[https://github.com/go-swagger/go-swagger](C:/Program%20Files%20(x86)/TeamTalk/data/msgview/javascript:void(0);)

[https://doc.myoas.com/pages/viewpage.action?pageId=325232293](C:/Program%20Files%20(x86)/TeamTalk/data/msgview/javascript:void(0);)

[http://10.177.163.32:7081/swagger-ui.html#!/%E7%A7%98%E9%92%A5%E7%AE%A1%E7%90%86%E6%8E%A5%E5%8F%A3](C:/Program%20Files%20(x86)/TeamTalk/data/msgview/javascript:void(0);)(%E7%AE%A1%E7%90%86%E5%91%98%E4%B8%93%E7%94%A8)/listAllAppsUsingGET

[https://swaggo.github.io/swaggo.io/cli/#-dir-value](https://swaggo.github.io/swaggo.io/cli/" \l "-dir-value)

[https://github.com/swaggo/swag/blob/master/README\_zh-CN.md#%E9%A1%B9%E7%9B%AE%E7%9B%B8%E5%85%B3](https://github.com/swaggo/swag/blob/master/README_zh-CN.md" \l "%E9%A1%B9%E7%9B%AE%E7%9B%B8%E5%85%B3)

<https://swagger.io/docs/specification/2-0/basic-structure/>

首先需要github.com/swaggo/gin-swagger和github.com/swaggo/gin-swagger/swaggerFiles。然后根据 github.com/swaggo/swag/cmd/swag文档获取到swag工具；执行swag init在项目根目录下生成docs文件夹。然后在路由中import \_ "/docs"。这时候编译程序，打开http://localhost:8080/swagger/index.html就可以看到API。

通用API信息

General API Info

| annotation | description | example |
| --- | --- | --- |
| **title** | **Required. The title of the application.** | **// @title Swagger Example API** |
| **version** | **Required. Provides the version of the application API.** | **// @version 1.0** |
| description | A short description of the application. | // @description This is a sample server celler server. |
| termsOfService | The Terms of Service for the API. | // @termsOfService <http://swagger.io/terms/> |
| contact.name | The contact information for the exposed API. | // @contact.name API Support |
| contact.url | The URL pointing to the contact information. MUST be in the format of a URL. | // @contact.url <http://www.swagger.io/support> |
| contact.email | The email address of the contact person/organization. MUST be in the format of an email address. | // @contact.email support@swagger.io |
| license.name | Required.demo显示，好像不是必须的。The license name used for the API. | // @license.name Apache 2.0 |
| license.url | A URL to the license used for the API. MUST be in the format of a URL. | // @license.url <http://www.apache.org/licenses/LICENSE-2.0.html> |
| **host** | **The host (name or ip) serving the API.** | **// @host localhost:8080** |
| **BasePath** | **The base path on which the API is served.** | **// @BasePath /api/v1** |

安全性

| 注释 | 描述 | 参数 | 示例 |
| --- | --- | --- | --- |
| securitydefinitions.basic | [Basic](https://swagger.io/docs/specification/2-0/authentication/basic-authentication/) auth. |  | // @securityDefinitions.basic BasicAuth |
| securitydefinitions.apikey | [API key](https://swagger.io/docs/specification/2-0/authentication/api-keys/) auth. | in, name | // @securityDefinitions.apikey ApiKeyAuth |
| securitydefinitions.oauth2.application | [OAuth2 application](https://swagger.io/docs/specification/authentication/oauth2/) auth. | tokenUrl, scope | // @securitydefinitions.oauth2.application OAuth2Application |
| securitydefinitions.oauth2.implicit | [OAuth2 implicit](https://swagger.io/docs/specification/authentication/oauth2/) auth. | authorizationUrl, scope | // @securitydefinitions.oauth2.implicit OAuth2Implicit |
| securitydefinitions.oauth2.password | [OAuth2 password](https://swagger.io/docs/specification/authentication/oauth2/) auth. | tokenUrl, scope | // @securitydefinitions.oauth2.password OAuth2Password |
| securitydefinitions.oauth2.accessCode | [OAuth2 access code](https://swagger.io/docs/specification/authentication/oauth2/) auth. | tokenUrl, authorizationUrl, scope | // @securitydefinitions.oauth2.accessCode OAuth2AccessCode |

| 参数注释 | 示例 |
| --- | --- |
| in | // @in header |
| name | // @name Authorization |
| tokenUrl | // @tokenUrl <https://example.com/oauth/token> |
| authorizationurl | // @authorizationurl <https://example.com/oauth/authorize> |
| scope.hoge | // @scope.write Grants write access |

