

Struct-

Struct

```
g.Validator().Data(object).Run(ctx)
```

tag

Structtag

```
[@][#]
```

-
- structerrorstruct
- |required|between:1,100
-

tag

```
package main

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

type User struct {
    Uid    int    `v:"uid        @integer|min:1#|ID"`
    Name   string `v:"name        @required|length:6,30#|"`
    Pass1  string `v:"password1@required|password3"`
    Pass2  string `v:"password2@required|password3|same:Pass1#|"`
}

func main() {
    var (
        ctx = gctx.New()
        user = &User{
            Name:   "john",
            Pass1:  "Abc123!@#",
            Pass2:  "123",
        }
    )

    err := g.Validator().Data(user).Run(ctx)
    if err != nil {
        g.Dump(err.Items())
    }
}
```

structgvalidgvalid tagssame:password1same:Pass1structstructstruct

Content Menu

- [tag](#)
- [tag](#)
- [map](#)
-

```
[
  {
    "uid": {
      "min": "ID",
    },
  },
  {
    "name": {
      "length": "",
    },
  },
  {
    "password2": {
      "password3": "",
    },
  },
]
```

map

```
package main

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

func main() {
    type User struct {
        Age  int
        Name string
    }
    var (
        ctx = gctx.New()
        user = User{Name: "john"}
        rules = map[string]string{
            "Name": "required|length:6,16",
            "Age": "between:18,30",
        }
        messages = map[string]interface{}{
            "Name": map[string]string{
                "required": "",
                "length": "{min}{max}",
            },
            "Age": "1830",
        }
    )

    err := g.Validator().Rules(rules).Messages(messages).Data(user).Run
(ctx)
    if err != nil {
        g.Dump(err.Maps())
    }
}
```

Age0requiredrequiredbetween

```
{
    "Age": {
        "between": "1830"
    },
    "Name": {
        "length": "616"
    }
}
```

embedded

```
package main

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

func main() {
    type Pass struct {
        Pass1 string `v:"password1@required|same:password2#|"`
        Pass2 string `v:"password2@required|same:password1#|"`
    }
    type User struct {
        Pass
        Id    int
        Name string `valid:"name@required#"`
    }
    var (
        ctx = gctx.New()
        user = &User{
            Name: "john",
            Pass: Pass{
                Pass1: "1",
                Pass2: "2",
            },
        }
    )
    err := g.Validator().Data(user).Run(ctx)
    g.Dump(err.Maps())
}
```

embedded

```

package main

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

func main() {
    type Pass struct {
        Pass1 string `v:"password1@required|same:password2#|"`
        Pass2 string `v:"password2@required|same:password1#|"`
    }
    type User struct {
        Id    int
        Name  string `valid:"name@required#"`
        Pass  Pass
    }
    var (
        ctx = gctx.New()
        user = &User{
            Name: "john",
            Pass: Pass{
                Pass1: "1",
                Pass2: "2",
            },
        }
    )
    err := g.Validator().Data(user).Run(ctx)
    g.Dump(err.Maps())
}

```

```

{
  "password1": {
    "same": "",
  },
  "password2": {
    "same": "",
  },
}

```