全部开发者教程

1. go最常用的设计模式 – 函数选项
2. 单例模式和懒加载
3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码
2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动

bobby · 更新于 2022-11-16

上一节 4. 用户服务 – p... 6. userop-web... 下一节

1. message

代码块

```
package handler

import (
    "context"
    "mxshop_srvs/userop_srv/proto"
    "mxshop_srvs/userop_srv/model"
    "mxshop_srvs/userop_srv/global"
)

func (*UserOpServer) MessageList(ctx context.Context, req *proto.MessageRequest)
    var rsp proto.MessageListResponse
    var messages []model.LeavingMessages
    var messageList  []*proto.MessageResponse

    result := global.DB.Where(&model.LeavingMessages{User:req.UserId}).Find(&mess
    rsp.Total = int32(result.RowsAffected)

    for _, message := range messages {
        messageList = append(messageList, &proto.MessageResponse{
            Id:          message.ID,
            UserId:      message.User,
            MessageType: message.MessageType,
            Subject:     message.Subject,
            Message:     message.Message,
            File:        message.File,
        })
    }

    rsp.Data = messageList
    return &rsp, nil
}


func (*UserOpServer) CreateMessage(ctx context.Context, req *proto.MessageReques
    var message model.LeavingMessages

    message.User = req.UserId
    message.MessageType = req.MessageType
    message.Subject = req.Subject
    message.Message = req.Message
    message.File = req.File

    global.DB.Save(&message)
```

```
46      return &proto.MessageResponse{Id:message.ID}, nil
    }
```

2. address

代码块

```
1   package handler
2
3   import (
4       "context"
5       "google.golang.org/grpc/codes"
6       "google.golang.org/grpc/status"
7       "google.golang.org/protobuf/types/known/emptypb"
8       "mxshop_srvs/userop_srv/model"
9       "mxshop_srvs/userop_srv/proto"
10      "mxshop_srvs/userop_srv/global"
11  )
12
13  func (*UserOpServer) GetAddressList(ctx context.Context, req *proto.AddressReques
14      var addresses []model.Address
15      var rsp proto.AddressListResponse
16      var addressResponse []*proto.AddressResponse
17
18      if result := global.DB.Where(&model.Address{User:req.UserId}).Find(&addresses
19          rsp.Total = int32(result.RowsAffected)
20      }
21
22      for _, address := range addresses {
23          addressResponse = append(addressResponse, &proto.AddressResponse{
24              Id: address.ID,
25              UserId: address.User,
26              Province: address.Province,
27              City: address.City,
28              District: address.District,
29              Address:  address.Address,
30              SignerName: address.SignerName,
31              SignerMobile:address.SignerMobile,
32          })
33      }
34      rsp.Data = addressResponse
35
36      return &rsp, nil
37  }
38
39  func (*UserOpServer) CreateAddress(ctx context.Context, req *proto.AddressRequest
40      var address model.Address
41
42      address.User = req.UserId
43      address.Province = req.Province
44      address.City = req.City
45      address.District = req.District
46      address.Address = req.Address
47      address.SignerName = req.SignerName
48      address.SignerMobile = req.SignerMobile
49
50      global.DB.Save(&address)
51
52      return &proto.AddressResponse{Id:address.ID}, nil
53  }
54
```

意见反馈 收藏教程 标记书签

```
56      if result := global.DB.Where("id=? and user=?", req.Id, req.UserId).Delete(&
57          return nil, status.Errorf(codes.NotFound, "收货地址不存在")
58      }
59      return &emptypb.Empty{}, nil
60  }
61
62  func (*UserOpServer) UpdateAddress(ctx context.Context, req *proto.AddressRequest
63      var address model.Address
64
65      if result := global.DB.Where("id=? and user=?", req.Id, req.UserId).First(&ad
66          return nil, status.Errorf(codes.NotFound, "购物车记录不存在")
67      }
68
69      if address.Province != "" {
70          address.Province = req.Province
71      }
72
73      if address.City != "" {
74          address.City = req.City
75      }
76
77      if address.District != "" {
78          address.District = req.District
79      }
80
81      if address.Address != "" {
82          address.Address = req.Address
83      }
84
85      if address.SignerName != "" {
86          address.SignerName = req.SignerName
87      }
88
89      if address.SignerMobile != "" {
90          address.SignerMobile = req.SignerMobile
91      }
92
93      global.DB.Save(&address)
94
95      return &emptypb.Empty{}, nil
96  }
97
98
```

3. userfav

代码块

```
1   package handler
2
3   import (
4       "context"
5       "google.golang.org/protobuf/types/known/emptypb"
6       "mxshop_srvs/userop_srv/model"
7
8       "google.golang.org/grpc/codes"
9       "google.golang.org/grpc/status"
10      "mxshop_srvs/userop_srv/proto"
11      "mxshop_srvs/userop_srv/global"
12  )
13
```

意见反馈 收藏教程 标记书签

```
15      var rsp proto.UserFavListResponse
16      var userFavs []model.UserFav
17      var userFavList  []*proto.UserFavResponse
18
19      result := global.DB.Where(&model.UserFav{User:req.UserId, Goods:req.GoodsId}
20      rsp.Total = int32(result.RowsAffected)
21
22      for _, userFav := range userFavs {
23          userFavList = append(userFavList, &proto.UserFavResponse{
24              UserId: userFav.User,
25              GoodsId: userFav.Goods,
26          })
27      }
28
29      rsp.Data = userFavList
30
31      return &rsp, nil
32  }
33
34  func (*UserOpServer) AddUserFav(ctx context.Context, req *proto.UserFavRequest) (
35      var userFav model.UserFav
36
37      userFav.User = req.UserId
38      userFav.Goods = req.GoodsId
39
40      global.DB.Save(&userFav)
41
42      return &emptypb.Empty{}, nil
43  }
44
45  func (*UserOpServer) DeleteUserFav(ctx context.Context, req *proto.UserFavRequest
46      if result := global.DB.Unscoped().Where("goods=? and user=?", req.GoodsId, re
47          return nil, status.Errorf(codes.NotFound, "收藏记录不存在")
48      }
49      return &emptypb.Empty{}, nil
50  }
51
52  func (*UserOpServer) GetUserFavDetail(ctx context.Context, req *proto.UserFavRequ
53      var userfav model.UserFav
54      if result := global.DB.Where("goods=? and user=?", req.GoodsId, req.UserId).F
55          return nil, status.Errorf(codes.NotFound, "收藏记录不存在")
56      }
57      return &emptypb.Empty{}, nil
58  }
59
60
61
62
```

4. 用户服务 – proto定义 ◂ 上一节 下一节 ▸ 6. userop-web代码

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