

**I rzekła Adnotacja - “Niech
stanie się dokumentacja!”**

Przemek Skwiercz

Kilka słów o mnie...

- ✦ Ojciec, programista i cyklista
- ✦ Zajmuje się tworzeniem oprogramowania od ponad 20 lat więc ze względu na wiek już na pewno senior ;-)
- ✦ Jeden z założycieli i organizatorów Bydgoszcz JUG
- ✦ W wolnym czasie miłośnik dobrej literatury, piwa kraftowego, teatru i starego punk rocka (niekoniecznie w tej kolejności)
- ✦ Java-backend developer w **deepsense.ai**
- ✦ Twitter **@pskwiercz**



Definicje

- ✦ **API** - **A**pplication **P**rogramming **I**nterface - ściśle określony zestaw reguł i ich opisów, w jaki programy komputerowe komunikują się między sobą
- ✦ **REST** - **R**epresentational **S**tate **T**ransfer
- ✦ **API Definition File** - plik zawierający opis wszystkich możliwych operacji jakie możemy wykonać przy użyciu tego API
- ✦ **OAS** - **O**pen **A**PI **S**pecification
- ✦ **Swagger** - zbiór narzędzi które generują/używają/operują na pliku OAS

REST request

Method

URL

**Query
parameters**

POST **http://api.com/user?device=ipad**

Schema

Host

Path

(parameters)

Accept: application/json

Content-type: application/json

Headers

{

name: "Peter",

age: 20

Body

}

Open API

- ✦ **OpenAPI Specification** jest specyfikacją opisującą zawartość pliku w **YAML** lub **JSON**, który to plik opisuje interfejs REST naszej aplikacji
- ✦ Plik ten jest używany do opisu, wysyłania, pobierania i wizualizacji serwisów REST'owych
- ✦ Początkowo była to część **framework'a Swagger** ale w 2016 roku stała się osobnym projektem przejętym przez inicjatywę OpenAPI (pod patronatem **Linux Foundation**)

Przykładowy plik OAS

```
1. paths:
2.   /users:
3.     post:
4.       summary: Adds a new user
5.       requestBody:
6.         content:
7.           application/json:
8.             schema: # Request body contents
9.               type: object
10.            properties:
11.              id:
12.                type: integer
13.              name:
14.                type: string
15.            example: # Sample object
16.              id: 10
17.              name: Jessica Smith
18.          responses:
19.            '200':
20.              description: OK
```


Swagger Annotations versions

- ✦ Swagger Annotations version **1.5.x** generates file in OpenAPI Specification version **2.0**
- ✦ Swagger Annotations version **2.0.x** generates file in OpenAPI Specification version **3.0**

Swagger - moduł

- ✦ Swagger Editor
- ✦ Swagger Code Generator
- ✦ Swagger UI
- ✦ Swagger HUB



Swagger Security

Swagger wspiera następujące rodzaje zabezpieczeń:

- ✦ None
- ✦ Basic Auth
- ✦ API Key
- ✦ OAuth

Swagger Annotations

Typy adnotacji:

- Endpoints Annotations
 - **@Api, @ApiOperation, @ApiResponses, @ApiResponse, @ApiParam**
- Model Annotations
 - **@ApiModel, @ApiModelProperty**
- Configuration
 - **@EnableSwagger2**

Linki

- ✦ Swagger

<https://swagger.io>

- ✦ Swagger Annotations

<https://github.com/swagger-api/swagger-core/wiki/Annotations-1.5.X>

















- ✦ Open API











<https://www.openapis.orggithub.com>

- ✦ GitHub repository

https://github.com/pskwiercz/JUG_Swagger_Annotations

Aplikacja - klasy

 	Book	
 	id	long
 	author	String
 	title	String
 	pages	int
 	discount	boolean
 	<i>Book(long, String, String, int, boolean)</i>	
 	<i>Book()</i>	

 	BookControllerImpl	
 	bookRepository	BookRepository
 	getAllBooks()	List<Book>
 	getBookById(Long)	ResponseEntity<Book>
 	createBook(Book)	Book

Aplikacja - endpoints

- ✦ **GET** <http://localhost:8080/api/book/>
- ✦ **GET** <http://localhost:8080/api/book/{id}>
- ✦ **POST** <http://localhost:8080/api/book/>

Plan działania

- ✦ Uruchomić aplikację opartą na SpringBoot z trzema REST'owymi endpointami
- ✦ Dodać Swagger'a do naszej aplikacji (Maven dependencies)
- ✦ Sprawdzić wygenerowaną przez Swagger'a dokumentację i przy jej użyciu przetestować naszą aplikację
- ✦ Dodać dodatkowe informacje o modelu do dokumentacji
- ✦ Dodać szczegółowe informacje o endpointach do dokumentacji
- ✦ **Udać się na afterparty :-)**

***“Koniec pierdół czas na
mięso!”***

*– Sebastian O. w trakcie wygrzebywania
ostatniego kawałka zieleniny z kebaba*

Q & A

Dzieki!