

- 1.
- 2.

## Content Menu

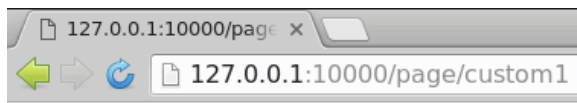
•  
•

```
package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/gview"
    "github.com/gogf/gf/v2/text/gstr"
    "github.com/gogf/gf/v2/util/gpage"
)

// wrapContent wraps each of the page tag with html li and ul.
func wrapContent(page *gpage.Page) string {
    content := page.GetContent(4)
    content = gstr.ReplaceByMap(content, map[string]string{
        "<span": " <li><span",
        "/span>": "/span></li>",
        "<a": " <li><a",
        "/a>": " /a></li>",
    })
    return "<ul>" + content + "</ul>"
}

func main() {
    s := g.Server()
    s.BindHandler("/page/custom1/*page", func(r *ghttp.Request) {
        page := r.GetPage(100, 10)
        content := wrapContent(page)
        buffer, _ := gview.ParseContent(`
<html>
    <head>
        <style>
            a,span {padding:8px; font-size:16px;}
            div{margin:5px 5px 20px 5px}
        </style>
    </head>
    <body>
        <div>{{.page}}</div>
    </body>
</html>
`, g.Map{
            "page": content,
        })
        r.Response.Write(buffer)
    })
    s.SetPort(10000)
    s.Run()
}
```



- [首页](#)
- [上一页](#)
- [1](#)
- [2](#)
- [3](#)
- [4](#)
- [5](#)
- [6](#)
- [7](#)
- [8](#)
- [9](#)
- [10](#)
- [下一页](#)
- [尾页](#)

```

package main

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

// pageContent customizes the page tag name.
func pageContent(page *gpage.Page) string {
    page.NextPageTag = "NextPage"
    page.PrevPageTag = "PrevPage"
    page.FirstPageTag = "HomePage"
    page.LastPageTag = "LastPage"
    pageStr := page.FirstPage()
    pageStr += page.PrevPage()
    pageStr += page.PageBar()
    pageStr += page.NextPage()
    pageStr += page.LastPage()
    return pageStr
}

func main() {
    s := g.Server()
    s.BindHandler("/page/custom2/*page", func(r *ghttp.Request) {
        page := r.GetPage(100, 10)
        buffer, _ := gview.ParseContent(`
<html>
    <head>
        <style>
            a,span {padding:8px; font-size:16px;}
            div{margin:5px 5px 20px 5px}
        </style>
    </head>
    <body>
        <div>{{.page}}</div>
    </body>
</html>
`, g.Map{
            "page": pageContent(page),
        })
        r.Response.Write(buffer)
    })
    s.SetPort(10000)
    s.Run()
}

```

