bobby · 更新于 2022-11-08

上一节 7. 品牌相关的接口 1. 商品列表接口 下一节

1. CategoryBrandList

代码块 预览 复制

```
func (s *GoodsServer) CategoryBrandList(ctx context.Context, req *proto.CategoryB
    var categoryBrands []model.GoodsCategoryBrand
    categoryBrandListResponse := proto.CategoryBrandListResponse{}

    var total int64
    global.DB.Model(&model.GoodsCategoryBrand{}).Count(&total)
    categoryBrandListResponse.Total = int32(total)

    global.DB.Scopes(Paginate(int(req.Pages), int(req.PagePerNums))).Find(&categ

    var categoryResponses []*proto.CategoryBrandResponse
    for _, categoryBrand := range categoryBrands {
        categoryResponses = append(categoryResponses, &proto.CategoryBrandRespons
            Category: &proto.CategoryInfoResponse{
                Id: categoryBrand.Category.ID,
                Name: categoryBrand.Category.Name,
                Level: categoryBrand.Category.Level,
                IsTab: categoryBrand.Category.IsTab,
                ParentCategory: categoryBrand.Category.ParentCategoryID,
            },
            Brand: &proto.BrandInfoResponse{
                Id:   categoryBrand.Brands.ID,
                Name: categoryBrand.Brands.Name,
                Logo: categoryBrand.Brands.Logo,
            },
        })
    }

    categoryBrandListResponse.Data = categoryResponses
    return &categoryBrandListResponse, nil
}
```

2. GetCategoryBrandList

代码块

```
func (s *GoodsServer) GetCategoryBrandList(ctx context.Context, req *proto.Categ
    brandListResponse := proto.BrandListResponse{}

    var category model.Category
```

意见反馈 收藏教程 标记书签

```
    }

    var categoryBrands []model.GoodsCategoryBrand
    if result := global.DB.Where(&model.GoodsCategoryBrand{CategoryID: category.I
        brandListResponse.Total = int32(result.RowsAffected)
    }

    var brandInfoResponses []*proto.BrandInfoResponse
    for _, categoryBrand := range categoryBrands {
        brandInfoResponses = append(brandInfoResponses, &proto.BrandInfoResponse{
            Id: int32(categoryBrand.Brand.ID),
            Name: categoryBrand.Brand.Name,
            Logo: categoryBrand.Brand.Logo,
        })
    }

    brandListResponse.Data = brandInfoResponses

    return &brandListResponse, nil
}
```

3. CreateCategoryBrand

代码块

```
func (s *GoodsServer) CreateCategoryBrand(ctx context.Context, req *proto.Categor
    var category model.Category
    if result := global.DB.First(&category, req.CategoryId); result.RowsAffected
        return nil, status.Errorf(codes.InvalidArgument, "商品分类不存在")
    }

    var brand model.Brands
    if result := global.DB.First(&brand, req.BrandId); result.RowsAffected == 0 {
        return nil, status.Errorf(codes.InvalidArgument, "品牌不存在")
    }

    categoryBrand := model.GoodsCategoryBrand{
        CategoryID: req.CategoryId,
        BrandID: req.BrandId,
    }

    global.DB.Save(&categoryBrand)
    return &proto.CategoryBrandResponse{Id: int32(categoryBrand.ID)}, nil
}
```

4. DeleteCategoryBrand

代码块

```
func (s *GoodsServer) DeleteCategoryBrand(ctx context.Context, req *proto.Categor
    if result := global.DB.Delete(&model.GoodsCategoryBrand{}, req.Id); result.Ro
        return nil, status.Errorf(codes.NotFound, "品牌分类不存在")
    }
    return &emptypb.Empty{}, nil
}
```

意见反馈 收藏教程 标记书签

5. UpdateCategoryBrand

代码块

```
1  func (s *GoodsServer) UpdateCategoryBrand(ctx context.Context, req *proto.Categor
2      var categoryBrand model.GoodsCategoryBrand
3
4      if result := global.DB.First(&categoryBrand, req.Id); result.RowsAffected ==
5          return nil, status.Errorf(codes.InvalidArgument, "品牌分类不存在")
6      }
7
8      var category model.Category
9      if result := global.DB.First(&category, req.CategoryId); result.RowsAffected
10         return nil, status.Errorf(codes.InvalidArgument, "商品分类不存在")
11     }
12
13     var brand model.Brands
14     if result := global.DB.First(&brand, req.BrandId); result.RowsAffected == 0 {
15         return nil, status.Errorf(codes.InvalidArgument, "品牌不存在")
16     }
17
18     categoryBrand.CategoryID = req.CategoryId
19     categoryBrand.BrandID = req.BrandId
20
21     global.DB.Save(&categoryBrand)
22
23     return &emptypb.Empty{}, nil
24 }
```

7. 品牌相关的接口 上一节 下一节 1. 商品列表接口

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

意见反馈 收藏教程 标记书签