

title	date	categories	tags
Redis 数据结构之 hash	Sat Jun 04 2016 21:26:34 GMT+0800 (CST)	Redis	redis

## 前言

**Redis**作为cache服务器，支持多种数据结构，String、List、Hash、Set、Zset。多种数据结构的存在，使得 Redis 适用于多种业务,Redis的适用也越发广泛。又过了一周，今天我们来介绍Redis中的数据结构 Hash 的操作命令。

## 数据结构简介

Hash 数据结构即数据存储为 field 、 value 的格式存储，支持针对指定的field所对应的value操作。

## 命令简介

### HSET 命令

语法：HSET key field value

作用：设置一对 field value

返回：如果field不存在则设置成功返回 1 ,否则更新value则返回 0

```
127.0.0.1:6379> HSET key field value
(integer) 1
```

```
127.0.0.1:6379> HSET key field valuevalue
```

(integer) 0

---

## HSETNX 命令

语法 : HSETNX key field value

作用 : 设置一对 field value,如果field已经存在,则不做任何操作

返回 : 如果field不存在则设置成功返回 1 ,否则不做人和网操作返回 0

```
127.0.0.1:6379> HSETNX key field value
```

```
(integer) 0
```

```
127.0.0.1:6379> HSETNX key field1 value1
```

```
(integer) 1
```

---

## HMSET 命令

语法 : HMSET key f1 v1 [f2 v2 ...]

作用 : 设置多对field、value值 返回 : 返货 OK

```
127.0.0.1:6379> HMSET key f1 v1 f2 v2 f3 v3
```

```
OK
```

---

## HINCRBY 命令

语法 : HINCRBY key field data

作用 : 给指定 field 对应的 value 值加上 data 数值

返回 : 成功返回操作后的 value 值, 失败返回对应的错误

```
127.0.0.1:6379> HINCRBY key f1 100
```

```
(error) ERR hash value is not an integer //f1 对应的值不为整型
```

```
127.0.0.1:6379> HSET key f11 100
```

```
(integer) 1
```

```
127.0.0.1:6379> HINCRBY key f11 100
```

```
(integer) 200
```

---

## HINCRFLOAT 命令

语法 : HINCRBYFLOAT key field data(支持浮点数)

作用 : 给指定 field 对应的 value 值加上 data 数值

返回 : 成功返回操作后的 value 值, 失败返回对应的错误

```
127.0.0.1:6379> HINCRBYFLOAT key f11 100.11
```

```
"300.109999999999999999"
```

```
127.0.0.1:6379> HINCRBYFLOAT key f1 100.11
```

```
(error) ERR hash value is not a valid float
```

---

## HGET 命令

语法 : HGET key field

作用 : 返回指定field对应的value值

返回 : 如果key不存在或field不存在则返回NULL, 否则返回对应的value

```
127.0.0.1:6379> HGET keykey field
```

```
(nil)
```

```
127.0.0.1:6379> HGET key field
```

```
"valuevalue"
```

---

## HMGET 命令

语法 : HMGET key field1 [field2 field3 ...]

**作用**：获取多个指定的field对应的value

**返回**：field存在则返回对应的value，否则返回NULL

```
127.0.0.1:6379> HMGET key field f1 f2 f333 f4
```

```
1) "valuevalue"
```

```
2) "v1"
```

```
3) "v2"
```

```
4) (nil)
```

```
5) (nil)
```

---

## HDEL 命令

**语法**：HDEL key field1 [field2 field3 ...]

**作用**：删除指定的 field

**返回**：返回删除的 field 个数

```
127.0.0.1:6379> HDEL key field f1 f2 f333 f4
```

```
(integer) 3
```

---

## HLEN 命令

**语法**：HLEN key

**作用**：获取指定hash中元素的个数

**返回**：返回元素个数

```
127.0.0.1:6379> HLEN key
```

```
(integer) 3
```

---

## HKEYS 命令

作用：获取指定hash中所有field值    返回：返回所有的field值

1) "field1 "

2) "f3"

3) "f11"

语法 : HVALS key

作用：获取指定hash中所有value值    返回：返回所有的value值

1) "value1"

2) "v3"

3) "300.109999999999999999"

语法：HGETALL key

作用：获取指定hash中所有field、value值    返回：返回所有的field、value值

1) "field1 "

2) "value1"

3) "f3"

4) "v3"

5) "f11"

6) "300.109999999999999999"

# HEXISTS 命令

语法: EXISTS key field

作用：检查指定的field是否存在

返回：fiedl存在返回 1，不存在返回 0

```
127.0.0.1:6379> HEXISTS key f3
```

(integer) 1

```
127.0.0.1:6379> HEXISTS key f333
```

(integer) 0

# HSCAN 命令

语法 : HSCAN key cursor

作用：遍历hash

返回：返回部分Hash节点数据

```
127.0.0.1:6379> HSCAN key 0
```

1) "0"

2) 1) "field1"

2) "value1"

3) "f3"

4) "v3"

5) "f11"

6) "300.109999999999999999"

## 总结

**Hash** 主要用来存储field-value值型的数据，可以很方便的存储需要映射的数据，且支持动态添加数据，很是方便。