

GoFrame

<https://pkg.go.dev/github.com/gogf/v2/os/gview#Config>

`g.View()ViewViewviewer.viewer`

```
[viewer]
  Paths      = ["/var/www/template"] #
  DefaultFile = "index.html"         # "index.html"
  Delimiters  = ["${", "}"]          # [{"{", "}"]}
  AutoEncode  = false                # XSSfalse
  [viewer.Data]
    Key1 = "Value1"                  # Key-Value
    Key2 = "Value2"
```

1

```
[viewer]
  paths      = ["template", "/var/www/template"]
  defaultFile = "index.html"
  delimiters  = ["${", "}"]
  [viewer.data]
    name      = "gf"
    version   = "1.10.0"
```

`g.View()`

2

View

```
[viewer]
  paths      = ["template", "/var/www/template"]
  defaultFile = "index.html"
  delimiters  = ["${", "}"]
  [viewer.data]
    name      = "gf"
    version   = "1.10.0"
  [viewer.view1]
    defaultFile = "layout.html"
    delimiters  = ["${", "}"]
  [viewer.view2]
    defaultFile = "main.html"
    delimiters  = [{"#", "}"]
```

View

Content Menu

-
- - 1
 - 2
 -
 - [SetConfigWithMap](#)
-

```
// viewer.view1
v1 := g.View("view1")
// viewer.view2
v2 := g.View("view2")
// viewer
v3 := g.View("none")
// viewer
v4 := g.View()
```

<https://pkg.go.dev/github.com/gogf/gf/v2/os/gview>

1. SetConfigSetConfigWithMap
2. ViewSet*
3. View

SetConfigWithMap

SetConfigWithMapKey-Value/ViewKeyConfigstruct-/ -Struct

```
package main

import (
    "context"
    "fmt"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gview"
)

func main() {
    view := gview.New()
    view.SetConfigWithMap(g.Map{
        "Paths":      []string{"template"},
        "DefaultFile": "index.html",
        "Delimiters": []string{"${", "}"},
        "Data": g.Map{
            "name":    "gf",
            "version": "1.10.0",
        },
    })
    result, err := view.ParseContent(context.TODO(), "hello ${.name},
version: ${.version}")
    if err != nil {
        panic(err)
    }
    fmt.Println(result)
}
```

DefaultFiledefaultFile, default-File, default_file, default file

delimiters["\${", "}"][["{"", "}"}}"]