

# -AdapterFile

## AdapterFile

AdapterFile

### g.Cfg

g.Cfg

config.yaml

```
server:
  address:      ":8888"
  openapiPath:  "/api.json"
  swaggerPath:  "/swagger"
  dumpRouterMap: false

database:
  default:
    link: "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
    debug: true
```

main.go

```
package main

import (
    "fmt"

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

func main() {
    var ctx = gctx.New()
    fmt.Println(g.Cfg().MustGet(ctx, "server.address").String())
    fmt.Println(g.Cfg().MustGet(ctx, "database.default").Map())
}
```

```
:8888
map[debug:true link:mysql:root:12345678@tcp(127.0.0.1:3306)/test]
```

## gcfg.NewWithAdapter

NewWithAdapterAdapterAdapterFile

config.yaml

### Content Menu

- [AdapterFile](#)
- [g.Cfg](#)
- [gcfg.NewWithAdapter](#)

```
server:
  address:      ":8888"
  openapiPath:  "/api.json"
  swaggerPath:  "/swagger"
  dumpRouterMap: false

database:
  default:
    link: "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
    debug: true
```

main.go

```
package main

import (
    "fmt"

    "github.com/gogf/gf/v2/os/gcfg"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var ctx = gctx.New()
    adapter, err := gcfg.NewAdapterFile("config")
    if err != nil {
        panic(err)
    }
    config := gcfg.NewWithAdapter(adapter)
    fmt.Println(config.MustGet(ctx, "server.address").String())
    fmt.Println(config.MustGet(ctx, "database.default").Map())
}
```

```
:8888
map[debug:true link:mysql:root:12345678@tcp(127.0.0.1:3306)/test]
```