



```
package main

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

func main() {
    var (
        ctx = gctx.New()
        rule = "length:6,16"
    )

    if err := g.Validator().Rules(rule).Data("123456").Run(ctx); err != nil {
        fmt.Println(err.String())
    }
    if err := g.Validator().Rules(rule).Data("12345").Run(ctx); err != nil {
        fmt.Println(err.String())
    }
}
```

The value `12345` length must be between 6 and 16

```
package main

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

func main() {
    var (
        ctx = gctx.New()
        rule = "integer|between:6,16"
        messages = "|"
        value = 5.66
    )

    if err := g.Validator().Rules(rule).Messages(messages).Data(value).Run(ctx); err != nil {
        g.Dump(err.Map())
    }
}
```

```
{
  "integer": "",
  "between": "",
}
```

| messagesstringmap[string]string

```
package main

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

func main() {
    var (
        ctx      = gctx.New()
        rule     = "url|min-length:11"
        value    = "goframe.org"
        messages = map[string]string{
            "url":      "URL",
            "min-length": "{min}",
        }
    )
    if err := g.Validator().Rules(rule).Messages(messages).Data(value).
        Run(ctx); err != nil {
        g.Dump(err.Map())
    }
}
```

```
{
  "url": "URL",
}
```

```
package main

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

func main() {
    var (
        ctx = gctx.New()
        rule = `regex:\d{6,}|\D{6,}|max-length:16`
    )

    if err := g.Validator().Rules(rule).Data(`123456`).Run(ctx); err != nil {
        fmt.Println(err)
    }

    if err := g.Validator().Rules(rule).Data(`abcde6`).Run(ctx); err != nil {
        fmt.Println(err)
    }
}
```

The value `abcde6` must be in regex of: `\d{6,}|\D{6,}`