

-OpenAPIv3

 OpenAPIv3-

OpenAPIv3

OpenAPIv3<https://swagger.io/specification/>

g.Meta

embedded g.Meta g.Meta

-gmeta

OpenAPIv3OpenAPIv3

Server

1

OpenAPIv3		
path	URI	g.Meta
tags		g.Meta
method	GET/PUT/POST/DELETE... ()	g.Meta
deprecated		g.Meta
summary	/	sm
description	/	dc
in		header/path/query /cookie
default		d
mime	MIMEmultipart/form-dataapplication/json	g.Meta
type	file	

 OpenAPIv3<https://swagger.io/specification/>

2

OpenAPIx-Golang struct tag

Content Menu

- [OpenAPIv3](#)
- [g.Meta](#)
- - 1
 - 2
- [OpenAPIv3](#)

```

package main

import (
    "context"

    "github.com/gogf/gf/v2/errors/gcode"
    "github.com/gogf/gf/v2/errors/gerror"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
)

type GetListReq struct {
    g.Meta `path:"/user" tags:"User" method:"get" x-group:"User/Info"
    summary:"Get user list with basic info."`
    Page  int `dc:"Page number" d:"1" x-sort:"1"`
    Size  int `dc:"Size for per page." d:"10" x-sort:"2"`
}

type GetListRes struct{}

type Controller struct{}

func (c *Controller) GetList(ctx context.Context, req *GetListReq) (res
*GetListRes, err error) {
    return nil, gerror.NewCode(gcode.CodeNotImplemented)
}

func main() {
    s := g.Server()
    s.Group("/", func(group *ghttp.RouterGroup) {
        group.Bind(new(Controller))
    })
    s.SetOpenApiPath("/api.json")
    s.SetSwaggerPath("/swagger")
    s.SetPort(8199)
    s.Run()
}

```

<http://127.0.0.1:8199/swagger> swagger ui <http://127.0.0.1:8199/api.json> OpenAPIv3OpenAPIv3

```

{
  "openapi": "3.0.0",
  "components": {
    "schemas": {
      "main.GetListReq": {
        "properties": {
          "Page": {
            "default": 1,
            "description": "Page number",
            "format": "int",
            "properties": {},
            "type": "integer",
            "x-sort": "1"
          },
          "Size": {
            "default": 10,
            "description": "Size for per page.",
            "format": "int",
            "properties": {},
            "type": "integer",
            "x-sort": "2"
          }
        },
        "type": "object",
        "x-group": "User/Info"
      },
      "main.GetListRes": {
        "properties": {},
        "type": "object"
      }
    }
  }
}

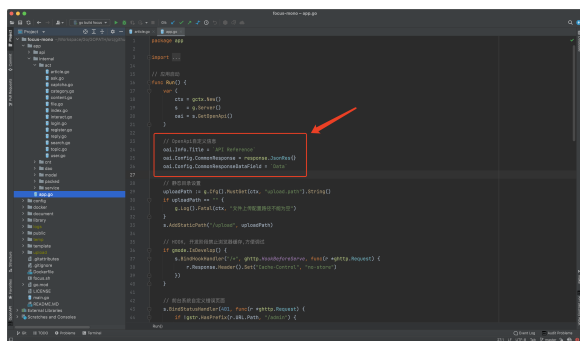
```

```

    }
  },
  "info": {
    "title": "",
    "version": ""
  },
  "paths": {
    "/user": {
      "get": {
        "parameters": [
          {
            "description": "Page number",
            "in": "query",
            "name": "Page",
            "schema": {
              "default": 1,
              "description": "Page number",
              "format": "int",
              "properties": {},
              "type": "integer",
              "x-sort": "1"
            },
            "x-sort": "1"
          },
          {
            "description": "Size for per page.",
            "in": "query",
            "name": "Size",
            "schema": {
              "default": 10,
              "description": "Size for per page.",
              "format": "int",
              "properties": {},
              "type": "integer",
              "x-sort": "2"
            },
            "x-sort": "2"
          }
        ],
        "responses": {
          "200": {
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/main.
GetListRes"
                }
              }
            },
            "description": ""
          }
        },
        "summary": "Get user list with basic info.",
        "tags": [
          "User"
        ],
        "x-group": "User/Info"
      },
      "x-group": "User/Info"
    }
  }
}

```

s.GetOpenApi()OpenAPIv3



OpenAPIv3