



链滴

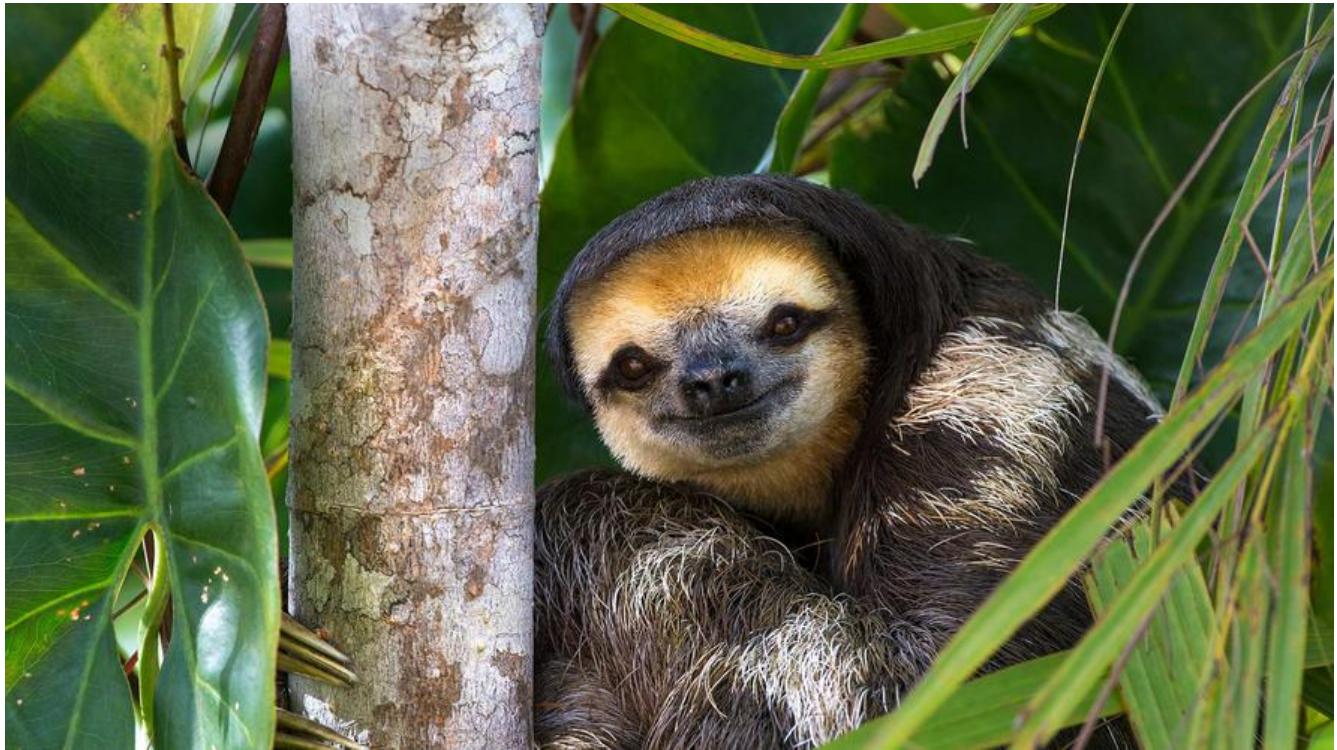
在复杂多变的业务中使用规则表达式

作者: [wangdy](#)

原文链接: <https://ld246.com/article/1582869620750>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



不废话，直接上 demo，实际应用中需要考虑的因素比较多，结合具体情况吧。

这边由于基础平台没有广告系统，所以自己使用 expr 做个通用的规则系统。

```
package main

import (
    "fmt"
    "time"

    "github.com/antonmedv/expr"
    "github.com/antonmedv/expr/vm"
)

func main() {
    var programs []*vm.Program
    for _, ruleExpr := range ruleExprList {
        p1, err := expr.Compile(ruleExpr, expr.Env(ruleFunc))
        if err != nil {
            fmt.Println(err)
            continue
        }
        programs = append(programs, p1)
    }

    for _, program := range programs {
        ruleFunc.RequestContext = RequestContext{Time: time.Now()}
        output, err := vm.Run(program, ruleFunc)
        fmt.Println("run:", output, err)
    }
}
```

```
var ruleExprList = []string{
    "NowStr(RequestContext)",
    "Config.Id",
}

var ruleFunc = &RuleContext{
    NowStr: Now,
    Config:Config,
}

type RuleContext struct {
    RequestContext
    NowStr func(RequestContext) interface{}
    Config map[string]int64
}

type RequestContext struct {
    time.Time
}

func Now(req RequestContext) interface{} {
    return req.Time
}

var Config = map[string]int64{
    "Id":111,
}
```