

-Map

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

✔ struct/*struct c/gconv/json -omitempty - oitempty nil,0,"", []

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

MapDeepstruct/*struct

✔ map<https://godoc.org/github.com/gogf/gf/util/gconv>

Content Menu

- [1](#)
- [2](#)
- [3](#)
- [4](#)

1

```
package main

import (
    "github.com/gogf/gf/frame/g"
    "github.com/gogf/gf/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
}
```

2

```

package main

import (
    "github.com/gogf/gf/frame/g"
    "github.com/gogf/gf/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"
}

```

3

```

package main

import (
    "github.com/gogf/gf/frame/g"
    "github.com/gogf/gf/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
}

```

4

map/struct/*struct/MapMapDeep

```

package main

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

func main() {
    type Base struct {
        Id    int    `c:"id"`
        Date string `c:"date"`
    }
    type User struct {
        Base      `c:"base"`
        Passport string `c:"passport"`
        Password  string `c:"password"`
        Nickname  string `c:"nickname"`
    }
    user := new(User)
    user.Id = 1
    user.Date = "2019"
    user.Nickname = "John"
    user.Passport = "johng"
    user.Password = "123456"
    g.Dump(gconv.MapDeep(user))
}

```

```

{
  "base": {
    "date": "2019",
    "id": 1
  },
  "nickname": "John",
  "passport": "johng",
  "password": "123456"
}

```