// @securityDefinitions.basic BasicAuth  
// @securityDefinitions.apikey ApiKeyAuth  
// @in header  
// @name Authorization  
// @securitydefinitions.oauth2.application OAuth2Application  
// @tokenUrl https://example.com/oauth/token  
// @scope.write Grants write access  
// @scope.admin Grants read and write access to administrative information  
// @securitydefinitions.oauth2.implicit OAuth2Implicit  
// @authorizationUrl https://example.com/oauth/authorize  
// @scope.write Grants write access  
// @scope.admin Grants read and write access to administrative information  
// @securitydefinitions.oauth2.password OAuth2Password  
// @tokenUrl https://example.com/oauth/token  
// @scope.read Grants read access  
// @scope.write Grants write access  
// @scope.admin Grants read and write access to administrative information  
// @securitydefinitions.oauth2.accessCode OAuth2AccessCode  
// @tokenUrl https://example.com/oauth/token  
// @authorizationUrl https://example.com/oauth/authorize  
// @scope.admin Grants read and write access to administrative information

示例 [celler/main.go](https://github.com/swaggo/swag/blob/master/example/celler/main.go)

| 注释 | 说明 | 示例 |
| --- | --- | --- |
| **title** | **必填 应用程序的名称。** | **// @title Swagger Example API** |
| **version** | **必填 提供应用程序API的版本。** | **// @version 1.0** |
| description | 应用程序的简短描述。 | // @description This is a sample server celler server. |
| tag.name | 标签的名称。 | // @tag.name This is the name of the tag |
| tag.description | 标签的描述。 | // @tag.description Cool Description |
| tag.docs.url | 标签的外部文档的URL。 | // @tag.docs.url [https://example.com](https://example.com/) |
| tag.docs.description | 标签的外部文档说明。 | // @tag.docs.description Best example documentation |
| termsOfService | API的服务条款。 | // @termsOfService <http://swagger.io/terms/> |
| contact.name | 公开的API的联系信息。 | // @contact.name API Support |
| contact.url | 联系信息的URL。 必须采用网址格式。 | // @contact.url <http://www.swagger.io/support> |
| contact.email | 联系人/组织的电子邮件地址。 必须采用电子邮件地址的格式。 | // @contact.email [support@swagger.io](mailto:support@swagger.io) |
| **license.name** | **必填 用于API的许可证名称。** | **// @license.name Apache 2.0** |
| license.url | 用于API的许可证的URL。 必须采用网址格式。 | // @license.url <http://www.apache.org/licenses/LICENSE-2.0.html> |
| host | 运行API的主机（主机名或IP地址）。 | // @host localhost:8080 |
| BasePath | 运行API的基本路径。 | // @BasePath /api/v1 |
| query.collection.format | 请求URI query里数组参数的默认格式：csv，multi，pipes，tsv，ssv。 如果未设置，则默认为csv。 | // @query.collection.format multi |
| schemes | 用空格分隔的请求的传输协议。 | // @schemes http https |
| x-name | 扩展的键必须以x-开头，并且只能使用json值 | // @x-example-key {"key": "value"} |

使用Markdown描述

如果文档中的短字符串不足以完整表达，或者需要展示图片，代码示例等类似的内容，则可能需要使用Markdown描述。要使用Markdown描述，请使用以下注释。

| 注释 | 说明 | 示例 |
| --- | --- | --- |
| title | 必填 应用程序的名称。 | // @title Swagger Example API |
| version | 必填 提供应用程序API的版本。 | // @version 1.0 |
| description.markdown | 应用程序的简短描述。 从api.md文件中解析。 这是@description的替代用法。 | // @description.markdown No value needed, this parses the description from api.md |
| tag.name | 标签的名称。 | // @tag.name This is the name of the tag |
| tag.description.markdown | 标签说明，这是tag.description的替代用法。 该描述将从名为tagname.md的文件中读取。 | // @tag.description.markdown |

API操作（实现函数前）

Example [celler/controller](https://github.com/swaggo/swag/tree/master/example/celler/controller)

| 注释 | 描述 |
| --- | --- |
| description | 操作行为的详细说明。 |
| description.markdown | 应用程序的简短描述。该描述将从名为endpointname.md的文件中读取。 |
| id | 用于标识操作的唯一字符串。在所有API操作中必须唯一。 |
| tags | 每个API操作的标签列表，以逗号分隔。 |
| summary | 该操作的简短摘要。 |
| accept | API可以使用的MIME类型的列表。值必须如“[Mime类型](https://github.com/swaggo/swag/blob/master/README_zh-CN.md" \l "mime-types)”中所述。 |
| produce | API可以生成的MIME类型的列表。值必须如“[Mime类型](https://github.com/swaggo/swag/blob/master/README_zh-CN.md" \l "mime-types)”中所述。 |
| **param** | **用空格分隔的参数。param name,param type,data type,is mandatory? ,comment attribute(optional)** |
| security | 每个API操作的[安全性](https://github.com/swaggo/swag/blob/master/README_zh-CN.md" \l "security)。 |
| **success** | **以空格分隔的成功响应。return code,{param type},data type,comment** |
| **failure** | **f** |
| **response** | **与success、failure作用相同** |
| header | 以空格分隔的头字段。 return code,{param type},data type,comment |
| router | 以空格分隔的路径定义。 path,[httpMethod] |
| x-name | 扩展字段必须以x-开头，并且只能使用json值。 |

