



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 14. gin集成nacos

- 全部开发者教程
8. 最大努力通知

9. mq (message queue) 的使用场景

10. mq技术选型

11. rocketmq安装和配置

12. rocketmq的基本概念

13. rocketmq的消息类型

14. go操作Rocketmq开发环境搭建

15. 什么是服务雪崩

16. grpc的超时和重试

17. 常见的幂等性解决方案

第19周 链路追踪、限流、熔断、降级

1. 什么是链路追踪



bobby · 更新于 2022-11-08

← 上一节 13. nacos的基... 1. model定义 下一节 →

nacos-sdk-go地址

1. go操作nacos

<> 代码块

```
1 package main
2
3 import (
4     "fmt"
5     "time"
6
7     "github.com/nacos-group/nacos-sdk-go/clients"
8     "github.com/nacos-group/nacos-sdk-go/common/constant"
9     "github.com/nacos-group/nacos-sdk-go/vo"
10 )
11 func main(){
12     sc := []constant.ServerConfig{
13         {
14             IpAddr: "192.168.1.103",
15             Port: 8848,
16         },
17     }
18
19     cc := constant.ClientConfig {
20         NamespaceId: "c1872978-d51c-4188-a497-4e0cd20b97d5", // 如果需要支
21         TimeoutMs: 5000,
22         NotLoadCacheAtStart: true,
23         LogDir: "tmp/nacos/log",
24         CacheDir: "tmp/nacos/cache",
25         RotateTime: "1h",
26         MaxAge: 3,
27         LogLevel: "debug",
28     }
29
30     configClient, err := clients.CreateConfigClient(map[string]interface{}{
31         "serverConfigs": sc,
32         "clientConfig": cc,
33     })
34     if err != nil {
35         panic(err)
36     }
37
38     content, err := configClient.GetConfig(vo.ConfigParam{
39         DataId: "user-web.yaml",
40         Group: "dev"})
41 }
```

意见反馈

收藏教程

标记书签



```
44     }
45     fmt.Println(content)
46
47     err = configClient.ListenConfig(vo.ConfigParam{
48         DataId: "user-web.yaml",
49         Group:  "dev",
50         OnChange: func(namespace, group, dataId, data string) {
51             fmt.Println("配置文件变化")
52             fmt.Println("group:" + group + ", dataId:" + dataId + ", data:" + data)
53         },
54     })
55     time.Sleep(3000 * time.Second)
56
57 }
```

2. 如何将nacos中的配置映射成go的struct

转换地址：<https://www.json2yaml.com/convert-yaml-to-json>

<> 代码块

```
1 //go语言本身支持将json的字符串反射成struct
```

13. nacos的基本使用 ◀ 上一节

下一节 ▶ 1. model定义

✎ 我要提出意见反馈

