

-Map

```
gconv.Mapmapstruct/*struct map[string]interface{} struct/*structstruct c/gconv  
/json Maptagsnil
```



```
struct/*struct c/gconv/json -omitempty - oitemptynil,0,"",[ ]
```

```
func Map(value interface{}, tags ...string) map[string]interface{}  
func MapDeep(value interface{}, tags ...string) map[string]interface{}
```

MapDeepstruct/*struct



map<https://pkg.go.dev/github.com/gogf/gf/v2/util/gconv>

Content Menu

-
-
-
-

```
package main  
  
import (  
    "github.com/gogf/gf/v2/frame/g"  
    "github.com/gogf/gf/v2/util/gconv"  
)  
  
func main() {  
    type User struct {  
        Uid int    `c:"uid"`  
        Name string `c:"name"`  
    }  
    //  
    g.Dump(gconv.Map(User{  
        Uid: 1,  
        Name: "john",  
    }))  
    //  
    g.Dump(gconv.Map(&User{  
        Uid: 1,  
        Name: "john",  
    }))  
  
    // map  
    g.Dump(gconv.Map(map[int]int{  
        100: 10000,  
    }))  
}
```

```
{
  "name": "john",
  "uid": 1,
}

{
  "name": "john",
  "uid": 1,
}

{
  "100": 10000,
}
```

c/gconv/json mapgconv/c/json

```
package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/util/gconv"
)

func main() {
    type User struct {
        Uid      int
        Name     string `c:"- "`
        NickName string `c:"nickname, omitempty"`
        Pass1    string `c:"password1"`
        Pass2    string `c:"password2"`
    }
    user := User{
        Uid:    100,
        Name:   "john",
        Pass1:  "123",
        Pass2:  "456",
    }
    g.Dump(gconv.Map(user))
}
```

```
{
  "Uid": 100,
  "password1": "123",
  "password2": "456",
  "nickname": "",
}
```

structmap

```

package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/util/gconv"
)

func main() {
    type User struct {
        Id    int    `c:"uid"`
        Name string `my-tag:"nick-name" c:"name"`
    }
    user := &User{
        Id:    1,
        Name:  "john",
    }
    g.Dump(gconv.Map(user, "my-tag"))
}

```

```

{
    "nick-name": "john",
    "uid": 1,
}

```

map/struct/*struct/embeddedMapMapDeepmap

```

package main

import (
    "fmt"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/util/gconv"
    "reflect"
)

func main() {
    type Base struct {
        Id    int    `c:"id"`
        Date string `c:"date"`
    }
    type User struct {
        UserBase Base    `c:"base"`
        Passport string `c:"passport"`
        Password string `c:"password"`
        Nickname string `c:"nickname"`
    }
    user := &User{
        UserBase: Base{
            Id:    1,
            Date: "2019-10-01",
        },
        Passport: "john",
        Password: "123456",
        Nickname: "JohnGuo",
    }
    m1 := gconv.Map(user)
    m2 := gconv.MapDeep(user)
    g.Dump(m1, m2)
    fmt.Println(reflect.TypeOf(m1["base"]))
    fmt.Println(reflect.TypeOf(m2["base"]))
}

```

```

{
  "base": {
    Id:    1,
    Date: "2019-10-01",
  },
  "passport": "john",
  "password": "123456",
  "nickname": "JohnGuo",
}
{
  "base": {
    "id":    1,
    "date": "2019-10-01",
  },
  "passport": "john",
  "password": "123456",
  "nickname": "JohnGuo",
}
main.Base
map[string]interface {}

```

