优质IT资源微信x923713



《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!



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 我拍 Q 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 > 5. grpc验证器 쿻 全部开发者教程 \equiv bobby・更新干 2022-11-08 ◆ 上一节 4. token认证6. grpc中的异常... 下一节 → 9. redis的安装 第10周 服务注册/发现、配 protoc-gen_validate 置中心、负载均衡 1. 什么是服务注册和发现 1. 安装和配置 2 consul的安装和配置 3. consul的api接口 linux 4. go操作consul <> 代码块 预览 复制 5. grpc下的健康检查 # fetches this repo into \$GOPATH 1 6. 动态获取可用端口号 go get -d github.com/envoyproxy/protoc-gen-validate 7. 负载均衡策略 4 # installs PGV into \$GOPATH/bin make build 8.常见的负载均衡算法 9. grpc的负载均衡策略 windows 10. 为什么需要分布式配置中心 ② exe 下载地址 11. 分布式配置中心选型 exe下载地址 \Box 可以点此下载: protoc-gen-validate.zip 12. nacos的安装和配置 0 13. nacos的基本使用 将 zip文件中的exe文件拷贝到 go的根补录的bin目录下 14. gin集成nacos 第11周 商品微服务的grpc服 生成python源码 <> 代码块 protoc -I . --go_out=plugins=grpc:. --validate_out="lang=go:." helloworld.proto 2. proto 1. 新建validate.proto文件内容从 https://github.com/envoyproxy/protoc-genvalidate/blob/master/validate/validate.proto 拷贝 2. 新建helloworl.proto文件 <> 代码块

1

syntax = "proto3";

♡ 收藏教程

□ 标记书签

▶ 意见反馈

不

优质IT资源微信x923713

```
service Greeter {
 6
         rpc SayHello (Person) returns (Person);
 7
     }
 9
     message Person {
10
11
         uint64 id
                      = 1 [(validate.rules).uint64.gt
                                                         = 999];
12
         string email = 2 [(validate.rules).string.email = true];
13
         string name = 3 [(validate.rules).string = {
14
                           pattern: "^{[0-9]A-Za-z]+([^{[0-9]A-Za-z]+)*}, max_bytes
15
16
     }
17
```

3. 服务端

```
<> 代码块
 1
     package main
     import (
 3
         "context"
 4
         "google.golang.org/grpc/codes"
         "google.golang.org/grpc/status"
 6
         "net"
 7
 8
         "google.golang.org/grpc"
10
         "start/pgv_test/proto"
11
12
13
14
15
     type Server struct{}
16
     func (s *Server) SayHello(ctx context.Context, request *proto.Person) (*proto.Per
17
         error){
18
19
         return &proto.Person{
             Id: 32,
20
         }, nil
21
     }
22
23
24
     type Validator interface {
         Validate() error
25
26
27
     func main(){
28
         var interceptor grpc.UnaryServerInterceptor
29
         interceptor = func(ctx context.Context, req interface{}, info *grpc.UnaryServ
30
             // 继续处理请求
31
             if r, ok := req.(Validator); ok {
32
33
                 if err := r.Validate(); err != nil {
                     return nil, status.Error(codes.InvalidArgument, err.Error())
34
35
             }
36
37
38
             return handler(ctx, req)
         }
39
40
         var opts []grpc.ServerOption
         opts = append(opts, grpc.UnaryInterceptor(interceptor))
41
42
         a := arpc.NewServer(opts...)
                                                                                        不
43
        ▶ 意见反馈
                         ♡ 收藏教程
                                          □ 标记书签
```

⊡

②

 \Box

0

优质IT资源微信x923713

```
lis, err := net.Listen("tcp", "0.0.0.0:50051")
         if err != nil{
46
             panic("failed to listen:"+err.Error())
47
48
         }
         err = g.Serve(lis)
49
         if err != nil{
50
             panic("failed to start grpc:"+err.Error())
51
52
53
     }
54
```

4. 客户端

```
<> 代码块
 1
     package main
2
     import (
3
         "context"
 4
         "fmt"
 5
         "google.golang.org/grpc"
 6
         "start/pgv_test/proto"
 7
 8
 9
     type customCredential struct{}
10
11
12
     func main() {
13
14
         var opts []grpc.DialOption
15
         //opts = append(opts, grpc.WithUnaryInterceptor(interceptor))
16
         opts = append(opts, grpc.WithInsecure())
17
18
19
         conn, err := grpc.Dial("localhost:50051", opts...)
         if err != nil {
20
             panic(err)
21
22
         }
         defer conn.Close()
23
24
         c := proto.NewGreeterClient(conn)
25
         //rsp, _ := c.Search(context.Background(), &empty.Empty{})
27
         rsp, err := c.SayHello(context.Background(), &proto.Person{
             Email: "bobby",
28
29
         })
         if err != nil {
30
             panic(err)
31
32
33
         fmt.Println(rsp.Id)
     }
34
35
```

▶ 我要提出意见反馈

⊡

?

 \Box

0

尤质IT资源微信x923713

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 代码打

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

*

 $\overline{\uparrow}$

∷

?

0