HTTPClient-/

Content Menu

HTTPClient///

```
func (c *Client) Use(handlers ...HandlerFunc) *Client
```

NextNext

```
func (c *Client) Next(req *http.Request) (*Response, error)
```

HTTPClientHTTPServer

Next

```
c := g.Client()
c.Use(func(c *gclient.Client, r *http.Request) (resp *gclient.Response,
err error) {
        //
        resp, err = c.Next(r)
    return resp, err
})
```

Next

```
c := g.Client()
c.Use(func(c *gclient.Client, r *http.Request) (resp *gclient.Response,
err error) {
        resp, err = c.Next(r)
    //
    return resp, err
})
```

JSON

JSONmapJSON

```
package main

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

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

```
package main

import (
        "bytes"
        "fmt"
        "io/ioutil"
        "net/http"

        "github.com/gogf/gf/v2/container/garray"
        "github.com/gogf/gf/v2/crypto/gmd5"
        "github.com/gogf/gf/v2/frame/g"
        "github.com/gogf/gf/v2/internal/json"
        "github.com/gogf/gf/v2/net/gclient"
        "github.com/gogf/gf/v2/os/gctx"
        "github.com/gogf/gf/v2/os/gtime"
        "github.com/gogf/gf/v2/util/gconv"
        "github.com/gogf/gf/v2/util/guid"
        "github.com/gogf/gf/v2/util/gutil"
)

const (
        appId     = "123"
        appSecret = "456"
)

//
func injectSignature(jsonContent []byte) []byte {
        var m map[string]interface{}
        _ = json.Unmarshal(jsonContent, &m)
        if len(m) > 0 {
                m["appid"] = appId
                m["nonce"] = guid.S()
                m["timestamp"] = gtime.Timestamp()
                var (
                        keyArray   = garray.NewSortedStrArrayFrom(gutil.
Keys(m))
                        sigContent string
                )
                keyArray.Iterator(func(k int, v string) bool {
                        sigContent += v
                        sigContent += gconv.String(m[v])
                        return true
                })
                m["signature"] = gmd5.MustEncryptString(gmd5.
MustEncryptString(sigContent) + appSecret)
```

```
                jsonContent, _ = json.Marshal(m)
        }
        return jsonContent
}

func main() {
        c := g.Client()
        c.Use(func(c *gclient.Client, r *http.Request) (resp *gclient.
Response, err error) {
                bodyBytes, _ := ioutil.ReadAll(r.Body)
                if len(bodyBytes) > 0 {
                        // Request
                        bodyBytes = injectSignature(bodyBytes)
                        r.Body = ioutil.NopCloser(bytes.NewBuffer
(bodyBytes))
                        r.ContentLength = int64(len(bodyBytes))
                }
                return c.Next(r)
        })
        content := c.ContentJson().PostContent(gctx.New(), "http://127.
0.0.1:8199/", g.Map{
                "name": "goframe",
                "site": "https://goframe.org",
        })
        fmt.Println(content)
}
```

```
$ go run server.go

  SERVER  | DOMAIN  | ADDRESS | METHOD | ROUTE |      HANDLER      |
MIDDLEWARE
----------|---------|---------|--------|-------|-------------------
|-------------
  default | default | :8199   | ALL    | /     | main.main.func1.1
|
----------|---------|---------|--------|-------|-------------------
|-------------

2021-05-18 09:23:41.865 97906: http server started listening on [:8199]
```

```
$ go run client.go
{"appid":"123","name":"goframe","nonce":"
12vd8tx23l6cbfz9k59xehk1002pixfo","signature":"
578a90b67bdc63d551d6a18635307ba2","site":"https://goframe.org","timestamp":
1621301076}
$
```

appid/nonce/timestamp/signature