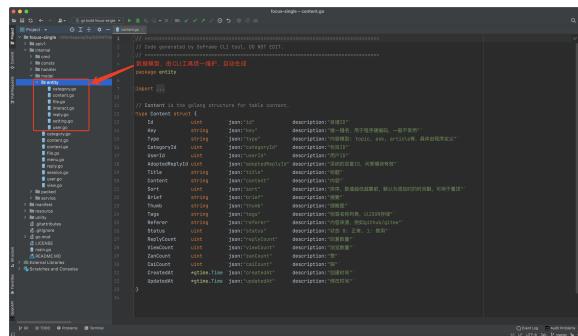


'''

GoFrame

MySQLRedisMongoDBKafka/internal/model/entityGoFrameCLI



```

// This generated by GoFrame CLI tool. DO NOT EDIT
// https://github.com/taobao/goframe
// https://gitee.com/taobao/goframe

// 密码
entity struct {
    // 基本信息
    Id string `json:"id" description:"ID"`
    Type string `json:"type" description:"对象类型：model, service, controller, logic, api"`
    CategoryId string `json:"category_id" description:"分类ID"`
    AbsentOrFieldId string `json:"absent_or_field_id" description:"未使用或字段ID"`
    Title string `json:"title" description:"标题"`
    Content string `json:"content" description:"内容"`
    Brief string `json:"brief" description:"简介"`
    Thumb string `json:"thumb" description:"缩略图"`
    Tags string `json:"tags" description:"标签, 使用逗号分隔, 例如: tag1, tag2"`
    Author string `json:"author" description:"作者"`
    Status int `json:"status" description:"状态, 0: 正常, 1: 禁用"`
    ReplyCount int `json:"reply_count" description:"回复数"`
    LikeCount int `json:"like_count" description:"点赞数"`
    CommentCount int `json:"comment_count" description:"评论数"`
    CreateTime string `json:"create_time" description:"创建时间"`
    UpdateTime string `json:"update_time" description:"更新时间"`
}

```

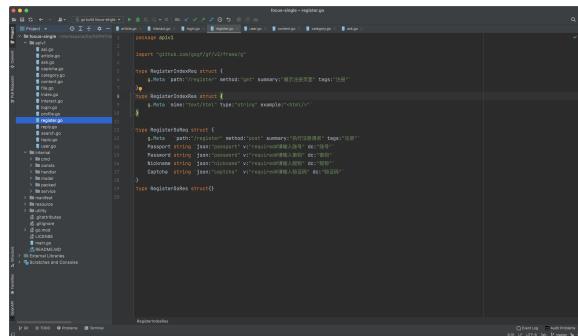
## Content Menu

- /
- /
- DO

//

/

//apicontroller, logic, modelapiapimodelGoFrameXxxReqXxxRes



```

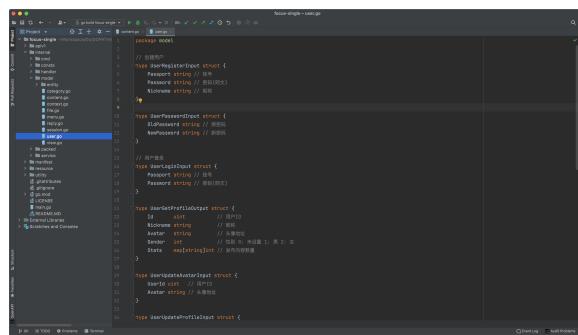
// 密码
entity struct {
    // 基本信息
    Id string `json:"id" description:"ID"`
    Type string `json:"type" description:"对象类型：model, service, controller, logic, api"`
    CategoryId string `json:"category_id" description:"分类ID"`
    AbsentOrFieldId string `json:"absent_or_field_id" description:"未使用或字段ID"`
    Title string `json:"title" description:"标题"`
    Content string `json:"content" description:"内容"`
    Brief string `json:"brief" description:"简介"`
    Thumb string `json:"thumb" description:"缩略图"`
    Tags string `json:"tags" description:"标签, 使用逗号分隔, 例如: tag1, tag2"`
    Author string `json:"author" description:"作者"`
    Status int `json:"status" description:"状态, 0: 正常, 1: 禁用"`
    ReplyCount int `json:"reply_count" description:"回复数"`
    LikeCount int `json:"like_count" description:"点赞数"`
    CommentCount int `json:"comment_count" description:"评论数"`
    CreateTime string `json:"create_time" description:"创建时间"`
    UpdateTime string `json:"update_time" description:"更新时间"`
}

// 登录
entity struct {
    // 基本信息
    Id string `json:"id" description:"ID"`
    Type string `json:"type" description:"对象类型：model, service, controller, logic, api"`
    CategoryId string `json:"category_id" description:"分类ID"`
    AbsentOrFieldId string `json:"absent_or_field_id" description:"未使用或字段ID"`
    Title string `json:"title" description:"标题"`
    Content string `json:"content" description:"内容"`
    Brief string `json:"brief" description:"简介"`
    Thumb string `json:"thumb" description:"缩略图"`
    Tags string `json:"tags" description:"标签, 使用逗号分隔, 例如: tag1, tag2"`
    Author string `json:"author" description:"作者"`
    Status int `json:"status" description:"状态, 0: 正常, 1: 禁用"`
    ReplyCount int `json:"reply_count" description:"回复数"`
    LikeCount int `json:"like_count" description:"点赞数"`
    CommentCount int `json:"comment_count" description:"评论数"`
    CreateTime string `json:"create_time" description:"创建时间"`
    UpdateTime string `json:"update_time" description:"更新时间"`
}

