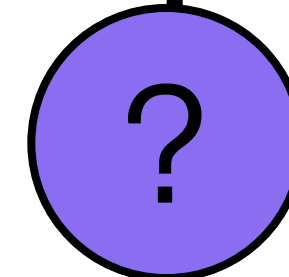


Documentación en Go

¿Qué es Swagger?

Swagger es una herramienta extremadamente útil para describir, producir, consumir y visualizar APIs RESTful.





Swagger en Go

Para implementar Swagger en Go se utilizará **swaggo**. Este nos permite generar la documentación del proyecto utilizando comentarios dentro del código.

```
$ go get -u github.com/swaggo/swag/cmd/swag
go get -u github.com/swaggo/files
go get -u github.com/swaggo/gin-swagger

export PATH=$PATH:$HOME/go/bin
```



Anotaciones en el main del programa

Arriba de la función **main**, se agregan las anotaciones para especificar y documentar a nivel global nuestra aplicación.

```
$ // @title Certified Tech Developer
// @version 1.0
// @description This API Handle Products.
// @termsOfService
https://developers.ctd.com.ar/es_ar/terminos-y-condiciones

// @contact.name API Support
// @contact.url https://developers.ctd.com.ar/support

// @license.name Apache 2.0
// @license.url http://www.apache.org/licenses/LICENSE-2.0.html
func main() {
```



Anotaciones en el controlador GetAll

Arriba del método **GetAll** del controlador, se agregan las anotaciones para documentar el endpoint.

{ }

```
// ListProducts godoc
// @Summary List products
// @Tags Products
// @Description get products
// @Accept json
// @Produce json
// @Param token header string true "token"
// @Success 200 {object} web.Response
// @Router /products [get]
func (c *Product) GetAll() gin.HandlerFunc {
```



Anotaciones en el controlador Store

Arriba del método **Store** del controlador, se agregan las anotaciones para documentar el endpoint.

```
{ }  
// StoreProducts godoc  
// @Summary Store products  
// @Tags Products  
// @Description store products  
// @Accept json  
// @Produce json  
// @Param token header string true "token"  
// @Param product body request true "Product to store"  
// @Success 200 {object} web.Response  
// @Router /products [post]  
func (c *Product) Store() gin.HandlerFunc {
```

