 返回空列表。

7, hdel

描述: 删除哈希表key中的一个或多个指定域, 不存在的域将被忽略。

参数: key field [field ...]

返回值: 被成功移除的域的数量, 不包括被忽略的域。

8, hlen

描述: 返回哈希表key中域的数量。

参数: key

返回值: 哈希表中域的数量。当key不存在时, 返回0。

9, hexists

描述: 查看哈希表key中, 给定域field是否存在。

参数: key field

返回值: 如果哈希表含有给定域, 返回1。如果哈希表不含有给定域, 或key不存在, 返回0。

10, hincrby

描述: 为哈希表key中的域field的值加上增量increment。增量也可以为负数, 相当于对给定域进行减法操作。

参数: key field increment

返回值: 执行HINCRBY命令之后, 哈希表key中域field的值。

11, hkeys

描述: 返回哈希表key中的所有域。

参数: key

返回值: 一个包含哈希表中所有域的表。当key不存在时, 返回一个空表。

12, hvals

描述: 返回哈希表key中的所有值。

参数: key

返回值: 一个包含哈希表中所有值的表。当key不存在时, 返回一个空表。

以上12个方法的代码例子:

[复制代码](#) 代码如下:

```

<?php
$redis = new redis();
$redis->connect('192.168.1.108', 6379);
$redis->delete('test');
$redis->hset('test', 'key1', 'hello');
echo $redis->hget('test', 'key1');    //结果: hello

echo "<br>";
$redis->hSetNx('test', 'key1', 'world');
echo $redis->hget('test', 'key1');    //结果: hello

$redis->delete('test');
$redis->hSetNx('test', 'key1', 'world');
echo "<br>";
echo $redis->hget('test', 'key1');    //结果: world

echo $redis->hlen('test');    //结果: 1
var_dump($redis->hdel('test', 'key1'));    //结果: bool(true)

$redis->delete('test');
$redis->hSet('test', 'a', 'x');
$redis->hSet('test', 'b', 'y');
$redis->hSet('test', 'c', 'z');
print_r($redis->hkeys('test'));    //结果: Array ( [0] => a [1] => b [2] => c )

print_r($redis->hvals('test'));    //结果: Array ( [0] => x [1] => y [2] => z )

print_r($redis->hgetall('test'));    //结果: Array ( [a] => x [b] => y [c] => z )

var_dump($redis->hExists('test', 'a'));    //结果: bool(true)

$redis->delete('test');
echo $redis->hIncrBy('test', 'a', 3);    //结果: 3
echo $redis->hIncrBy('test', 'a', 1);    //结果: 4

$redis->delete('test');
var_dump($redis->hmset('test', array('name' => 'tank', 'sex' => "man")));    //结果: bool(true)
print_r($redis->hmget('test', array('name', 'sex')));    //结果: Array ( [name] => tank [sex] => man )
?>

```