```

/

//controller->serviceservice->servicemodelGoFrameXxxInputXxxOutput



```

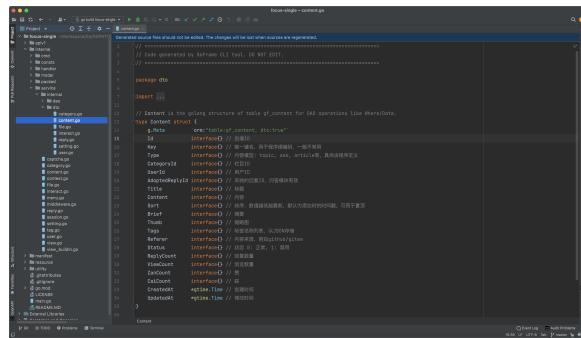
// 密码
entity struct {
    // 基本信息
    Id string `json:"id" description:"ID"`
    Type string `json:"type" description:"对象类型：model, service, controller, logic, api"`
    CategoryId string `json:"category_id" description:"分类ID"`
    AbsentOrFieldId string `json:"absent_or_field_id" description:"未使用或字段ID"`
    Title string `json:"title" description:"标题"`
    Content string `json:"content" description:"内容"`
    Brief string `json:"brief" description:"简介"`
    Thumb string `json:"thumb" description:"缩略图"`
    Tags string `json:"tags" description:"标签, 使用逗号分隔, 例如: tag1, tag2"`
    Author string `json:"author" description:"作者"`
    Status int `json:"status" description:"状态, 0: 正常, 1: 禁用"`
    ReplyCount int `json:"reply_count" description:"回复数"`
    LikeCount int `json:"like_count" description:"点赞数"`
    CommentCount int `json:"comment_count" description:"评论数"`
    CreateTime string `json:"create_time" description:"创建时间"`
    UpdateTime string `json:"update_time" description:"更新时间"`
}

// 登录
entity struct {
    // 基本信息
    Id string `json:"id" description:"ID"`
    Type string `json:"type" description:"对象类型：model, service, controller, logic, api"`
    CategoryId string `json:"category_id" description:"分类ID"`
    AbsentOrFieldId string `json:"absent_or_field_id" description:"未使用或字段ID"`
    Title string `json:"title" description:"标题"`
    Content string `json:"content" description:"内容"`
    Brief string `json:"brief" description:"简介"`
    Thumb string `json:"thumb" description:"缩略图"`
    Tags string `json:"tags" description:"标签, 使用逗号分隔, 例如: tag1, tag2"`
    Author string `json:"author" description:"作者"`
    Status int `json:"status" description:"状态, 0: 正常, 1: 禁用"`
    ReplyCount int `json:"reply_count" description:"回复数"`
    LikeCount int `json:"like_count" description:"点赞数"`
    CommentCount int `json:"comment_count" description:"评论数"`
    CreateTime string `json:"create_time" description:"创建时间"`
    UpdateTime string `json:"update_time" description:"更新时间"`
}

```

DO

GoFrameDOORMDAO



The screenshot shows a code editor with a file named "GoFrameDOORMDAO.go". The code is generated by a tool and contains annotations like "Generated by GoFrame CLI tool, DO NOT EDIT". It defines a struct "YfContext" with fields: "key" (interface{}), "type" (interface{}), "ownerId" (interface{}), "isContent" (bool), "title" (interface{}), "content" (interface{}), "list" (interface{}), "tags" (interface{}), and "before" (interface{}). It also includes methods: "GetContent", "GetList", "GetTags", "GetBefore", "GetIsContent", "GetOwnerId", "GetTitle", "GetKey", "GetType", "GetListCount", "GetTagCount", and "GetBeforeCount". A timestamp "updateTime" is also present.

```
local:go> GoFrameDOORMDAO.go
package goframe

// Generated by GoFrame CLI tool, DO NOT EDIT.
// See https://github.com/taobao/goframe

// YfContext is the entity structure of table yf_context for Dao operations like Insert/Update.
type YfContext struct {
    key           interface{} // 唯一标识，字符串
    type          interface{} // 内容类型，字符串
    ownerId      interface{} // 用户ID
    isContent     bool        // 是否是内容
    title         interface{} // 标题
    content       interface{} // 内容
    list          interface{} // 列表
    tags          interface{} // 标签
    before        interface{} // 前置
}

func (c *YfContext) GetContent() interface{} {
    return c.content
}

func (c *YfContext) GetList() interface{} {
    return c.list
}

func (c *YfContext) GetTags() interface{} {
    return c.tags
}

func (c *YfContext) GetBefore() interface{} {
    return c.before
}

func (c *YfContext) GetIsContent() bool {
    return c.isContent
}

func (c *YfContext) GetOwnerId() interface{} {
    return c.ownerId
}

func (c *YfContext) GetTitle() interface{} {
    return c.title
}

func (c *YfContext) GetKey() interface{} {
    return c.key
}

func (c *YfContext) GetType() interface{} {
    return c.type
}

func (c *YfContext) GetListCount() int {
    return len(c.list)
}

func (c *YfContext) GetTagCount() int {
    return len(c.tags)
}

func (c *YfContext) GetBeforeCount() int {
    return len(c.before)
}

var updateTime = "2018-07-10 10:00:00"
```

DODAO

logic