优质IT资源微信XQ23713



《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

Q



我自

쿻

 \Box

0

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 > 6. 品牌分类

全部开发者教程 :Ξ

3. go代码规范

4. go项目目录规范

5. 微服务该采用multi-repo还 是mono-repo?

6. 微服务的目录结构(mono-repo)

7. go代码的检测工具

8. go中常见的错误

第22周 设计模式和单元测试

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

```
bobby・更新干 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
      package forms
  1
  2
      type BrandForm struct {
  3
          Name string `form:"name" json:"name" binding:"required,min=3,max=10"`
  4
          Logo string `form:"logo" json:"logo" binding:"url"`
  5
      }
  6
  7
  8
      type CategoryBrandForm struct {
  9
                            `form:"category_id" json:"category_id" binding:"required"`
          CategoryId int
 10
          BrandId int    `form:"brand_id" json:"brand_id" binding:"required"`
 11
                                                                                           12
 13
                                                                                           ②
```

2. handler

```
<> 代码块
 1
    package brands
 2
 3
    import (
         "context"
 4
 5
         "github.com/gin-gonic/gin"
 6
         "github.com/go-playground/validator/v10"
         "github.com/golang/protobuf/ptypes/empty"
 7
         "google.golang.org/grpc/codes"
 8
         "google.golang.org/grpc/status"
 9
         "mxshop-api/goods-web/forms"
10
         "mxshop-api/goods-web/global"
11
12
         "mxshop-api/goods-web/proto"
13
         "net/http"
         "strconv"
14
         "strings"
15
16
17
     func removeTopStruct(fields map[string]string) map[string]string{
18
19
         rsp := map[string]string{}
         for field, err := range fields {
20
             rsp[field[strings.Index(field, ".")+1:]] = err
21
22
                                                                                         不
```

□ 标记书签

╱ 意见反馈

♡ 收藏教程

优质IT资源微。信x923漏13

```
26
    func HandleGrpcErrorToHttp(err error, c *gin.Context) {
27
        //将grpc的code转换成http的状态码
        if err != nil {
29
            if e, ok := status.FromError(err); ok {
30
31
                switch e.Code() {
                case codes.NotFound:
32
                    c.JSON(http.StatusNotFound. gin.H{
33
                         "msg": e.Message(),
34
35
                    })
                case codes.Internal:
36
                    c.JSON(http.StatusInternalServerError, gin.H{
37
                         "msg:": "内部错误",
38
39
                case codes.InvalidArgument:
40
                    c.JSON(http.StatusBadRequest, gin.H{
41
42
                        "msg": "参数错误",
43
                    })
                case codes.Unavailable:
44
45
                    c.JSON(http.StatusInternalServerError, gin.H{
                         "msg": "用户服务不可用",
46
                    })
47
48
                default:
                     c.JSON(http.StatusInternalServerError, gin.H{
49
                         "msq": e.Code(),
50
                    })
51
                }
52
53
                return
            }
54
        }
55
56
    }
57
    func HandleValidatorError(c *gin.Context, err error){
5.8
        errs, ok := err.(validator.ValidationErrors)
59
        if !ok {
60
            c.JSON(http.StatusOK, gin.H{
61
                "msg":err.Error(),
62
63
            })
64
        c.JSON(http.StatusBadRequest, gin.H{
65
            "error": removeTopStruct(errs.Translate(global.Trans)),
66
        return
68
69
    }
70
71
    func BrandList(ctx *gin.Context) {
        pn := ctx.DefaultQuery("pn", "0")
72
73
        pnInt, _ := strconv.Atoi(pn)
        pSize := ctx.DefaultQuery("psize", "10")
74
        pSizeInt, _ := strconv.Atoi(pSize)
75
76
77
        rsp, err := global.GoodsSrvClient.BrandList(context.Background(), &proto.Bran
78
            Pages: int32(pnInt),
            PagePerNums: int32(pSizeInt),
79
        })
80
        if err != nil {
82
            api.HandleGrpcErrorToHttp(err, ctx)
83
84
            return
85
86
87
        result := make([]interface{}, 0)
                                                                                        不
       ▶ 意见反馈
                        ♡ 收藏教程
                                         □ 标记书签
```

