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！ 点此查

bobby · 更新于 2022-11-16

上一节 5. 轮播图接口 7. oss快速入门 下一节

1. form

<> 代码块 预览 复制

```go
package forms

type BrandForm struct {
    Name string `form:"name" json:"name" binding:"required,min=3,max=10"`
    Logo string `form:"logo" json:"logo" binding:"url"`
}


type CategoryBrandForm struct {
    CategoryId int    `form:"category_id" json:"category_id" binding:"required"`
    BrandId int    `form:"brand_id" json:"brand_id" binding:"required"`
}

```

2. handler

<> 代码块

```go
package brands

import (
    "context"
    "github.com/gin-gonic/gin"
    "github.com/go-playground/validator/v10"
    "github.com/golang/protobuf/ptypes/empty"
    "google.golang.org/grpc/codes"
    "google.golang.org/grpc/status"
    "mxshop-api/goods-web/forms"
    "mxshop-api/goods-web/global"
    "mxshop-api/goods-web/proto"
    "net/http"
    "strconv"
    "strings"
)

func removeTopStruct(fields map[string]string) map[string]string{
    rsp := map[string]string{}
    for field, err := range fields {
        rsp[field[strings.Index(field, ".")+1:]] = err
    }
```

意见反馈 收藏教程 标记书签

优质IT资源微信x923713

```go


func HandleGrpcErrorToHttp(err error, c *gin.Context) {
    //将grpc的code转换成http的状态码
    if err != nil {
        if e, ok := status.FromError(err); ok {
            switch e.Code() {
            case codes.NotFound:
                c.JSON(http.StatusNotFound, gin.H{
                    "msg": e.Message(),
                })
            case codes.Internal:
                c.JSON(http.StatusInternalServerError, gin.H{
                    "msg:": "内部错误",
                })
            case codes.InvalidArgument:
                c.JSON(http.StatusBadRequest, gin.H{
                    "msg": "参数错误",
                })
            case codes.Unavailable:
                c.JSON(http.StatusInternalServerError, gin.H{
                    "msg": "用户服务不可用",
                })
            default:
                c.JSON(http.StatusInternalServerError, gin.H{
                    "msg": e.Code(),
                })
            }
            return
        }
    }
}

func HandleValidatorError(c *gin.Context, err error){
    errs, ok := err.(validator.ValidationErrors)
    if !ok {
        c.JSON(http.StatusOK, gin.H{
            "msg":err.Error(),
        })
    }
    c.JSON(http.StatusBadRequest, gin.H{
        "error": removeTopStruct(errs.Translate(global.Trans)),
    })
    return
}

func BrandList(ctx *gin.Context) {
    pn := ctx.DefaultQuery("pn", "0")
    pnInt, _ := strconv.Atoi(pn)
    pSize := ctx.DefaultQuery("psize", "10")
    pSizeInt, _ := strconv.Atoi(pSize)

    rsp, err := global.GoodsSrvClient.BrandList(context.Background(), &proto.Brar
        Pages: int32(pnInt),
        PagePerNums: int32(pSizeInt),
    })

    if err != nil {
        api.HandleGrpcErrorToHttp(err, ctx)
        return
    }

    result := make([]interface{}, 0)
    rsMap := make(map[string]interface{})
```

意见反馈 收藏教程 标记书签

优质IT资源微信x923713

```
    for _, value := range rsp.Data[pnInt:pnInt*pSizeInt+pSizeInt] {
        reMap := make(map[string]interface{})
        reMap["id"] = value.Id
        reMap["name"] = value.Name
        reMap["logo"] = value.Logo

        result = append(result, reMap)
    }

    reMap["data"] = result

    ctx.JSON(http.StatusOK, reMap)
}

func NewBrand(ctx *gin.Context) {
    brandForm := forms.BrandForm{}
    if err := ctx.ShouldBindJSON(&brandForm); err != nil {
        HandleValidatorError(ctx, err)
        return
    }

    rsp, err := global.GoodsSrvClient.CreateBrand(context.Background(), &proto.Br
        Name: brandForm.Name,
        Logo: brandForm.Logo,
    })

    if err != nil {
        HandleGrpcErrorToHttp(err, ctx)
        return
    }

    request := make(map[string]interface{})
    request["id"] = rsp.Id
    request["name"] = rsp.Name
    request["logo"] = rsp.Logo

    ctx.JSON(http.StatusOK, request)
}

func DeleteBrand(ctx *gin.Context) {
    id := ctx.Param("id")
    i, err := strconv.ParseInt(id, 10, 32)
    if err != nil {
        ctx.Status(http.StatusNotFound)
        return
    }
    _, err = global.GoodsSrvClient.DeleteBrand(context.Background(), &proto.Branc
    if err != nil {
        HandleGrpcErrorToHttp(err, ctx)
        return
    }

    ctx.Status(http.StatusOK)
}

func UpdateBrand(ctx *gin.Context) {
    brandForm := forms.BrandForm{}
    if err := ctx.ShouldBindJSON(&brandForm); err != nil {
        HandleValidatorError(ctx, err)
        return
    }

    id := ctx.Param("id")
    i, err := strconv.ParseInt(id, 10, 32)
```

