

-JSON/XML

GoFrame v1.11RequestJSON/XML

1

```
package main

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

func main() {
    s := g.Server()
    s.BindHandler("/", func(r *ghttp.Request) {
        r.Response.Writef("name: %v, pass: %v", r.Get("name"), r.
Get("pass"))
    })
    s.SetPort(8199)
    s.Run()
}
```

curl

1. Query

```
$ curl "http://127.0.0.1:8199/?name=john&pass=123"
name: john, pass: 123
```

2. Form

```
$ curl -d "name=john&pass=123" "http://127.0.0.1:8199/"
name: john, pass: 123
```

3. JSON

```
$ curl -d '{"name":"john","pass":"123"}' "http://127.0.0.1:8199/"
name: john, pass: 123
```

4. XML

```
$ curl -d '<?xml version="1.0" encoding="UTF-8"?><doc><name>john<
/name><pass>123</pass></doc>' "http://127.0.0.1:8199/"
name: john, pass: 123

$ curl -d '<doc><name>john</name><pass>123</pass></doc>' "http://127.
0.0.1:8199/"
name: john, pass: 123
```

2

Content Menu

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

```

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: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()
}

```

curl

1. JSON

```

$ curl -d '{"username":"johngcn","password1":"123456","password2":"123456"}' "http://127.0.0.1:8199/register"
{"code":0,"error":"","data":{"Name":"johngcn","Pass":"123456","Pass2":"123456"}}

$ curl -d '{"username":"johngcn","password1":"123456","password2":"1234567"}' "http://127.0.0.1:8199/register"
{"code":1,"error":"","data":null}

```

JSONParse

2. XML

```
$ curl -d '<?xml version="1.0" encoding="UTF-8"?><doc><username>johngcn</username><password1>123456</password1><password2>123456</password2></doc>' "http://127.0.0.1:8199/register"
{"code":0,"error":"","data":{"Name":"johngcn","Pass":"123456","Pass2":"123456"}}

$ curl -d '<?xml version="1.0" encoding="UTF-8"?><doc><username>johngcn</username><password1>123456</password1><password2>1234567</password2></doc>' "http://127.0.0.1:8199/register"
{"code":1,"error":"","data":null}
```

XMLParse