



《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

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 > 7. 品牌相关的接口

全部开发者教程



5. grpc下添加jaeger

6. 熔断、限流技术选型

第20周 api网关、部署

1. api网关对比

2. kong的安装和配置

3. 配置jwt

4. 什么是敏捷开发

5. jenkins的安装和配置

6. jenkins的插件管理

7. 部署到远程服务器并运行

8. jenkins的pipeline参数详解

9. 定时构建的语法

10. shell脚本启动go服务

第21周 开发规范和go基础扩展



bobby · 更新于 2022-11-08

◀ 上一节 6. 轮播图的各个... 8. 品牌分类接口 下一节 ▶

1. BrandList

<> 代码块

```
1 func (s *GoodsServer) BrandList(ctx context.Context, req *proto.BrandFilterRequest) (*proto.BrandListResponse, error) {
2     brandListResponse := proto.BrandListResponse{}
3
4     var brands []model.Brands
5     result := global.DB.Scopes(Paginate(int(req.Pages), int(req.PagePerNums))).Find(&brands)
6     if result.Error != nil {
7         return nil, result.Error
8     }
9
10    var total int64
11    global.DB.Model(&model.Brands{}).Count(&total)
12    brandListResponse.Total = int32(total)
13
14    var brandResponses []*proto.BrandInfoResponse
15    for _, brand := range brands {
16        brandResponses = append(brandResponses, &proto.BrandInfoResponse{
17            Id: brand.ID,
18            Name: brand.Name,
19            Logo: brand.Logo,
20        })
21    }
22    brandListResponse.Data = brandResponses
23    return &brandListResponse, nil
24 }
```

2. CreateBrand

<> 代码块

```
1 func (s *GoodsServer) CreateBrand(ctx context.Context, req *proto.BrandRequest) (*proto.BrandResponse, error) {
2     //新建品牌
3     if result := global.DB.First(&model.Brands{}); result.RowsAffected == 1 {
4         return nil, status.Errorf(codes.InvalidArgument, "品牌已存在")
5     }
6
7     brand := &model.Brands{
8         Name: req.Name,
9         Logo: req.Logo,
10    }
11    global.DB.Save(brand)
12 }
```

意见反馈

收藏教程

标记书签

```
14     return &proto.BrandInfoResponse{Id: brand.ID}, nil
    }
```

3. DeleteBrand

<> 代码块

```
1 func (s *GoodsServer) DeleteBrand(ctx context.Context, req *proto.BrandRequest) {
2     if result := global.DB.Delete(&model.Brands{}, req.Id); result.RowsAffected == 0 {
3         return nil, status.Errorf(codes.NotFound, "品牌不存在")
4     }
5     return &emptypb.Empty{}, nil
6 }
```

4. UpdateBrand

<> 代码块

```
1 func (s *GoodsServer) UpdateBrand(ctx context.Context, req *proto.BrandRequest) {
2     brands := model.Brands{}
3     if result := global.DB.First(&brands); result.RowsAffected == 0 {
4         return nil, status.Errorf(codes.InvalidArgument, "品牌不存在")
5     }
6
7     if req.Name != "" {
8         brands.Name = req.Name
9     }
10    if req.Logo != "" {
11        brands.Logo = req.Logo
12    }
13
14    global.DB.Save(&brands)
15
16    return &emptypb.Empty{}, nil
17 }
```

6. 轮播图的各个handler ◀ 上一节 下一节 ▶ 8. 品牌分类接口

✎ 我要提出意见反馈