

- 1.
- 2.
3. layout
- 4.
- 5.
- 6.
- 7.
- 8.
9. define/template
10. include

Content Menu

-
-
- 1
- 2
- 3
-
-
-

```
gviewgview.Instanceeg.View()gview
```

<https://godoc.org/github.com/gogf/gf/os/gview>

1. gview.Get
2. gview.New
3. SetPath/AddPath/SetPathAddPath
4. Assign/Assigns
5. BindFunc
6. Parse/ParseContent/
7. SetDelimiters{{ }}vuejs

1

```
index.tpl
id:{{.id}}, name:{{.name}}

main.go
package main

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

func main() {
    s := g.Server()
    s.BindHandler("/template", func(r *ghttp.Request) {
        r.Response.WriteTpl("index.tpl", g.Map{
            "id": 123,
            "name": "john",
        })
    })
    s.SetPort(8199)
    s.Run()
}
```

<http://127.0.0.1:8199/template> id:123, name:john

2

```

package main

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

func main() {
    s := g.Server()
    s.BindHandler("/template", func(r *ghttp.Request){
        tplContent := `id:{{.id}}, name:{{.name}}`
        r.Response.WriteTplContent(tplContent, g.Map{
            "id"    : 123,
            "name"  : "john",
        })
    })
    s.SetPort(8199)
    s.Run()
}

```

<http://127.0.0.1:8199/template> id:123, name:john

3

Golangvue{{ }}SetDelimitersGolang

```

// main.go
package main

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

func main() {
    v := g.View()
    v.SetDelimiters("${", "}")
    b, err := v.Parse("gview_delimiters.tpl", map[string]interface{} {
        "k" : "v",
    })
    fmt.Println(err)
    fmt.Println(string(b))
}

<!-- gview_delimiters.tpl -->
test.tpl content, vars: ${.}

<nil>
test.tpl content, vars: map[k:v]

gfSetPathAddPath

gview

1. template/home/www/home/www/home/www/template
2. template/tmp/tmp/tmp/template
3. maintemplatemain/home/john/workspace/gf-app/home/john/workspace/gf-app/
   home/john/workspace/gf-app/template

1. ViewSetPath
2. - gf.gview.path
3. - GF_GVIEW_PATH

main(Linux)

1.

```

```
g.View().SetPath("/opt/template")
```

2.

```
./main --gf.gview.path /opt/template/
```

3.

-

```
GF_GVIEW_PATH /opt/config/ ./main
```

- genv

```
genv.Set("GF_GVIEW_PATH", "/opt/template")
```

gf