Mime类型

swag 接受所有格式正确的MIME类型, 即使匹配 \*/\*。除此之外，swag还接受某些MIME类型的别名，如下所示：

| Alias | MIME Type |
| --- | --- |
| json | application/json |
| xml | text/xml |
| plain | text/plain |
| html | text/html |
| mpfd | multipart/form-data |
| x-www-form-urlencoded | application/x-www-form-urlencoded |
| json-api | application/vnd.api+json |
| json-stream | application/x-json-stream |
| octet-stream | application/octet-stream |
| png | image/png |
| jpeg | image/jpeg |
| gif | image/gif |

**参数类型**

**query**

**path**

header

**body**

formData

**数据类型**

string (string)

integer (int, uint, uint32, uint64)

number (float32)

boolean (bool)

user defined struct

import (

"github.com/gin-gonic/gin"

"github.com/swaggo/gin-swagger"

"github.com/swaggo/gin-swagger/swaggerFiles"

"net/http"

\_ "gin-demo/docs" // 执行swag init生成的docs文件夹路径 \_引包表示只执行init函数

)

// @Param enumstring query string false "string enums" Enums(A, B, C)

// @Param enumint query int false "int enums" Enums(1, 2, 3)

// @Param enumnumber query number false "int enums" Enums(1.1, 1.2, 1.3)

// @Param string query string false "string valid" minlength(5) maxlength(10)

// @Param int query int false "int valid" minimum(1) maximum(10)

// @Param default query string false "string default" default(A)

// @Param collection query []string false "string collection" collectionFormat(multi)

// @Success 200 {string} json "{"code":200,"data":"name","msg":"ok"}"  
// @Failure 400 {string} json "{"code":400,"data":"name","msg":"400"}"  
// @Failure 404 {string} json "{"code":404,"data":"name","msg":"404"}"  
// @Failure 500 {string} json "{"code":500,"data":"name","msg":"500"}"

Go mod init

Go mod tidy

Go mod vendor

删监控

[http://atms.itest.wanyol.com/scene/viewSceneCase/4373?treeSetId=0](C:/Program%20Files%20(x86)/TeamTalk/data/msgview/javascript:void(0);)

**/home/service/var/parker\_admin/bin**

**admin\_server:10.177.236.32**

**Rm 2**

proxy&&dataserver:

rmparker

/home/service  
10.52.128.98  
10.52.128.108  
10.52.128.109  
10.52.128.215

**Remove proxy 参数为”id”**

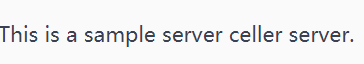
S3 返回的id($data)是 s3的id

**总界面**

// @title Swagger Example API

  
// @version 1.0

  
// @description This is a sample server celler server.



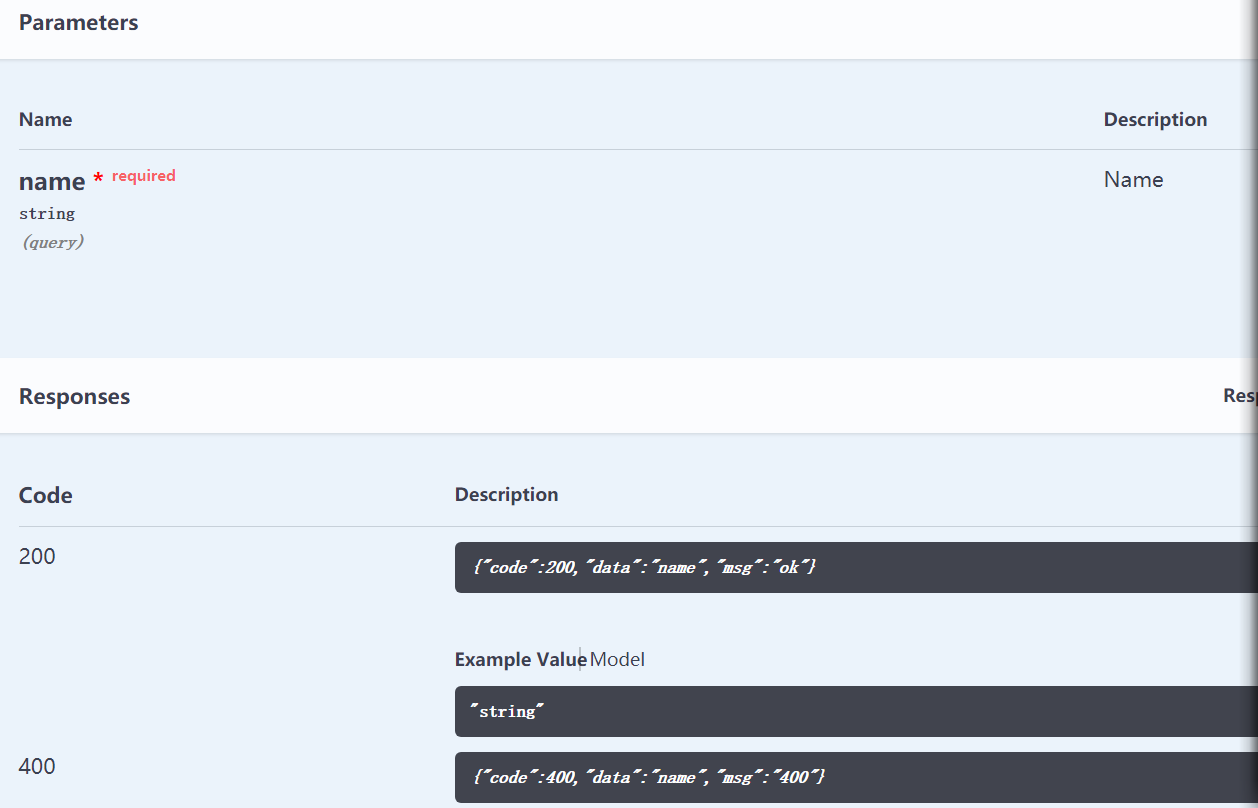
// @termsOfService <http://swagger.io/terms/>



