



OpenAPI 간편해요?

Container App 배포기



발표자

계획

준비

실행

종료

```
{  
  Name: Mireu Lim,  
  Email: limmireu1214@gmail.com,  
  HomePage: www.mireu.xyz,  
  
  Git: mireu-lab,  
  Group: [  
    [대구가톨릭대학교, AI빅데이터공학과],  
    None_Labs  
  ]  
}
```



Mireu Lim

Mireu-Lab

전기는 국산이지만 서버는 수입입니다.



발표자

계획

준비

실행

종료

저는 사실.....

파이썬 개발자예요....

FastAPI, Tensorflow,
Pytorch, Pandas,
Numpy, DockerSDK, Etc.....



발표자

계획

준비

실행

종료

Mireuemon 심심했는데 뭔가 해볼게 생각났네요

Mireuemon ~~GoLang Gin~~아니 ~~Echo~~에서 스웨거 굴리기

Mireuemon 있긴할것같은데

Mireuemon 있더라고요



발표자

계획

준비

실행

종료

그는 GoDoc이라는 기능을 모르고
일단은 찌르고봤던!

default

GET

/programming_languages List Programming Languages

그래서 꼬와서 만들게 된 Golang에다가 Swagger올리기!

GET

/programming_languages/{programming_language_id} Get Programming Language

PUT

/programming_languages/{programming_language_id} Update Programming Language

DELETE

/programming_languages/{programming_language_id} Delete Programming Language



발표자

계획

준비

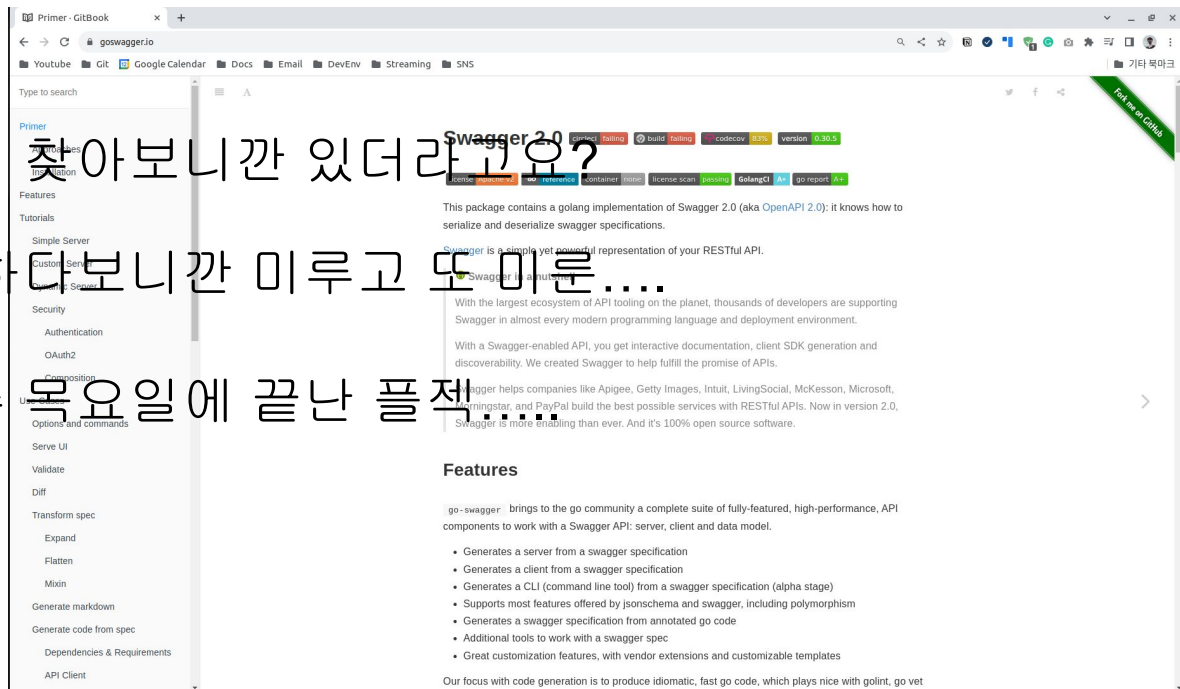
실행

종료

그전에 찾아보니까 있더라구요?

근데 하다보니까 미루고 또 미룬....

