

required*nil

1

```
package main

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

func main() {
    type Params struct {
        Page      int    `v:"required|min:1"          # page is
required`
        Size      int    `v:"required|between:1,100" # size is
required`
        ProjectId string `v:"between:1,10000"        # project id
must between {min}, {max}`
    }
    var (
        ctx = gctx.New()
        obj = &Params{
            Page: 1,
            Size: 10,
        }
    )

    err := g.Validator().Data(obj).Run(ctx)
    fmt.Println(err)

    // Output:
    // <nil>
}
```

Content Menu

-
- [1](#)
- [2](#)
- [3](#)
- [4map](#)

2

```

package main

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

func main() {
    type Params struct {
        Page      int      `v:"required|min:1"          # page is
required"`
        Size       int      `v:"required|between:1,100" # size is
required"`
        ProjectId  *gvar.Var `v:"between:1,10000"        # project
id must between {min}, {max}"`
    }
    var (
        ctx = gctx.New()
        obj = &Params{
            Page: 1,
            Size: 10,
        }
    )
    err := g.Validator().Data(obj).Run(ctx)
    fmt.Println(err)

    // Output:
    // <nil>
}

```

3

Ofalse

```

package main

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

func main() {
    type Params struct {
        Page      int `v:"required|min:1"          # page is
        required`
        Size      int `v:"required|between:1,100 # size is
        required`
        ProjectId int `v:"between:1,10000"      # project id must
        between {min}, {max}`
    }
    var (
        ctx = gctx.New()
        obj = &Params{
            Page: 1,
            Size: 10,
        }
    )
    err := g.Validator().Data(obj).Run(ctx)
    fmt.Println(err)

    // Output:
    // <nil>
}

```

```

project id must between 1, 10000

```

4map

```

package main

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

func main() {
    var (
        ctx    = gctx.New()
        params = map[string]interface{}{
            "passport": "",
            "password": "123456",
            "password2": "1234567",
        }
        rules = []string{
            "passport@length:6,16",
            "password@required|length:6,16|same:password2",
            "password2@required|length:6,16",
        }
    )
    err := g.Validator().Rules(rules).Data(params).Run(ctx)
    if err != nil {
        g.Dump(err.Maps())
    }
}

```

passportrequiredpassport

```
{
  "password": {
    "same": "The password value `123456` must be the same as field
password2",
  },
}
```