

# Map-

```
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 = map[string]string{
            "passport": "required|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())
    }
}
```

```
{
    "passport": {
        "required": "The passport field is required",
        "length": "The passport value `` length must be between 6 and
16",
    },
    "password": {
        "same": "The password value `123456` must be the same as field
password2",
    },
}
```

## Content Menu

•  
•

```

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 = map[string]string{
            "passport": "required|length:6,16",
            "password": "required|length:6,16|same:password2",
            "password2": "required|length:6,16",
        }
        messages = map[string]interface{}{
            "passport": "{min}{max}",
            "password": map[string]string{
                "required": "",
                "same": "",
            },
        }
    )

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

```

messagesstringmap[string]stringmap[string]string

```

{
    "passport": {
        "length": "616",
        "required": ""
    },
    "password": {
        "same": ""
    }
}

```