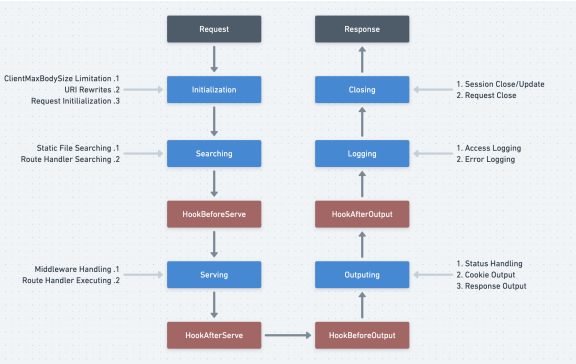
HOOK



ghttp.Server

ghttp.Serverpattern(pattern)ghttp.ServerHOOK

```
func (s *Server) BindHookHandler(pattern string, hook string, handler
HandlerFunc) error
func (s *Server) BindHookHandlerByMap(pattern string, hookmap map[string]
HandlerFunc) error
```

```
func (d *Domain) BindHookHandler(pattern string, hook string, handler
HandlerFunc) error
func (d *Domain) BindHookHandlerByMap(pattern string, hookmap map[string]
HandlerFunc) error
```

Hook

1. ghttp.HookBeforeServe

   /
2. ghttp.HookAfterServe

3. ghttp.HookBeforeOutput

4. ghttp.HookAfterOutput

-

/*HOOKHOOK/*

HOOK(/user/*)HOOK

() (: ABC)HOOK(: A-B-C)

HOOK

Content Menu

ExitHook

HOOKHOOKHOOKRequest.ExitHookHOOKHOOK

/ghttp.HookBeforeServe(/*)r.ExitAll()()

r.Redirect*r.ExitAll()http multiple response writeheader callsr.Redirect*
headerLocationheaderContent-Type/Content-Length

MiddlewareHOOKGF

1. “”Server
2. “”
3.

Request.URLRequest.Router

Request.Router Request.URLURLhttp.RequestURLRequest.URL.PathURI

Request.RouternilRequest.URLURLRequest.Router

APInginx

gfWebServer

/*/static/js/index.js/upload/images/thumb.jpg

Request.IsFileRequest()

1

```
package main

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

func main() {
        //
        p := "/:name/info/{uid}"
        s := g.Server()
        s.BindHookHandlerByMap(p, map[string]ghttp.HandlerFunc{
                ghttp.HookBeforeServe:  func(r *ghttp.Request) { glog.
Println(ghttp.HookBeforeServe) },
                ghttp.HookAfterServe:   func(r *ghttp.Request) { glog.
Println(ghttp.HookAfterServe) },
                ghttp.HookBeforeOutput: func(r *ghttp.Request) { glog.
Println(ghttp.HookBeforeOutput) },
                ghttp.HookAfterOutput:  func(r *ghttp.Request) { glog.
Println(ghttp.HookAfterOutput) },
        })
        s.BindHandler(p, func(r *ghttp.Request) {
                r.Response.Write(":", r.Get("name"), ", uid:", r.Get
("uid"))
        })
        s.SetPort(8199)
        s.Run()
}
```

http://127.0.0.1:8199/john/info/10000 WebServer

2

```
package main

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

// HOOK
func beforeServeHook1(r *ghttp.Request) {
        r.SetParam("name", "GoFrame")
        r.Response.Writeln("set name")
}

// HOOK
func beforeServeHook2(r *ghttp.Request) {
        r.SetParam("site", "https://goframe.org")
        r.Response.Writeln("set site")
}

//
// HOOK
func main() {
        s := g.Server()
        s.BindHandler("/", func(r *ghttp.Request) {
                r.Response.Writeln(r.Get("name"))
                r.Response.Writeln(r.Get("site"))
        })
        s.BindHookHandler("/", ghttp.HookBeforeServe, beforeServeHook1)
        s.BindHookHandler("/", ghttp.HookBeforeServe, beforeServeHook2)
        s.SetPort(8199)
        s.Run()
}
```

```
SERVER  | ADDRESS | DOMAIN  | METHOD | P | ROUTE |        HANDLER
|    MIDDLEWARE
|---------|---------|---------|--------|---|-------|-----------------------
|-------------------|
  default |  :8199  | default | ALL    | 1 | /     | main.main.func1
|
|---------|---------|---------|--------|---|-------|-----------------------
|-------------------|
  default |  :8199  | default | ALL    | 2 | /     | main.beforeServeHook1
| HOOK_BEFORE_SERVE
|---------|---------|---------|--------|---|-------|-----------------------
|-------------------|
  default |  :8199  | default | ALL    | 1 | /     | main.beforeServeHook2
| HOOK_BEFORE_SERVE
|---------|---------|---------|--------|---|-------|-----------------------
|-------------------|
```