// @host localhost:8080  
// @BasePath /api/v1

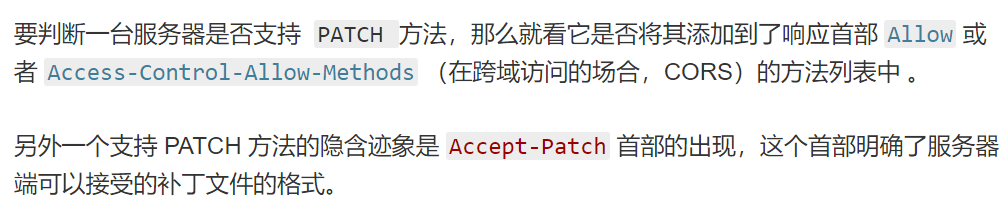


// @Tags accounts  
// @Summary 打印测试功能  
// @Accept json  
// @Produce json  
// @description This is a sample server Petstore server.  
// @Host 127.0.0.1:8080  
// @BasePath /api/v1  
// @Param name query string true "Name"  
// @Success 200 {string} json "{"code":200,"data":"name","msg":"ok"}"  
// @Failure 400 {string} json "{"code":400,"data":"name","msg":"400"}"  
// @Failure 404 {string} json "{"code":404,"data":"name","msg":"404"}"  
// @Failure 500 {string} json "{"code":500,"data":"name","msg":"500"}"  
// @Router /print [get]



 PATCH不会代替整个项目，而只是替换指定的项目





**在包含main.go文件的项目根目录运行swag init**

**如果通用API指数没有写在main.go中，可以使用-g标识符来告知swag。**

**swag init -g http/api.go**

swag init [command options] [arguments...]

OPTIONS:

--generalInfo value, **-g value** API通用信息所在的**go源文件路径**，如果是相对路径则基于API解析目录 (默认: "main.go")

--dir value, **-d value** **API解析目录 (默认: "./")**

--propertyStrategy value, -p value 结构体字段命名规则，三种：snakecase,camelcase,pascalcase (默认: "camelcase")

--output value, **-o value** 文件(swagger.json, swagger.yaml and doc.go)输出目录 (默认: "./docs")

--parseVendor 是否解析vendor目录里的go源文件，默认不

--parseDependency 是否解析依赖目录中的go源文件，默认不

--markdownFiles value, --md value 指定API的描述信息所使用的markdown文件所在的目录

--generatedTime 是否输出时间到输出文件docs.go的顶部，默认是

**当main不在最外层 swag init -g package/main.go**

**具体函数注释前会有**

**// FuncName godoc**

**// @**

**用户自定义的具有数组类型的结构**

**// @Success 200 {array} model.Account**

**package model**

**type Account struct {**

**ID int `json:"id" example:"1"`**

**Name string `json:"name" example:"account name"`**

**}**

**swag init -g cmd/admin\_server/main.go**

// @Failure 400 {string} json "{"code":400,"data":"name","msg":"400"}"  
// @Failure 404 {string} json "{"code":404,"data":"name","msg":"404"}"  
// @Failure 500 {string} json "{"code":500,"data":"name","msg":"500"}"