

# NestJS アプリケーションから Swagger を自動生成する

2019/11/29 NestJS meetup Tokyo #1 @ Eureka

#nestjs\_meetup

# Who?

Yuta Suzuki (@euxn23)



- Engineer @ Japan Digital Design, Inc
- すき: ドリュウズ
- きらい: ウオノラゴン

## NestJS の概要振り返り

- TypeScript + DI で硬く実装できる
- NestJS の Opinionated な思想をベースに設計を共有しやすい
- スケールする Node.js アプリケーションで得に嬉しい

💣 スケールするアプリケーションはメンバーもスケールしがち 💣

## スケールに耐えられないドキュメンテーション

- 誰もメンテしないので実装と乖離していても気づかない
- 定義とサンプルが異なっていてどちらが正しいのか分からない
- モックに使おうと思っても使える状態になっていない

## なぜか

- swagger がアプリケーションと独立しているから
- swagger をメンテしている時間がないから
- swagger が正しいことを検証する仕組みがないから

NestJS の Swagger 生成で解決に近づけます

## 解説: Swagger とは

- API 定義のドキュメンテーション仕様
- 現在では OpenAPI 仕様に統合されている
- NestJS では実装やデコレータから Swagger 定義を追加できる





# Controller

```
@Controller()
export class AppController {
  constructor(private readonly appService: AppService) {}

  @Get('/users')
  @ApiResponse({ status: HttpStatus.OK, type: GetUsersResponse })
  async getUsers(
    @Query() { userIds }: GetUsersRequest,
  ): Promise<GetUsersResponse> {
    const users = await this.appService.getUsers(userIds);

    return { users };
  }
}
```

- `@ApiResponse()` で定義したステータスが swagger に吐かせる
- `@Query()` (POST の場合は `@Body()`) で定義した param が sagger に吐かれる
- [docs.nestjs.com/controllers#request-object](https://docs.nestjs.com/controllers#request-object)

# DTO

```
export abstract class GetUsersRequest {  
  @ApiModelProperty({ example: ['10', '11'] })  
  userIds?: string | string[];  
}  
  
export abstract class GetUsersResponse {  
  @ApiModelProperty({ example: usersStub })  
  users!: IUser[];  
}
```

- `@ApiModelProperty()` でパラメータの詳細を定義する
  - ここで `example` を設定しておくと、後述する `stub` の値になる

# Swagger Entry

```
export async function createSwaggerApp(): Promise<INestApplication> {  
  const app = await NestFactory.create(AppModule);  
  
  const options = new DocumentBuilder().build();  
  
  const document = SwaggerModule.createDocument(app, options);  
  
  SwaggerModule.setup('api', app, document);  
  
  return app;  
}  
  
export async function bootstrap(): Promise<void> {  
  const app = await createSwaggerApp();  
  await app.listen(8081);  
}  
  
bootstrap();
```

- `/api` でドキュメントページへ
- `/api-json` で swagger json 吐き出し

## swagger(v2) から openapi(v3) への変換

```
$ swagger2openapi http://localhost:8081/api-json >| src/swagger/swagger.json
```

- NestJS の吐き出す swagger は v2 である
- 容易に stub 化するために使用したい apisprout は openapi(v3) のみ対応
- swagger を openapi に変換してファイルに出力しておくとドキュメントにもなるので便利

## stub の起動

```
$ docker run -p 8082:8000 -v $PWD/src/swagger/swagger.json:/api.json danielgtaylor/apisprout /api.json
```

- apisprout の docker を使って stub 化
- CI 等でも簡単に stub 化できる
- gulp 等をうまく使うことで、test のフローにうまく組み込むことも可能

## まとめ

- NestJS を使うと実装と近い箇所にコードでドキュメントを書けるので便利
- 適切にドキュメントを書くことでそのまま Stub にもなるので便利
- 組織のスケールに際して腐らないドキュメンテーションを！