

i18n.go: frame i18n.I18n()

i18n*i18n*

i18n

<https://github.com/gogf/gf/tree/master/util/gvalid/i18n/en>

i18n<https://github.com/gogf/gf/tree/master/util/gvalid/i18n/cn>

i18n__default__

i18n



g.i18n()

```
main.go
i18n
  en.toml
  zh-CN.toml
```

i18n

en.toml

```
"ReuiredUserName" = "Please input user name"
"ReuiredUserType" = "Please select user type"
```

zh-CN.toml

```
"ReuiredUserName" = ""
"ReuiredUserType" = ""
```

Content Menu

- - [i18n](#)
 -
 -
- -
 - [i18n](#)
 -

```

package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/i18n/gi18n"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/util/gconv"
)

func main() {
    type User struct {
        Name string `v:"required#ReuiredUserName"`
        Type int    `v:"required#ReuiredUserType"`
    }
    var (
        ctx = gctx.New()
        data = g.Map{
            "name": "john",
        }
        user = User{}
        ctxEn = gi18n.WithLanguage(ctx, "en")
        ctxCh = gi18n.WithLanguage(ctx, "zh-CN")
    )

    if err := gconv.Scan(data, &user); err != nil {
        panic(err)
    }
    //
    if err := g.Validator().Assoc(data).Data(user).Run(ctxEn); err !=
nil {
        g.Dump(err.String())
    }
    //
    if err := g.Validator().Assoc(data).Data(user).Run(ctxCh); err !=
nil {
        g.Dump(err.String())
    }
}

```

Please select user type