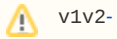


Requestvgvalid -Struct



v1v2-

Content Menu

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

1

v

```
package main

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

// RegisterReq
type RegisterReq struct {
    Name string `p:"username" v:"required|length:4,30#{min}:{max}"`
    Pass string `p:"password1" v:"required|length:6,30#|"`
    Pass2 string `p:"password2" v:"required|length:6,30|same:password1#`
}

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

func main() {
    s := g.Server()
    s.Group("/", func(group *ghttp.RouterGroup) {
        group.ALL("/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



p +

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

```

package main

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

type RegisterReq struct {
    Name string `p:"username" v:"required|length:4,30#{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.Group("/", func(group *ghttp.RouterGroup) {
        group.ALL("/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.FirstError().
Error(),
                                })
                }
                // 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) -



gerror.Current

```

var req *RegisterReq
if err := r.Parse(&req); err != nil {
    r.Response.WriteJsonExit(RegisterRes{
        Code: 1,
        Error: gerror.Current(err).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=1234567"
{"code":1,"error":"","data":null}
```