

-

structGoFrameQueryJSON/XMLstructstruct

RequestParseGet*StructAPI <https://pkg.go.dev/github.com/gogf/gf/v2/net/ghttp#Request>

Content Menu
• • ◦ ◦ • Parse

struct

1. struct()
2. -/_/ struct
- 3.

map	struct	
name	Name	match
Email	Email	match
nickname	NickName	match
NICKNAME	NickName	match
Nick-Name	NickName	match
nick_name	NickName	match
nick name	NickName	match
NickName	Nick_Name	match
Nick-name	Nick_Name	match
nick_name	Nick_Name	match
nick name	Nick_Name	match

 gconvc/gconv/json -[Struct](#)



structtagtag/param/params

```
type User struct{
    Id    int
    Name  string
    Pass1 string `p:"password1"`
    Pass2 string `p:"password2"`
}
```

ppassword1Pass1password2Pass2tag

Parse

Parsestructstructtag



GoFrame v2-

```

package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
)

type RegisterReq struct {
    Name string
    Pass string `p:"password1"`
    Pass2 string `p:"password2"`
}

type RegisterRes struct {
    Code int    `json:"code"`
    Error string `json:"error"`
    Data interface{} `json:"data"`
}

func main() {
    s := g.Server()
    s.BindHandler("/register", func(r *ghttp.Request) {
        var req *RegisterReq
        if err := r.Parse(&req); err != nil {
            r.Response.WriteJsonExit(RegisterRes{
                Code: 1,
                Error: err.Error(),
            })
        }
        // ...
        r.Response.WriteJsonExit(RegisterRes{
            Data: req,
        })
    })
    s.SetPort(8199)
    s.Run()
}

```

RegisterReqRegisterRes

r.Parse(&req)RegisterReqreqnilerrreqnilRegisterResJSONr.Response.WriteJsonExit
RegisterResjsonJSON

DataRegisterReqjsonJSON

curl

```

$ curl "http://127.0.0.1:8199/register?
name=john&password1=123&password2=456"
{"code":0,"error":"","data":{"Name":"john","Pass":"123","Pass2":"456"}}

$ curl -d "name=john&password1=123&password2=456" -X POST "http://127.
0.0.1:8199/register"
{"code":0,"error":"","data":{"Name":"john","Pass":"123","Pass2":"456"}}

```

GETPOST