Content Menu

gcache

```
import "github.com/gogf/gf/v2/os/gcache"
```

https://pkg.go.dev/github.com/gogf/gf/v2/os/gcache

1. gcacheNewgcachegcache.Cache
2. gcacheinterface{}stringstring[]byte
3. gcacheinterface{}interface{}gcacheGet*interface{}
4. gcachedurationtime.DurationSetduration = 0duration < 0duration > 0

interface{}interface{}

```
package main

import (
        "fmt"
        "github.com/gogf/gf/v2/os/gcache"
        "github.com/gogf/gf/v2/os/gctx"
)

func main() {
        var (
                ctx           = gctx.New()
                key1  int32   = 1
                key2  float64 = 1
                value         = `value`
        )
        _ = gcache.Set(ctx, key1, value, 0)
        fmt.Println(gcache.MustGet(ctx, key1).Val())
        fmt.Println(gcache.MustGet(ctx, key2).Val())
}
```

```
value
<nil>
```

key1key2key2

interface{}gcache/

interface{}*gvar.Var

```go
package main

import (
	"fmt"
	"github.com/gogf/gf/v2/os/gcache"
	"github.com/gogf/gf/v2/os/gctx"
)

func main() {
	type User struct {
		Id   int
		Name string
		Site string
	}
	var (
		ctx   = gctx.New()
		user  *User
		key   = `UserKey`
		value = &User{
			Id:   1,
			Name: "GoFrame",
			Site: "https://goframe.org",
		}
	)
	err := gcache.Set(ctx, key, value, 0)
	if err != nil {
		panic(err)
	}
	v, err := gcache.Get(ctx, key)
	if err != nil {
		panic(err)
	}
	if err = v.Scan(&user); err != nil {
		panic(err)
	}
	fmt.Printf(`%#v`, user)
}
```

```
&main.User{Id:1, Name:"GoFrame", Site:"https://goframe.org"}
```

- -
- -
- -Redis
- -