

# CORS

-/ WebSocket

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

```
func (r *Response) CORS(options CORSOptions)
func (r *Response) CORSAllowedOrigin(options CORSOptions) bool
func (r *Response) CORSDefault()
func (r *Response) DefaultCORSOptions() CORSOptions
```

## CORS

CORSW3HTTPghttpCORSOptions

```
// See https://www.w3.org/TR/cors/ .
//
type CORSOptions struct {
    AllowDomain []string // Used for allowing requests from
custom domains
    AllowOrigin      string // Access-Control-Allow-Origin
    AllowCredentials string // Access-Control-Allow-Credentials
    ExposeHeaders    string // Access-Control-Expose-Headers
    MaxAge           int    // Access-Control-Max-Age
    AllowMethods     string // Access-Control-Allow-Methods
    AllowHeaders     string // Access-Control-Allow-Headers
}
```

W3

## CORS

### CORSOptions

ghttpDefaultCORSOptionsCORSDefault()

### Origin

AllowDomain

```
//
func Middleware(r *ghttp.Request) {
    corsOptions := r.Response.DefaultCORSOptions()
    corsOptions.AllowDomain = []string{"goframe.org", "johng.cn"}
    r.Response.CORS(corsOptions)
    r.Middleware.Next()
}
```

## OPTIONS

AJAXOPTIONSGFServerW3COPTIONSCORSOPTIONS

1

### Content Menu

- [CORS](#)
- [CORS](#)
  - [CORSOptions](#)
  - [Origin](#)
- [OPTIONS](#)
- [1](#)
- [2Origin](#)
- [3](#)

```
package main

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

func Order(r *ghttp.Request) {
    r.Response.Write("GET")
}

func main() {
    s := g.Server()
    s.Group("/api.v1", func(group *ghttp.RouterGroup) {
        group.GET("/order", Order)
    })
    s.SetPort(8199)
    s.Run()
}
```

<http://localhost/api.v1/order> (jQuery)F12consoleAJAX

```
$.get("http://localhost:8199/api.v1/order", function(result){
    console.log(result)
});
```



```
package main

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

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

func Order(r *ghttp.Request) {
    r.Response.Write("GET")
}

func main() {
    s := g.Server()
    s.Group("/api.v1", func(group *ghttp.RouterGroup) {
        group.Middleware(MiddlewareCORS)
        group.GET("/order", Order)
    })
    s.SetPort(8199)
    s.Run()
}
```

[/api.v1MiddlewareCORSCORSDefault/api.v1](#)

## AJAX



CORSOptionsCORS

## 2Origin

Origingoframe.orgbaidu.com

```
package main

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

func MiddlewareCORS(r *ghttp.Request) {
    corsOptions := r.Response.DefaultCORSOptions()
    corsOptions.AllowDomain = []string{"goframe.org", "baidu.com"}
    r.Response.CORS(corsOptions)
    r.Middleware.Next()
}

func Order(r *ghttp.Request) {
    r.Response.Write("GET")
}

func main() {
    s := g.Server()
    s.Group("/api.v1", func(group *ghttp.RouterGroup) {
        group.Middleware(MiddlewareCORS)
        group.GET("/order", Order)
    })
    s.SetPort(8199)
    s.Run()
}
```

## 3

OriginCORSAllowedOriginOrigin

goframe.org403

```
package main

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

func MiddlewareCORS(r *ghttp.Request) {
    corsOptions := r.Response.DefaultCORSOptions()
    corsOptions.AllowDomain = []string{"goframe.org"}
    if !r.Response.CORSAllowedOrigin(corsOptions) {
        r.Response.WriteStatus(http.StatusForbidden)
        return
    }
    r.Response.CORS(corsOptions)
    r.Middleware.Next()
}

func Order(r *ghttp.Request) {
    r.Response.Write("GET")
}

func main() {
    s := g.Server()
    s.Group("/api.v1", func(group *ghttp.RouterGroup) {
        group.Middleware(MiddlewareCORS)
        group.GET("/order", Order)
    })
    s.SetPort(8199)
    s.Run()
}
```