



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

Add

- Addcontentprefix
-

```
func Add(content string, prefix ...string) error
```

-

Content Menu

- [Add](#)
- [Load](#)
- [Get](#)
- [GetWithIndex](#)
- [GetContent](#)
- [Contains](#)
- [IsEmpty](#)
- [ScanDir](#)
- [ScanDirFile](#)
- [Export](#)
- [Dump](#)

```
package main

import "github.com/gogf/gf/v2/os/gres"

func main() {
    if err := gres.Add("H4sIAAAAAAAC
/5TWVlTtydsH8B+EpoAginEBDVIEQYp0kUTpSjAJhCwsxSwlgEtRwVcKkKwJS09IWCCChS
AkoTZoivfeF0JEEQpMgvQrvcc/xlX/Zi/9zMtFzzPfMc/OZQcFBLKcBDoAD6DTsRQE
/FS9wDLC
/64xz13PHuXng3GRt3R5iTfgBJt6X83cexsJLqjT4mrd9w0fQ6OPh+uAVEbGTj1k4TK1b
dBu6nrT2Sab5GyC7ao5yc61313Ly29Yfbl6fSovmph6AbK6L+TBrhl03yQh09w7hHUCIc
WSAEJd
/z6I0MtJHaaKscBab6STB5unZoVdhgDIFqQaCzyEuLwhza87xLoKCvMJtmjmYMYn63Sjx
/X3IuE66VuXJKcVB08OurMS3IObBV6ilar8HjNhjTAGCzixCKecGJI4HfU3NBhngfflmm
vfuihCWagSzqpwiiwaLhGkW8ZUMtrUOV0nZWSiGUl8ADw2xVAvrVMgUy2Fdmmmo4n15CL
KwynbmI/hlC5qZSd5INHqjEshnzRH8Sy60vatCdrS2MpPTWqGMEQEfExDuC0T6
/aRAmQZc8NLZ/0revTVbeKKswKpFFBw4NFq4J7WWY2yE9p6i/oLwNCCT2HkioAirGf
/cEcnDGH+6SCBPXAaCsiyOHNOMrfy8MpzqcyqbmcIUh9NqM3d240wHkieqIsLVQvYCCrF
HtxYwQsjoEQNW6VrAbc
/d9estPr1Cspt8kdMzResBrnITixfkXV+dH6IvwaET9ilfEFbKv2SHOHvYj8S86uPXU3z
Wv6LnglKVf37PpeU0WIVfU/lJFEJS++4ZAPt2x7NU1Ci+DwKxJCcWDN2IknmvjBEBoro
/g0UJw2rLTQ17sct42H05ZJlKHyoWakNRBajN4lEB7di4EUDnXH9yfEp40qd719eKB6BR
sV2o63HvMbKsgno9JmOKCeTCKsmLKpQdg+XwplYW9fUSi1IggemeV1
/710WoQUzxyPavxZc3selg7g4r9fS8uaZYvb7d0QmPlQgof6MyYqy8ojAytWI2CZkYyHD
erexvnz/THI3+bNekf4Tedlwm4CRRXXwPpNvaf/RKdEvJ0PVwlh4zbCdtZVH5
/Y3nlk9cid3laPS+v4wPytp7FgDk2t6hUhHQ3pUryfB0516FWr8Y500QUWOCdzmfMJAGR
5zDKIizdPHTiZBLNr2KKSEiQdJ/XyDq790DfrfjEmWdw
/fqATXWcJqy8GeaC4g3dVTUAXU0vjc1n3Rcwwvv+Rpt2iKQStpOVGrPat55gbqaiag8Lg
BNkRdj3T7nBUE1+OB2o7aCJNRjzP7QFukFSRqGONqF7atay17bt2payyZJ5YEWuBmVnMC
7KqjKHXLUCj0y3ZTgsnp3ZyPmc7U7Ar1Jj5cckJfjJFD3PM6ZL7o7zuVxt5eZWttT8do+
Lwy+5fzU63SjjL4lMk9fzWuuui63PlnLWsLF19Y+NmEDbqWVPDu3UZuzgWUCgKMjFJydY3
YuFzkBAGAHVGD4H4xwwnn