②

 \Box

优质IT资源微信x923漏13

```
_, value := range rsp.Data[pnInt:pnInt*pSizeInt+pSizeInt] {
             reMap := make(map[string]interface{})
 91
             reMap["id"] = value.Id
 92
             reMap["name"] = value.Name
 93
             reMap["logo"] = value.Logo
 94
 95
 96
              result = append(result, reMap)
 97
 98
         reMap["data"] = result
 99
100
         ctx.JSON(http.StatusOK, reMap)
101
102
103
104
     func NewBrand(ctx *gin.Context) {
         brandForm := forms.BrandForm{}
105
         if err := ctx.ShouldBindJSON(&brandForm); err != nil {
106
107
             HandleValidatorError(ctx, err)
108
             return
109
110
         rsp, err := global.GoodsSrvClient.CreateBrand(context.Background(), &proto.Br
111
             Name: brandForm.Name,
112
             Logo: brandForm.Logo,
113
114
         })
115
         if err != nil {
116
117
             HandleGrpcErrorToHttp(err, ctx)
118
             return
         }
119
120
121
         request := make(map[string]interface{})
         request["id"] = rsp.Id
122
         request["name"] = rsp.Name
123
         request["logo"] = rsp.Logo
124
125
         ctx.JSON(http.StatusOK, request)
126
127
     }
128
129
     func DeleteBrand(ctx *gin.Context) {
         id := ctx.Param("id")
130
         i, err := strconv.ParseInt(id, 10, 32)
131
         if err != nil {
132
             ctx.Status(http.StatusNotFound)
133
             return
134
135
         }
         _, err = global.GoodsSrvClient.DeleteBrand(context.Background(), &proto.Brand
136
         if err != nil {
137
138
             HandleGrpcErrorToHttp(err, ctx)
              return
139
140
141
142
         ctx.Status(http.Status0K)
143
     }
144
     func UpdateBrand(ctx *gin.Context) {
145
         brandForm := forms.BrandForm{}
         if err := ctx.ShouldBindJSON(&brandForm); err != nil {
147
             HandleValidatorError(ctx, err)
148
149
             return
150
151
152
         id := ctx.Param("id")
                                                                                         不
        ▶ 意见反馈
                         ♡ 收藏教程
                                           □ 标记书签
```

⊡

(?)

 \Box

优质IT资源微信x923从13

```
ctx.Status(http.StatusNotFound)
             return
156
         }
157
158
         _, err = global.GoodsSrvClient.UpdateBrand(context.Background(), &proto.Brand
159
             Id: int32(i).
160
             Name: brandForm.Name,
161
             Logo: brandForm.Logo,
162
         })
163
         if err != nil {
164
             HandleGrpcErrorToHttp(err, ctx)
165
             return
166
167
168
         ctx.Status(http.Status0K)
169
170
171
172
     func GetCategoryBrand(ctx *gin.Context) {
         id := ctx.Param("id")
173
         i, err := strconv.ParseInt(id, 10, 32)
174
         if err != nil {
175
             ctx.Status(http.StatusNotFound)
176
             return
177
178
         rsp, err := global.GoodsSrvClient.GetCategoryBrandList(context.Background(),
179
             Id: int32(i),
180
         })
181
182
183
         if err != nil {
             HandleGrpcErrorToHttp(err, ctx)
184
185
             return
186
187
         ctx.JSON(http.StatusOK, rsp.Data)
188
189
     }
190
     func GetCategoryBrandList(ctx *gin.Context) {
191
         id := ctx.Param("id")
192
193
         i, err := strconv.ParseInt(id, 10, 32)
         if err != nil {
194
             ctx.Status(http.StatusNotFound)
195
196
             return
197
198
         rsp, err := global.GoodsSrvClient.GetCategoryBrandList(context.Background(),
199
200
             Id:int32(i),
201
         })
         if err != nil {
202
203
             HandleGrpcErrorToHttp(err, ctx)
              return
204
205
206
207
         result := make([]interface{}, 0)
208
         for _, value := range rsp.Data {
             reMap := make(map[string]interface{})
209
             reMap["id"] = value.Id
210
211
             reMap["name"] = value.Name
             reMap["logo"] = value.Logo
212
213
214
             result = append(result, reMap)
215
216
217
         ctx.JSON(http.StatusOK, result)
        ▶ 意见反馈
                         ♡ 收藏教程
                                           □ 标记书签
```

