

GoFrameprefix/

<https://pkg.go.dev/github.com/gogf/gf/v2/net/ghttp#RouterGroup>

```
//
func (s *Server) Group(prefix string, groups ...func(g *RouterGroup)
*RouterGroup

// Method
func (g *RouterGroup) ALL(pattern string, object interface{}, params...
interface{})
func (g *RouterGroup) GET(pattern string, object interface{}, params...
interface{})
func (g *RouterGroup) PUT(pattern string, object interface{}, params...
interface{})
func (g *RouterGroup) POST(pattern string, object interface{}, params...
interface{})
func (g *RouterGroup) DELETE(pattern string, object interface{}, params...
interface{})
...

//
func (g *RouterGroup) Middleware(handlers ...HandlerFunc) *RouterGroup

//
func (g *RouterGroup) Map(m map[string]interface{})
func (g *RouterGroup) ALLMap(m map[string]interface{})
```

1. Group
2. HTTP MethodHTTP MethodALLHTTP Method//RESTRESTful
3. Middleware

```
package main

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

func main() {
    s := g.Server()
    s.Group("/api", func(group *ghttp.RouterGroup) {
        group.ALL("/all", func(r *ghttp.Request) {
            r.Response.Write("all")
        })
        group.GET("/get", func(r *ghttp.Request) {
            r.Response.Write("get")
        })
        group.POST("/post", func(r *ghttp.Request) {
            r.Response.Write("post")
        })
    })
    s.SetPort(8199)
    s.Run()
}
```

## Content Menu



- [Map](#)
- [AllMap](#)

SERVER	DOMAIN	ADDRESS	METHOD	ROUTE	HANDLER
MIDDLEWARE					
default	default	:8199	ALL	/api/all	main.main.func1
default	default	:8199	GET	/api/get	main.main.func2
default	default	:8199	POST	/api/post	main.main.func3

/api/getGET/api/postPOST/api/all

curl

1. /api/get

```
$ curl http://127.0.0.1:8199/api/get
get
$ curl -X POST http://127.0.0.1:8199/api/get
Not Found
```

2. /api/post

```
$ curl http://127.0.0.1:8199/api/post
Not Found
$ curl -X POST http://127.0.0.1:8199/api/post post
```

3. /api/all

```
$ curl http://127.0.0.1:8199/api/all
all
$ curl -X POST http://127.0.0.1:8199/api/all
all
$ curl -X DELETE http://127.0.0.1:8199/api/all
all
$ curl -X OPTIONS http://127.0.0.1:8199/api/all
all
```



GoFramePHP Laravel🤔

GoFrameHOOKHTTP Method

```

package main

import (
    "fmt"
    "net/http"

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

func MiddlewareAuth(r *ghttp.Request) {
    if r.Get("token").String() != "123456" {
        r.Response.WriteStatus(http.StatusForbidden)
        return
    }
    r.Middleware.Next()
}

func MiddlewareCORS(r *ghttp.Request) {
    r.Response.CORSDefault()
    r.Middleware.Next()
}

func MiddlewareLog(r *ghttp.Request) {
    r.Middleware.Next()
    fmt.Println(r.Response.Status, r.URL.Path)
}

func main() {
    s := g.Server()
    s.Use(MiddlewareLog)
    s.Group("/api.v2", func(group *ghttp.RouterGroup) {
        group.Middleware(MiddlewareAuth, MiddlewareCORS)
        group.GET("/test", func(r *ghttp.Request) {
            r.Response.Write("test")
        })
        group.Group("/order", func(group *ghttp.RouterGroup) {
            group.GET("/list", func(r *ghttp.Request) {
                r.Response.Write("list")
            })
            group.PUT("/update", func(r *ghttp.Request) {
                r.Response.Write("update")
            })
        })
        group.Group("/user", func(group *ghttp.RouterGroup) {
            group.GET("/info", func(r *ghttp.Request) {
                r.Response.Write("info")
            })
            group.POST("/edit", func(r *ghttp.Request) {
                r.Response.Write("edit")
            })
            group.DELETE("/drop", func(r *ghttp.Request) {
                r.Response.Write("drop")
            })
        })
        group.Group("/hook", func(group *ghttp.RouterGroup) {
            group.Hook("/*", ghttp.HookBeforeServe, func(r
*ghttp.Request) {
                r.Response.Write("hook any")
            })
            group.Hook("/:name", ghttp.HookBeforeServe, func(r
*ghttp.Request) {
                r.Response.Write("hook name")
            })
        })
    })
    s.SetPort(8199)
    s.Run()
}

```

SERVER HANDLER	DOMAIN	ADDRESS	METHOD	ROUTE	MIDDLEWARE
default	default	:8199	ALL	/*	main. MiddlewareLog   GLOBAL MIDDLEWARE
default	default	:8199	ALL	/api.v2/hook/*	main.main. func1.4.1   HOOK_BEFORE_SERVE
default	default	:8199	ALL	/api.v2/hook/:name	main.main. func1.4.2   HOOK_BEFORE_SERVE
default	default	:8199	GET	/api.v2/order/list	main.main. func1.2.1   main.MiddlewareAuth,main.MiddlewareCORS
default	default	:8199	PUT	/api.v2/order/update	main.main. func1.2.2   main.MiddlewareAuth,main.MiddlewareCORS
default	default	:8199	GET	/api.v2/test	main.main. func1.1   main.MiddlewareAuth,main.MiddlewareCORS
default	default	:8199	DELETE	/api.v2/user/drop	main.main. func1.3.3   main.MiddlewareAuth,main.MiddlewareCORS
default	default	:8199	POST	/api.v2/user/edit	main.main. func1.3.2   main.MiddlewareAuth,main.MiddlewareCORS
default	default	:8199	GET	/api.v2/user/info	main.main. func1.3.1   main.MiddlewareAuth,main.MiddlewareCORS

## Map

MapURIHTTP MethodHTTP Method

```

package main

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

func UserGet(r *ghttp.Request) {
    r.Response.Write("get")
}

func UserDelete(r *ghttp.Request) {
    r.Response.Write("delete")
}

func main() {
    s := g.Server()
    s.Group("/api", func(group *ghttp.RouterGroup) {
        group.Map(g.Map{
            "GET":    /user": UserGet,
            "DELETE": /user": UserDelete,
        })
    })
    s.SetPort(8199)
    s.Run()
}

```

## AllMap

ALLMap/HTTP Method

```

s := g.Server()
//
s.Group("/", func(group *ghttp.RouterGroup) {
    group.Middleware(service.Middleware.Ctx)
    group.ALLMap(g.Map{
        "/":          api.Index,          //
        "/login":     api.Login,          //
        "/register":  api.Register,        //
        "/category":  api.Category,        //
        "/topic":     api.Topic,          //
        "/topic/:id": api.Topic.Detail,    // -
        "/ask":       api.Ask,            //
        "/ask/:id":   api.Ask.Detail,      // -
        "/article":   api.Article,        //
        "/article/:id": api.Article.Detail, // -
        "/reply":     api.Reply,          //
        "/search":    api.Search,         //
        "/captcha":   api.Captcha,        //
        "/user/:id":  api.User.Index,     // -
    })
    //
    group.Group("/", func(group *ghttp.RouterGroup) {
        group.Middleware(service.Middleware.Auth)
        group.ALLMap(g.Map{
            "/user":    api.User,    //
            "/content": api.Content, //
            "/interact": api.Interact, //
            "/file":     api.File,     //
        })
    })
})

```