/vxH0g6p8+cPMLL0pmJqUvac3hnndgyJUmS8oRH7R8yrZw65fJCKuLSzOHKykcDwIGoL0
zB2pnefaBe+QnvRMsv25wEJjYk21l6E6yahC5s051G18yKcNwSdom5bmPkaKirYs9P0SSM
9wXevPii0iGUv2PxJTM8bRPKsfL11fVkl06kqX0Exhj3mnyDZf1FMCqqZ9sinQaph574+
UOFgU3H0uIwNuVcYbrZ2S06elo89E8pajoz3OZCPHLq9Ikd4M8GjMX0TsiFdasw8gQumb
czMQLiVK64At+IE09jrKshRT0FtiHS97tHnJXik2CrTTmKLzeEigV2yHBamd1Vzt
/Oy+uY388ttv0KnXsEJHtLZdrDZS7HDyhpvc3oMhdPg7KClBBw1Hi5HY11hxb1Zrd33L
YKE7Dfbs61cLeyliUt7Jt++vi4dq3VXJ8ctsGM6ylFyVlfsietOLY6+Axns8mbGLdyFY2
8cyncgHm4KfXIK1Bo9gZeKdGrmdngtkd6WoJ8aDksDwluwbCXZX1MkUC9JW8
/G4UPxBltxXLbVrnrCjg
/gOr1hlRKEnrZZY8adJ5KcebmdIkRpi9C8+fKrIImcV8QL8UeTe8uGHNTbyMpmSpLG80n
/xvUXIQ00r7Qe7xMgVKZqecduOPnrznT9G2FU7cAWnfc85917PAm65sBNTB84QJueNjR
vxFoVKWQqZ6ngI9Qi2Faj8t3lQvNxMprPhz13oT+dd6pfXUHySH34lvNyUT6plmYFEMu
U3DCIr628SKmRGY00ZqsdwfpdoybrHg2HEX3Y8Du2vsvB08nypJIn1
/Xerq8icMliM7HK+LRZ2KYeFnGLnBtk4UXIljwBTrPsNDUkWmyhtquc055AbRDTqOqtUf
K6adlDC2AreiOSUzDUHw2bZK0vG
/uLedasI62TKvzrOdCTqz3Fb2m0UUsRFNC7ORz56dyDxWk0agq
/fh2QfcorOXDhTuqJHqUy7uVce60IDcvI8ZJZinotfsta+Li0HUFgweztjFPM4It8cq0+
H5pa2KwcTKc3Ihd0R72ssr+fQnzhlWmPJhq6CYItPnckjp9nuHkSnRmzym6C2GN5I8I8W
VbWKju75d+OvbfLDghD25N2GIFMzPee7ngK5wwbJ7cGar8NJqalr8gEM053Lh9BKii86
rjgew7aqAoS
/f33+2YdCXrDwEM7kkAzBhl4d3UX2HGTXkQ1iz1RpJLp2okp8fL9aJt9zWTnvAEncndWB
9RusvDMeNiehnp
/zO86fTyUk+BZkmS4r2gnR4zbV98UF8TlTjflRuItnalBGiXnMoa0dydUvRMX9fV2cd21
2waFRny7EQ4L4
/FqalQRfKzC5C+ocOv3L+FOSL97dku7TetZ7Ur3glKo2ZnvuAbPyEvmAme6vmBQ1QJq0v
42aWHzE4u+9Z/RG69ZtoEs6h9IbrccQe+yK9kg1+yIyE7jPpk2wjL6
/SVacpfnzKef+FdwlZMZzGDbBalPLcncNr++XOPVuXbbzHuqjLmpdmuad6MtiPLWK2fSv
WSHSbuSedlgiFhWqgA
/FebVe9NopPn5Dd9xRPFBr1B5KL5q+SbnDSP0gQHe7r5+o2VHNaYUIaRHkWygeRyC72aU
5sT6mRuUR0W/4eBw0ZQvzfDpz1
/TuZ7T4zfLpSJfOz6GbMkrTitSSv4ws1kn55L1h4RyMbcUaB4p
/VVU+4vmOKYfAsggbKwBesP3AFACgAWFGP2PK+h3TvWUNyejBfjv7cwcKzsR8GvSD4p+R
/kbx98oM+rb+4+ft32N+vt6/voixAP/s+4+Y/z7L9yfiSPPBMeA
/JmNl+7YLAKBAOWAA5L/b/y8AAP//A6tAlY0KAAA="); err != nil {
    panic("add binary content to resource manager
failed: " + err.Error())
}
}
```