不

⊡

②

 \Box

优质IT资源微信x923从13

```
func CategoryBrandList(ctx *gin.Context) {
         rsp, err := global.GoodsSrvClient.CategoryBrandList(context.Background(), &pr
221
222
         if err != nil {
             HandleGrpcErrorToHttp(err, ctx)
223
224
              return
         }
225
226
         reMap := map[string]interface{}{
             "total": rsp.Total,
227
228
229
         result := make([]interface{}, 0)
230
         for _, value := range rsp.Data {
231
              reMap := make(map[string]interface{})
232
              reMap["id"] = value.Id
233
234
              reMap["category"] = map[string]interface{}{
                  "id": value.Categorv.Id.
235
                  "name": value.Category.Name,
236
237
             }
              reMap["brand"] = map[string]interface{}{
238
                  "id": value.Brand.Id.
239
                  "name": value.Brand.Name,
240
                  "logo": value.Brand.Logo,
242
243
              result = append(result, reMap)
244
245
246
247
         reMap["data"] = result
248
         ctx.JSON(http.StatusOK, reMap)
249
250
     func NewCategoryBrand(ctx *gin.Context) {
251
         categoryBrandForm := forms.CategoryBrandForm{}
252
         if err := ctx.ShouldBindJSON(&categoryBrandForm); err != nil {
253
254
             HandleValidatorError(ctx, err)
              return
255
256
257
258
         rsp, err := global.GoodsSrvClient.CreateCategoryBrand(context.Background(), &
             CategoryId: int32(categoryBrandForm.CategoryId),
259
                             int32(categoryBrandForm.BrandId),
              BrandId:
260
         })
261
262
         if err != nil {
263
             HandleGrpcErrorToHttp(err, ctx)
264
265
              return
266
267
268
         response := make(map[string]interface{})
         response["id"] = rsp.Id
270
         ctx.JSON(http.StatusOK, response)
271
272
     }
273
     func UpdateCategoryBrand(ctx *gin.Context) {
274
         categoryBrandForm := forms.CategoryBrandForm{}
275
         if err := ctx.ShouldBindJSON(&categoryBrandForm); err != nil {
276
             HandleValidatorError(ctx, err)
277
              return
278
279
         }
280
         id := ctx.Param("id")
281
282
         i, err := strconv.ParseInt(id, 10, 32)
                                                                                         不
        ▶ 意见反馈
                         ♡ 收藏教程
                                           □ 标记书签
```

(?)

 \Box

 \odot

优质IT资源微。信x923系13

```
return
         }
286
287
          _, err = global.GoodsSrvClient.UpdateCategoryBrand(context.Background(), &prc
288
289
                          int32(i),
                               int32(categoryBrandForm.CategoryId),
             CategorvId:
290
291
             BrandId:
                             int32(categoryBrandForm.BrandId),
         })
292
          if err != nil {
293
             HandleGrpcErrorToHttp(err, ctx)
294
              return
295
296
          ctx.Status(http.Status0K)
297
298
299
300
     func DeleteCategoryBrand(ctx *gin.Context) {
301
         id := ctx.Param("id")
302
303
          i, err := strconv.ParseInt(id, 10, 32)
         if err != nil {
304
305
             ctx.Status(http.StatusNotFound)
306
307
         _, err = global.GoodsSrvClient.DeleteCategoryBrand(context.Background(), &prc
308
         if err != nil {
309
             HandleGrpcErrorToHttp(err, ctx)
310
              return
311
312
313
         ctx.JSON(http.StatusOK, "")
314
315
```

3. router

```
<> 代码块
     package router
 1
 2
     import (
 3
         "github.com/gin-gonic/gin"
 4
 5
         "mxshop-api/goods-web/api/brands"
 6
 7
 8
     func InitBrandRouter(Router *gin.RouterGroup) {
         BrandRouter := Router.Group("brands")
         {
10
11
             BrandRouter.GET("", brands.BrandList)
                                                           // 品牌列表页
12
             BrandRouter.DELETE("/:id", brands.DeleteBrand) // 删除品牌
13
             BrandRouter.POST("", brands.NewBrand)
14
                                                        //新建品牌
             BrandRouter.PUT("/:id", brands.UpdateBrand) //修改品牌信息
15
16
17
         CategoryBrandRouter := Router.Group("categorybrands")
18
19
20
             CategoryBrandRouter.GET("", brands.CategoryBrandList)
                                                                           // 类别品牌
             CategoryBrandRouter.DELETE("/:id", brands.DeleteCategoryBrand) // 删除类别
21
             CategoryBrandRouter.POST("", brands.NewCategoryBrand)
22
                                                                        //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
24
25
                                                                                     不
```

□ 标记书签

▶ 意见反馈

♡ 收藏教程

(?)

 \Box

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





