



《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

慕课网首页

免费课

实战课

体系课

慕课教程

专栏

手记

企业服务



我的

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 > 4.token认证

全部开发者教程



7. 图形验证码

8. 阿里云发送短信

9. redis的安装

第10周 服务注册/发现、配置中心、负载均衡

1. 什么是服务注册和发现

2. consul的安装和配置

3. consul的api接口

4. go操作consul

5. grpc下的健康检查

6. 动态获取可用端口号

7. 负载均衡策略

8. 常见的负载均衡算法

9. grpc的负载均衡策略

10. 为什么需要分布式配置中心



bobby · 更新于 2022-11-08

◀ 上一节 3. grpc拦截器 -... 5. grpc验证器 下一节 ▶

1. proto

<> 代码块

预览 复制

```
1 syntax = "proto3";
2 option go_package = ".;proto";
3 service Greeter {
4     rpc SayHello (HelloRequest) returns (HelloReply);
5 }
6
7 message HelloRequest {
8     string name = 1;
9 }
10
11 message HelloReply {
12     string message = 1;
13 }
```

2. 服务端

<> 代码块

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6     "google.golang.org/grpc/codes"
7     "google.golang.org/grpc/metadata"
8     "google.golang.org/grpc/status"
9     "net"
10
11     "google.golang.org/grpc"
12
13     "start/token_interceptor/proto"
14 )
15
16
17 type Server struct{}
18
19 func (s *Server) SayHello(ctx context.Context, request *proto.HelloRequest) (*proto.HelloReply, error){
20     return &proto.HelloReply{
21         Message: "hello "+request.Name,
22     }, nil
23 }
24 }
```

意见反馈

收藏教程

标记书签

```
26
27 func main(){
28     lis, err := net.Listen("tcp", ":50051")
29     if err != nil {
30         fmt.Println(err.Error())
31         return
32     }
33     var interceptor grpc.UnaryServerInterceptor
34     interceptor = func(ctx context.Context, req interface{}, info *grpc.UnaryServerInfo) (resp interface{}, err error) {
35         md, ok := metadata.FromIncomingContext(ctx)
36         if !ok {
37             return resp, status.Errorf(codes.Unauthenticated, "无Token认证信息")
38         }
39
40         var (
41             appid string
42             appkey string
43         )
44
45         if val, ok := md["appid"]; ok {
46             appid = val[0]
47         }
48
49         if val, ok := md["appkey"]; ok {
50             appkey = val[0]
51         }
52
53         if appid != "imooc" || appkey != "bobby" {
54             return resp, status.Errorf(codes.Unauthenticated, "Token认证信息无效:appid=%s,appkey=%s", appid, appkey)
55         }
56
57         // 继续处理请求
58         return handler(ctx, req)
59     }
60     var opts []grpc.ServerOption
61     opts = append(opts, grpc.UnaryInterceptor(interceptor))
62
63     s := grpc.NewServer(opts...)
64     ser := & Server{}
65     proto.RegisterGreeterServer(s, ser)
66     s.Serve(lis)
67 }
68
```

3. 客户端

<> 代码块

```
1 package main
2
3 import (
4     "context"
5     "fmt"
6     "google.golang.org/grpc"
7     "start/token_interceptor/proto"
8 )
9
10 type customCredential struct{}
11
12 func (c customCredential) GetRequestMetadata(ctx context.Context, uri ...string) (map[string]string, error) {
13     return map[string]string{
```

意见反馈

收藏教程

标记书签

```
15         "appkey": "i am key",
16     }, nil
17 }
18
19 func (c customCredential) RequireTransportSecurity() bool {
20     return false
21 }
22
23
24 func main() {
25     var opts []grpc.DialOption
26
27     //opts = append(opts, grpc.WithUnaryInterceptor(interceptor))
28     opts = append(opts, grpc.WithInsecure())
29     opts = append(opts, grpc.WithPerRPCCredentials(new(customCredential)))
30     // 指定客户端interceptor
31
32     conn, err := grpc.Dial("localhost:50051", opts...)
33     if err != nil {
34         panic(err)
35     }
36     defer conn.Close()
37
38     c := proto.NewGreeterClient(conn)
39     //rsp, _ := c.Search(context.Background(), &empty.Empty{})
40     rsp, err := c.SayHello(context.Background(), &proto.HelloRequest{
41         Name: "bobby",
42     })
43     if err != nil {
44         panic(err)
45     }
46     fmt.Println(rsp.Message)
47 }
48
49
```

[3. grpc拦截器](#) – go ◀ 上一节下一节 ▶ [5. grpc验证器](#)[✎ 我要提出意见反馈](#)[企业服务](#) [网站地图](#) [网站首页](#) [关于我们](#) [联系我们](#) [讲师招募](#) [帮助中心](#) [意见反馈](#) [代码托管](#)

Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号

[✎ 意见反馈](#)[♡ 收藏教程](#)[🔖 标记书签](#)