

Requestvgvalid-

## 1

v

package main

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

type RegisterReq struct {
    Name string `p:"username" v:"required|length:6,30#|:min:max"`
    Pass string `p:"password1" v:"required|length:6,30#|"`
    Pass2 string `p:"password2" v:"required|length:6,30|same:password1#|"`
}

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()
}
```

RegisterReqRegisterResJSONjsonpRegisterReqjson

DataRegisterReqjsonJSON

curl

```
$ curl "http://127.0.0.1:8199/register?
name=john&password1=123456&password2=123456"
{"code":0,"error":"","data":{"Name":"john","Pass":"123456","Pass2":"123456"}}

$ curl "http://127.0.0.1:8199/register?
name=john&password1=123456&password2=12345"
{"code":1,"error":"","data":null}

$ curl "http://127.0.0.1:8199/register"
{"code":1,"error":"","data":null}
```

## 2

\*gvalid.Error

### Content Menu

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

```

package main

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

type RegisterReq struct {
    Name string `p:"username" v:"required|length:6,30#|:min:max"`
    Pass string `p:"password1" v:"required|length:6,30#|"`
    Pass2 string `p:"password2" v:"required|length:6,30|same:password1#||"`
}

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 {
            // Validation error.
            if v, ok := err.(*gvalid.Error); ok {
                r.Response.WriteJsonExit(RegisterRes{
                    Code: 1,
                    Error: v.FirstString(),
                })
            }
            // Other error.
            r.Response.WriteJsonExit(RegisterRes{
                Code: 1,
                Error: err.Error(),
            })
        }
        // ...
        r.Response.WriteJsonExit(RegisterRes{
            Data: req,
        })
    })
    s.SetPort(8199)
    s.Run()
}

err.(*gvalid.Error) -

curl

$ curl "http://127.0.0.1:8199/register"
{"code":1,"error":"","data":null}

$ curl "http://127.0.0.1:8199/register?
name=john&password1=123456&password2=12345"
{"code":1,"error":"","data":null}

```