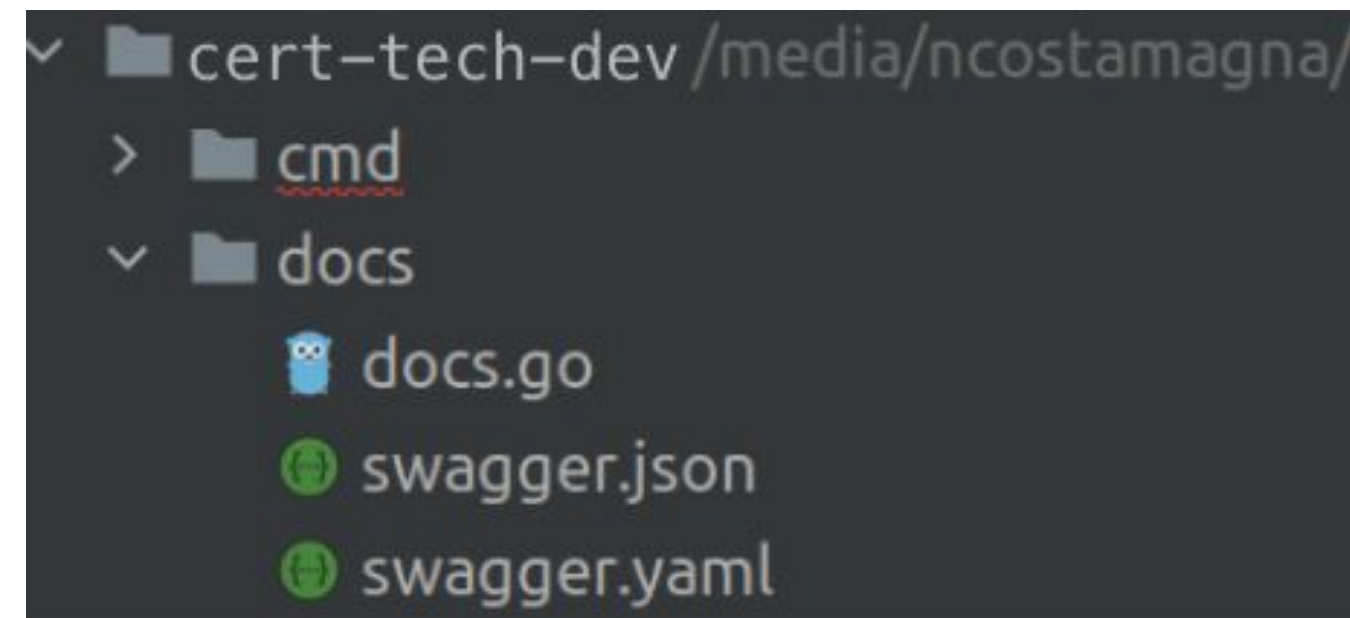


Generar documentación

Para generar la documentación se utilizará el comando que nos proporciona **swaggo**, especificando dónde se encuentra el archivo **main.go**:

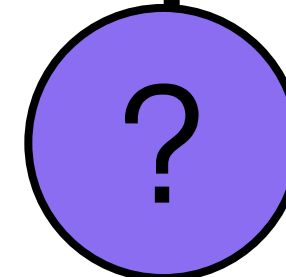
```
$ swag init -g cmd/server/main.go
```

Una vez ejecutado el comando, **swaggo** nos generará un paquete **docs** que contendrá toda la documentación del proyecto basándose en las anotaciones generadas.



¿Cómo se podrá visualizar la documentación?

Para visualizar la documentación, **swaggo** nos proporciona el paquete **gin-swagger** que nos ayudará a visualizar la documentación desde Gin.





Importación de paquetes

Se importan los paquetes **files** y **gin-swagger** que permitirán visualizar la documentación desde un endpoint. También, se debe importar el paquete **docs** que generó swaggo.

```
{ }
```

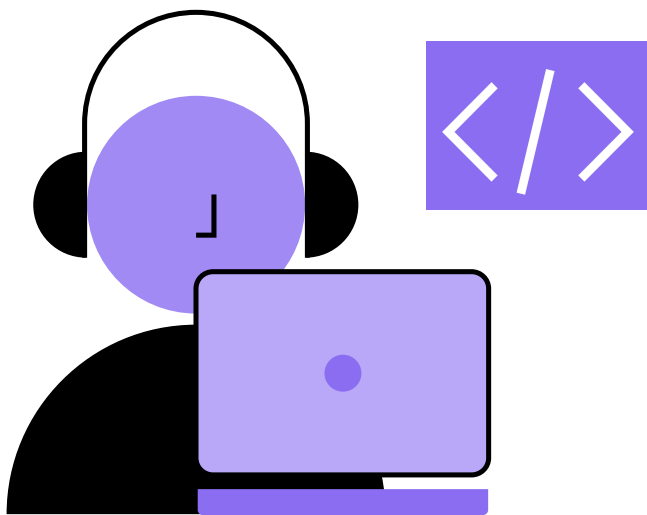
```
"github.com/ncostamagna/meli-bootcamp/docs"  
"github.com/swaggo/files"  
ginSwagger "github.com/swaggo/gin-swagger"
```

Dentro de los **env** se debe agregar la variable **HOST** con la URL base de la aplicación.

```
.env HOST=localhost:8080
```

Main del programa

En el main del programa se debe agregar el endpoint correspondiente a la documentación generada. Debemos asignarle el HOST de la variable de entorno a la documentación.



```
func main() {
    _ = godotenv.Load()
    db := store.New(store.FileType, "./products.json")
    repo := products.NewRepository(db)
    service := products.NewService(repo)
    p := handler.NewProduct(service)
    r := gin.Default()

    docs.SwaggerInfo.Host = os.Getenv("HOST")
    r.GET("/docs/*any",
    ginSwagger.WrapHandler(swaggerFiles.Handler))

    pr := r.Group("/products")
    pr.POST("/", p.Store())
    pr.GET("/", p.GetAll())
    pr.PUT("/:id", p.Update())
    pr.PATCH("/:id", p.UpdateName())
    pr.DELETE("/:id", p.Delete())
    r.Run()
}
```

Documentación

Al ingresar a <http://localhost:8080/docs/index.html> se podrá visualizar la documentación.

Certified Tech Developer 1.0

[Base URL: localhost:8080]
[doc.json](#)

This API Handle MELI Products.
[Terms of service](#)
[API Support - Website](#)
[Apache 2.0](#)

Products ▼

GET	/products	List products
POST	/products	Store products
PUT	/products/{id}	Update products
DELETE	/products/{id}	Delete products
PATCH	/products/{id}	Update products name

¡Muchas gracias!