

Redis-

Set/Get

```
package main

import (
    "fmt"

    _ "github.com/gogf/gf/contrib/nosql/redis/v2"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var ctx = gctx.New()
    _, err := g.Redis().Set(ctx, "key", "value")
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    value, err := g.Redis().Get(ctx, "key")
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    fmt.Println(value.String())
}
```

value

Content Menu

- [Set/Get](#)
- [SetEx](#)
- [HSet/HGetAll](#)
- [HMSet/HMGet](#)

SetEx

```

package main

import (
    "fmt"
    "time"

    _ "github.com/gogf/gf/contrib/nosql/redis/v2"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var ctx = gctx.New()
    err := g.Redis().SetEX(ctx, "key", "value", 1)
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    value, err := g.Redis().Get(ctx, "key")
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    fmt.Println(value.IsNil())
    fmt.Println(value.String())

    time.Sleep(time.Second)

    value, err = g.Redis().Get(ctx, "key")
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    fmt.Println(value.IsNil())
    fmt.Println(value.Val())
}

```

```

false
value
true
<nil>

```

HSet/HGetAll

```

package main

import (
    "fmt"

    _ "github.com/gogf/gf/contrib/nosql/redis/v2"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var (
        ctx = gctx.New()
        key = "key"
    )
    _, err := g.Redis().HSet(ctx, key, g.Map{
        "id": 1,
        "name": "john",
        "score": 100,
    })
    if err != nil {
        g.Log().Fatal(ctx, err)
    }

    // retrieve hash map
    value, err := g.Redis().HGetAll(ctx, key)
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    fmt.Println(value.Map())

    // scan to struct
    type User struct {
        Id      uint64
        Name    string
        Score   float64
    }
    var user *User
    if err = value.Scan(&user); err != nil {
        g.Log().Fatal(ctx, err)
    }
    g.Dump(user)
}

```

```

map[id:1 name:john score:100]
{
    Id: 1,
    Name: "john",
    Score: 100,
}

```

HMSet/HMGet

```

package main

import (
    _ "github.com/gogf/gf/contrib/nosql/redis/v2"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var (
        ctx = gctx.New()
        key = "key"
    )
    err := g.Redis().HmSet(ctx, key, g.Map{
        "id":    1,
        "name":  "john",
        "score": 100,
    })
    if err != nil {
        g.Log().Fatal(ctx, err)
    }

    // retrieve hash map
    values, err := g.Redis().HmGet(ctx, key, "id", "name")
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    g.Dump(values.Strings())
}

```

```

[
    "1",
    "john",
]

```

As per Redis 4.0.0, HMSET is considered deprecated. Please use HSET in new code.

Redis 4.0.0HMSETHSET