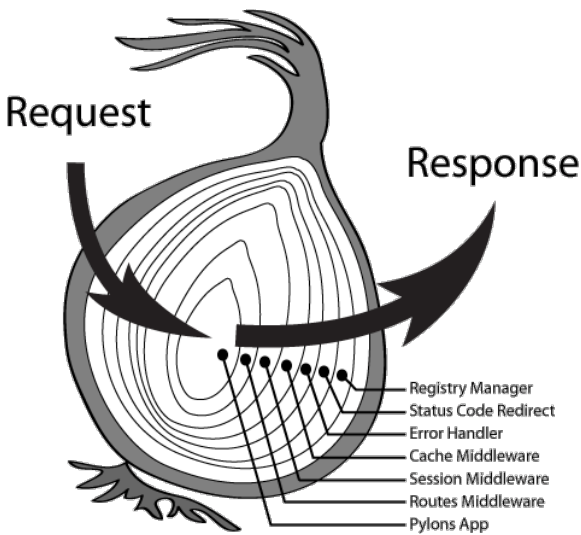


/-

GoFrameWebServerWebServer



Content Menu

- -
 -
 -
 -
- -
 -
 -
 -

HTTPHandlerFuncRequestMiddleware

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

r.Middleware.Next()r.Middleware.Next()

```
func Middleware(r *ghttp.Request) {  
    //  
    r.Middleware.Next()  
}
```

```
func Middleware(r *ghttp.Request) {  
    r.Middleware.Next()  
    //  
}
```

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

```
func (s *Server) Use(handlers ...HandlerFunc)
```

Server



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

Middleware

- 1.
- 2.
- 3.



gRPC

```
package main

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

type HelloReq struct {
    g.Meta `path:"/hello" method:"get"`
}
type HelloRes struct {
}

type Hello struct{}

func (Hello) Say(ctx context.Context, req *HelloReq) (res *HelloRes, err error) {
    glog.Debug(ctx, "")
    return
}

func RequestHandle1(r *ghttp.Request) {
```

```

        glog.Debug(r.GetCtx(), "1")
        r.Middleware.Next()
    }

    func RequestHandle2(r *ghttp.Request) {
        glog.Debug(r.GetCtx(), "2")
        r.Middleware.Next()
    }

    func RequestHandle3(r *ghttp.Request) {
        glog.Debug(r.GetCtx(), "3")
        r.Middleware.Next()
    }

    func RequestHandle4(r *ghttp.Request) {
        glog.Debug(r.GetCtx(), "4")
        r.Middleware.Next()
    }

    func ResponseHandle1(r *ghttp.Request) {
        r.Middleware.Next()
        glog.Debug(r.GetCtx(), "1")
    }

    func ResponseHandle2(r *ghttp.Request) {
        r.Middleware.Next()
        glog.Debug(r.GetCtx(), "2")
    }

    func ResponseHandle3(r *ghttp.Request) {
        r.Middleware.Next()
        glog.Debug(r.GetCtx(), "3")
    }

    func ResponseHandle4(r *ghttp.Request) {
        r.Middleware.Next()
        glog.Debug(r.GetCtx(), "4")
    }

    func main() {
        s := g.Server()
        s.Use(ghttp.MiddlewareHandlerResponse)
        s.Group("/", func(group *ghttp.RouterGroup) {
            //
            group.Middleware(RequestHandle1)
            group.Middleware(RequestHandle2)

            //
            group.Middleware(ResponseHandle1)
            group.Middleware(ResponseHandle2)

            group.Group("/sub", func(group *ghttp.RouterGroup) {
                //
                group.Middleware(RequestHandle3)
                group.Middleware(RequestHandle4)

                //
                group.Middleware(ResponseHandle3)
                group.Middleware(ResponseHandle4)

                group.Bind(new>Hello))
            })
        })
        s.Run()
    }

```

```
2023-09-12 10:13:13.133 [DEBU] {8444030cf10484177459495dba2a6880} 前 1
2023-09-12 10:13:13.133 [DEBU] {8444030cf10484177459495dba2a6880} 前 2
2023-09-12 10:13:13.134 [DEBU] {8444030cf10484177459495dba2a6880} 前 3
2023-09-12 10:13:13.134 [DEBU] {8444030cf10484177459495dba2a6880} 前 4
2023-09-12 10:13:13.135 [DEBU] {8444030cf10484177459495dba2a6880} 中
2023-09-12 10:13:13.135 [DEBU] {8444030cf10484177459495dba2a6880} 后 4
2023-09-12 10:13:13.135 [DEBU] {8444030cf10484177459495dba2a6880} 后 3
2023-09-12 10:13:13.135 [DEBU] {8444030cf10484177459495dba2a6880} 后 2
2023-09-12 10:13:13.135 [DEBU] {8444030cf10484177459495dba2a6880} 后 1
```

`Next()return`

- `RequestHandle1Next()RequestHandle1`
- `RequestHandle2Next()RequestHandle1RequestHandle2`
- `RequestHandle3Next()RequestHandle1RequestHandle2RequestHandle3ResponseHandle2ResponseHandle1`
- `RequestHandle4Next()RequestHandle1RequestHandle2RequestHandle3RequestHandle4ResponseHandle2ResponseHandle1`

`returnExit-Exit`