이번주 목요일에 끝난 플젝





발표자

계획

준비

실행

종료

목표

- GET, POST, DELETE, PUT

- Auth



발표자

계획

준비

실행

종료

진행하기 앞서

Repo에 있는 코드를 읽으면서
들어주세요!



Mireu-Lab/Golang-OpenAPI



발표자

계획

준비

실행

종료

일단은 실행 해볼까요?



발표자

계획

준비

실행

종료

- GET

```
1 // @BasePath /
2 // PingExample godoc
3 // @Schemes
4 // @Description do ping
5 // @Accept json
6 // @Produce json
7 // @Success 200 {string} Helloworld
8 // @Router /helloworld [get]
9 func Helloworld(g *gin.Context) {
10     g.JSONP(http.StatusOK, H{"msg": "helloworld"})
11 }
```



발표자

계획

준비

실행

종료

- POST

```
1 // @BasePath /
2 // @Schemes http,https
3 // @Accept json
4 // @Produce json
5 // @Param filename query string true "The filename"
6 // @Success 200 {object} H{filename string} "Success"
7 // @Router /createFile [post]
8 func postfile(g *gin.Context) {
9     filename := g.Query("filename")
10
11     // Create the file.
12     file, _ := os.Create("file/" + filename)
13     defer file.Close()
14
15     // Return the filename.
16     g.JSON(http.StatusOK, gin.H{"filename": filename})
17 }
```



발표자

계획

준비

실행

종료

- Delete



```
1 // @BasePath /
2 // @Schemes http,https
3 // @Accept json
4 // @Produce json
5 // @Param filename query string true "The filename"
6 // @Success 200 {string} string "File deleted successfully"
7 // @Router /deleteFile [delete]
8 func deleteFile(c *gin.Context) {
9     filename := c.Query("filename")
10
11     os.Remove("file/" + filename)
12
13     // Return success message.
14     c.JSON(http.StatusOK, gin.H{"message": "File deleted successfully"})
15 }
```



발표자

계획

준비

실행

종료

- PUT



```
1 // @BasePath /
2 // @Schemes http,https
3 // @Accept json
4 // @Produce json
5 // @Param filename query string true "The filename"
6 // @Param newFilename query string true "The new filename"
7 // @Success 200 {string} string "File name updated successfully"
8 // @Router /updateFileName [put]
9 func updateFileName(c *gin.Context) {
10     filename := c.Query("filename")
11     newFilename := c.Query("newFilename")
12
13     // Check if the file exists.
14     if _, err := os.Stat("file/" + filename); os.IsNotExist(err) {
15         c.JSON(http.StatusNotFound, gin.H{"error": err.Error()})
16         return
17     }
18
19     // Rename the file.
20     os.Rename("file/"+filename, "file/"+newFilename)
21
22     // Return success message.
23     c.JSON(http.StatusOK, gin.H{"message": "File name updated successfully"})
24 }
25
```



발표자

계획

준비

실행

종료

아..... 거기까지만 생각만 했어야했어.....



발표자

계획

준비

실행

종료

- PUT

```
1 // @BasePath /
2 // @Schemes http,https
3 // @Accept json
4 // @Produce json
5 // @SecurityDefinitions.basic BasicAuth
6 // @Security "BasicAuth"
7 // @Router /protected [get]
8 func protected(c *gin.Context) {
9     // Get the username and password from the Authorization header.
10    username, password, _ := c.Request.BasicAuth()
11
12    // Check if the username and password are valid.
13    if username != "admin" || password != "secret" {
14        c.JSON(http.StatusUnauthorized, H{"error": "Invalid username or password"})
15        return
16    }
17
18    // Return a success message.
19    c.JSON(http.StatusOK, H{"message": "You are authorized"})
20 }
21
```



발표자

계획

준비

실행

종료

아 **FastAPI**가 착하고 또 착한거구나.....

아..... **pip**에 이어서 **Python**의 소중함을 느끼게된....

그렇다고 **Golang**를 안할생각은 없습니다
다만 언.젠.간 할게요 ㅎ

추후에 나중에 **FastAPI**처럼 프레임워크를
지원하는 프로젝트를 진행하고자 합니다

너무 꿈고 불편하네요



발표자

계획

준비

실행

종료

으로..... 끝날줄 알았죠?

라이브러리 안만들거예요 ㅎ

^^7



OpenAPI 간편해요?

Container App 배포기