意见反馈 收藏教程 标记书签

优质IT资源微信x923713

```go
            ctx.Status(http.StatusNotFound)
            return
        }

        _, err = global.GoodsSrvClient.UpdateBrand(context.Background(), &proto.Branc
            Id:   int32(i),
            Name: brandForm.Name,
            Logo: brandForm.Logo,
        })
        if err != nil {
            HandleGrpcErrorToHttp(err, ctx)
            return
        }
        ctx.Status(http.StatusOK)
    }


    func GetCategoryBrand(ctx *gin.Context) {
        id := ctx.Param("id")
        i, err := strconv.ParseInt(id, 10, 32)
        if err != nil {
            ctx.Status(http.StatusNotFound)
            return
        }
        rsp, err := global.GoodsSrvClient.GetCategoryBrandList(context.Background(),
            Id: int32(i),
        })

        if err != nil {
            HandleGrpcErrorToHttp(err, ctx)
            return
        }

        ctx.JSON(http.StatusOK, rsp.Data)
    }

    func GetCategoryBrandList(ctx *gin.Context) {
        id := ctx.Param("id")
        i, err := strconv.ParseInt(id, 10, 32)
        if err != nil {
            ctx.Status(http.StatusNotFound)
            return
        }

        rsp, err := global.GoodsSrvClient.GetCategoryBrandList(context.Background(),
            Id:int32(i),
        })
        if err != nil {
            HandleGrpcErrorToHttp(err, ctx)
            return
        }

        result := make([]interface{}, 0)
        for _, value := range rsp.Data {
            reMap := make(map[string]interface{})
            reMap["id"] = value.Id
            reMap["name"] = value.Name
            reMap["logo"] = value.Logo

            result = append(result, reMap)
        }

        ctx.JSON(http.StatusOK, result)
    }
```

意见反馈 收藏教程 标记书签

优质IT资源微信x923713

```go
func CategoryBrandList(ctx *gin.Context) {
	rsp, err := global.GoodsSrvClient.CategoryBrandList(context.Background(), &pı
	if err != nil {
		HandleGrpcErrorToHttp(err, ctx)
		return
	}
	reMap := map[string]interface{}{
		"total": rsp.Total,
	}

	result := make([]interface{}, 0)
	for _, value := range rsp.Data {
		reMap := make(map[string]interface{})
		reMap["id"] = value.Id
		reMap["category"] = map[string]interface{}{
			"id": value.Category.Id,
			"name": value.Category.Name,
		}
		reMap["brand"] = map[string]interface{}{
			"id": value.Brand.Id,
			"name": value.Brand.Name,
			"logo": value.Brand.Logo,
		}

		result = append(result, reMap)
	}

	reMap["data"] = result
	ctx.JSON(http.StatusOK, reMap)
}

func NewCategoryBrand(ctx *gin.Context) {
	categoryBrandForm := forms.CategoryBrandForm{}
	if err := ctx.ShouldBindJSON(&categoryBrandForm); err != nil {
		HandleValidatorError(ctx, err)
		return
	}

	rsp, err := global.GoodsSrvClient.CreateCategoryBrand(context.Background(), &
		CategoryId: int32(categoryBrandForm.CategoryId),
		BrandId:      int32(categoryBrandForm.BrandId),
	})

	if err != nil {
		HandleGrpcErrorToHttp(err, ctx)
		return
	}

	response := make(map[string]interface{})
	response["id"] = rsp.Id

	ctx.JSON(http.StatusOK, response)
}

func UpdateCategoryBrand(ctx *gin.Context) {
	categoryBrandForm := forms.CategoryBrandForm{}
	if err := ctx.ShouldBindJSON(&categoryBrandForm); err != nil {
		HandleValidatorError(ctx, err)
		return
	}

	id := ctx.Param("id")
	i, err := strconv.ParseInt(id, 10, 32)
	if err != nil {
```

意见反馈　收藏教程　标记书签

优质IT资源微信x923713

```
            return
        }

        _, err = global.GoodsSrvClient.UpdateCategoryBrand(context.Background(), &prc
            Id:         int32(i),
            CategoryId:     int32(categoryBrandForm.CategoryId),
            BrandId:       int32(categoryBrandForm.BrandId),
        })
        if err != nil {
            HandleGrpcErrorToHttp(err, ctx)
            return
        }
        ctx.Status(http.StatusOK)
    }


    func DeleteCategoryBrand(ctx *gin.Context) {
        id := ctx.Param("id")
        i, err := strconv.ParseInt(id, 10, 32)
        if err != nil {
            ctx.Status(http.StatusNotFound)
            return
        }
        _, err = global.GoodsSrvClient.DeleteCategoryBrand(context.Background(), &prc
        if err != nil {
            HandleGrpcErrorToHttp(err, ctx)
            return
        }

        ctx.JSON(http.StatusOK, "")
    }
```

3. router

<> 代码块

```
package router

import (
    "github.com/gin-gonic/gin"
    "mxshop-api/goods-web/api/brands"
)

func InitBrandRouter(Router *gin.RouterGroup) {
    BrandRouter := Router.Group("brands")
    {

        BrandRouter.GET("", brands.BrandList)          // 品牌列表页
        BrandRouter.DELETE("/:id", brands.DeleteBrand) // 删除品牌
        BrandRouter.POST("", brands.NewBrand)       //新建品牌
        BrandRouter.PUT("/:id", brands.UpdateBrand) //修改品牌信息
    }

    CategoryBrandRouter := Router.Group("categorybrands")
    {
        CategoryBrandRouter.GET("", brands.CategoryBrandList)          // 类别品牌
        CategoryBrandRouter.DELETE("/:id", brands.DeleteCategoryBrand) // 删除类别
        CategoryBrandRouter.POST("", brands.NewCategoryBrand)       //新建类别品牌
        CategoryBrandRouter.PUT("/:id", brands.UpdateCategoryBrand) //修改类别品牌
        CategoryBrandRouter.GET("/:id", brands.GetCategoryBrandList) //获取分类的品
    }
```

意见反馈 收藏教程 标记书签

优质IT资源微信x923713

4. 在初始化配置router

5. 轮播图接口 ◂ 上一节 下一节 ▸ 7. oss快速入门

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