http://127.0.0.1:8199/

```
set name
set site
GoFrame
https://goframe.org
```

```
package main

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

func main() {
        s := g.Server()
        // 1
        pattern1 := "/:name/info"
        s.BindHookHandlerByMap(pattern1, map[string]ghttp.HandlerFunc{
                ghttp.HookBeforeServe: func(r *ghttp.Request) {
                        r.SetParam("uid", 1000)
                },
        })
        s.BindHandler(pattern1, func(r *ghttp.Request) {
                r.Response.Write(":", r.Get("name"), ", uid:", r.Get
("uid"))
        })

        // 2
        pattern2 := "/{object}/list/{page}.java"
        s.BindHookHandlerByMap(pattern2, map[string]ghttp.HandlerFunc{
                ghttp.HookBeforeOutput: func(r *ghttp.Request) {
                        r.Response.SetBuffer([]byte(
                                fmt.Sprintf(", object:%s, page:%s", r.Get
("object"), r.GetRouterString("page"))),
                        )
                },
        })
        s.BindHandler(pattern2, func(r *ghttp.Request) {
                r.Response.Write(r.Router.Uri)
        })
        s.SetPort(8199)
        s.Run()
}
```

1/:name/infoGET2/{object}/list/{page}.javaURL

- http://127.0.0.1:8199/user/info
- http://127.0.0.1:8199/user/list/1.java
- http://127.0.0.1:8199/user/list/2.java

4

```
package main

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

func main() {
        s := g.Server()
        s.BindHandler("/priority/show", func(r *ghttp.Request) {
                r.Response.Writeln("priority service")
        })

        s.BindHookHandlerByMap("/priority/:name", map[string]ghttp.
HandlerFunc{
                ghttp.HookBeforeServe: func(r *ghttp.Request) {
                        r.Response.Writeln("/priority/:name")
                },
        })
        s.BindHookHandlerByMap("/priority/*any", map[string]ghttp.
HandlerFunc{
                ghttp.HookBeforeServe: func(r *ghttp.Request) {
                        r.Response.Writeln("/priority/*any")
                },
        })
        s.BindHookHandlerByMap("/priority/show", map[string]ghttp.
HandlerFunc{
                ghttp.HookBeforeServe: func(r *ghttp.Request) {
                        r.Response.Writeln("/priority/show")
                },
        })
        s.SetPort(8199)
        s.Run()
}
```

3/priority/show

http://127.0.0.1:8199/priority/show

```
/priority/show
/priority/:name
/priority/*any
priority service
```

5

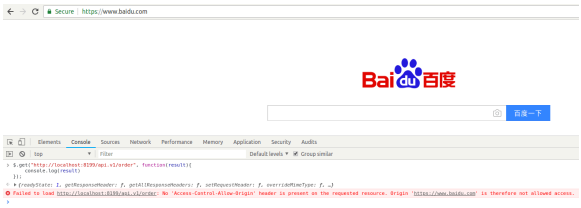
-/ CORS

HOOKHOOK

```
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:8199/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 Order(r *ghttp.Request) {
        r.Response.Write("GET")
}

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

/api.v1/*anyghttp.HookBeforeServeCORSDefault/api.v1

AJAX


Secure
https://www.baidu.com
Baidu 百度
百度一下
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
Group similar
$.get("http://localhost:8190/api.v1/order", function(result){
console.log(result)
});
{readyState: 1, getResponseHeader: ƒ, getAllResponseHeaders: ƒ, setRequestHeader: ƒ, overrideMimeType: ƒ, …}
Failed to load http://localhost:8190/api.v1/order: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'https://www.baidu.com' is therefore not allowed access.
$.get("http://localhost:8190/api.v1/order", function(result){
console.log(result)
});
{readyState: 1, getResponseHeader: ƒ, getAllResponseHeaders: ƒ, setRequestHeader: ƒ, overrideMimeType: ƒ, …}
GET