

HTTPClient-Cookie

HTTPCookieSetCookie*

```
func (c *Client) SetCookie(key, value string) *Client
func (c *Client) SetCookieMap(m map[string]string) *Client
```

Content Menu

-
-

Cookie

```
package main

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

func main() {
    s := g.Server()
    s.BindHandler("/", func(r *ghttp.Request){
        r.Response.Write(r.Cookie.Map())
    })
    s.SetPort(8199)
    s.Run()
}
```

Cookie

1. SetCookie

```
package main

import (
    "fmt"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    c := g.Client()
    c.SetCookie("name", "john")
    c.SetCookie("score", "100")
    if r, e := c.Get(gctx.New(), "http://127.0.0.1:8199/"); e !=
nil {
        panic(e)
    } else {
        fmt.Println(r.ReadAllString())
    }
}
```

g.Client()HTTPc.SetCookieCookieCookienamescore

2. SetCookieMap

Cookie

```
package main

import (
    "fmt"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    c := g.Client()
    c.SetCookieMap(g.MapStrStr{
        "name": "john",
        "score": "100",
    })
    if r, e := c.Get(gctx.New(), "http://127.0.0.1:8199/"); e !=
nil {
        panic(e)
    } else {
        fmt.Println(r.ReadAllString())
    }
}
```

3.

```
map[name:john score:100]
```

Cookie