Load

- Loadpathprefix
-

```
func Load(path string, prefix ...string) error
```

-

```
package main

import "github.com/gogf/gf/v2/os/gres"

func main() {
    if err := gres.Load("../res/myfile"); err != nil {
        panic("load binary content to resource manager failed: " + err.Error())
    }
}
```

Get

- Get
-

```
func Get(path string) *File
```

-

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    file := gres.Get("../res/myfile")
    if file == nil {
        glog.Error(gctx.New(), "get file failed!")
        return
    }

    fmt.Println("Get File Name:", file.Name())
}
```

GetWithIndex

- GetWithIndexpath GetWithIndexhttp
-

```
func GetWithIndex(path string, indexFiles []string) *File
```

-

```

package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    file := gres.GetWithIndex("../res", []string{"myfile",
"myconfig"})
    if file == nil {
        glog.Error(gctx.New(), "get file failed!")
        return
    }

    fmt.Println("Get File Name:", file.Name())
}

```

GetContent

- GetContentpath
-

```

func GetContent(path string) []byte

```

-

```

package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    fileContent := gres.GetContent("../res/myfile")
    fmt.Println("Get File Content:", fileContent)
}

```

Contains

- Containspath
-

```

func Contains(path string) bool

```

-

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    if gres.Contains("../res/myfile") {
        fmt.Println("myfile is exist!")
    } else{
        fmt.Println("myfile is not exist!")
    }
}
```

IsEmpty

- IsEmpty

```
func IsEmpty() bool
```

-

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    fmt.Println(gres.IsEmpty())

    gres.Add("xxxxxxxxxxxxxxxxxxx")

    fmt.Println(gres.IsEmpty())

    // Output:
    // true
    // false
}
```

ScanDir

- ScanDirpathpatternrecursive>true

```
func ScanDir(path string, pattern string, recursive ...bool) []*File
```

-

```

package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    files := gres.ScanDir("../res", "*.doc,*.go", true)
    if len(files) > 0 {
        for _, file := range files {
            fmt.Println("ScanDir Result:", file.Name())
        }
    }
}

```

ScanDirFile

- ScanDirFilepathrecursive>true
-
-

```

func ScanDirFile(path string, pattern string, recursive ...bool) []
*File

```

-

```

package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    files := gres.ScanDirFile("../res", "*.*", true)
    if len(files) > 0 {
        for _, file := range files {
            fmt.Println("ScanDirFile Result:", file.
Name())
        }
    }
}

```

Export

- Exportsrcdst
-

```

func Export(src, dst string, option ...ExportOption) error

```

-

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    err := gres.Export("../res/src", "../res/dst")
    if err != nil {
        fmt.Println("gres.Export Error:", err)
    }
}
```

Dump

- Dump
-

```
func Dump()
```

-

```
package main

import (
    "github.com/gogf/gf/v2/os/gres"
)

func main() {
    gres.Add("xxxxxxxxx")

    gres.Dump()
}
```