



OPENAPI INITIATIVE

Live de Python # 178



1. OpenAPI

Como funciona o padrão?

2. Ferramentas

Swagger e seus amigos

3. Conversão

Convertendo especificação em código

4. Ecossistema

Outras ferramentas que incrementam o ecossistema



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Acássio Anjos, Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysso Oliveira, Amaziles Carvalho, Andre Azevedo, André Rocha, Antonio Lins, Arnaldo Turque, Artur Zalewska, Bruno Barcellos, Bruno Batista, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Diego Ubirajara, Dilenon Delfino, Dino Aguilar, Donivaldo Sarzi, Elias Soares, Emerson Rafael, Eric Niens, Eugenio Mazzini, Euripedes Borges, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Flavkaze Flavkaze, Flávio Meira, Francisco Alencar, Franklin Silva, Fábio Barros, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Suto, Henrique Junqueira, Henrique Machado, Ismael Ventura, Israel Fabiano, Israel Gomes, Italo Silva, Jair Andrade, Jairo Rocha, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Gomes, José Prado, João Lugão, Juan Gutierrez, Jônatas Silva, Kaio Peixoto, Kaneson Alves, Leonardo Cruz, Leonardo Mello, Leonardo Nascimento, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Luciano Silva, Maiquel Leonel, Marcela Campos, Marcelino Pinheiro, Marcos Ferreira, Maria Clara, Marina Passos, Matheus Vian, Murilo Cunha, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Brito, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Richard Nixon, Riverfount Riverfount, Robson Maciel, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Rodrigo Vieira, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sandro Mio, Sara Selis, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Moraes, Tony Dias, Tony Santos, Tyrone Damasceno, Vinícius Bastos, Vlademir Souza, Vítor Gomes, Wellington Abreu, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Yuri Fialho, Yury Barros, Érico Andrei



Obrigado você



Documentação de
api? Que raios é
isso

Open
API



A OAS é uma forma agnóstica de linguagem de descrever as interfaces de uma API HTTP.

É uma maneira de dizer as pessoas e máquinas como usar sua API.

<https://spec.openapis.org/oas/latest.html#openapi-specification>

Uma palinha

Só para matarmos a curiosidade.

A especificação pode ser feita com YAML (da imagem) ou Json

```
openapi: 3.0.0

info:
  version: 0.0.1
  title: Todo API

paths:
  /todo:
    get:
      summary: Pega todas as tarefas
      responses:
        200:
          description: Retorna todas as tarefas
    post:
      summary: Cria uma nova tarefa
      description: Cria uma nova tarefa, baseada em xpto
      responses:
        201:
          description: Tarefa criada com sucesso
        422:
          description: Erro no payload
```

Entendendo a especificação (OAS)



Basicamente a especificação é dividida em um esquema de campos "fixos" e objetos

Porém, somente dois campos são requeridos

Campo	Para que serve?
openapi	Especifica a versão da OAS em uso
info	Informações sobre a sua API

Entendendo a especificação (OAS)



Campos mais utilizados

Campo	Para que serve?
openapi	Especifica a versão da OAS em uso
info	Informações sobre a sua API
components	Objetos reutilizáveis como jsons de payload ou resposta
paths	Caminhos dos endpoints da sua API
servers	URLs dos servidores que servem a API documentada

Entendendo a especificação (OAS)



Embora existam outros campos como:

- webhooks
- security
-

A estrutura do YAML é basicamente essa.

```
openapi: 3.0.0
```

```
info:
```

```
servers:
```

```
components:
```

```
paths:
```

Quando a coisa
começa a ficar
divertida

Objetos

Nosso projeto



Vamos imaginar uma pequena API de tarefas a fazer (um todo list).

O que precisamos para fazer isso?

- Listar tarefas
- Criar tarefas
- Alterar tarefas
- Deletar tarefas

Objetos do OAS



Da mesma forma que temos os "campos" na base do nosso arquivo YAML, agora começaremos a descrever os dados e metadados da nossa API.

Vamos pensar na criação de uma tarefa. Precisamos de um endpoint, do dado que vamos enviar e o que vamos receber.

Paths

Para definir recursos, rotas, endpoints, paths, como você preferir chamar. Usamos a chave **path** do nosso arquivo de especificação. Dentro dele vamos inserir o nosso objeto path.

```
paths:
  /todo: # Nosso endpoint
    post: # Verbo HTTP utilizado
          # Post geralmente é usado para "criar" algo

    description: Cria uma nova tarefa a fazer
    responses: # Códigos de resposta que pode ser gerado
      201:
        description: Tarefa criada com sucesso
      422:
        description: Erro no payload
```

Objeto Info

Agora que temos um endpoint com post e sabemos o que queremos fazer, podemos também documentar o objeto info

```
openapi: 3.0.0
```

```
info:
```

```
  version: 0.0.1
```

```
  title: Tarefas a fazer
```

```
paths:
```

```
  /todo:
```

```
    post:
```

```
      description: Cria uma nova tarefa a fazer
```

```
      responses:
```

```
        201:
```

```
          description: Tarefa criada com sucesso
```

```
        422:
```

```
          description: Erro no payload
```

<https://editor.swagger.io/>



Bora tornar esse documentação mais dinâmica?



<https://spec.openapis.org/>



O site mais importante



Conhecendo!

swa
gger

Swagger



Swagger é geralmente confundido com OpenAPI, pois a equipe do swagger foi quem iniciou as discussões sobre openAPI e um padrão "comum" e agnóstico para criação de APIs.

Basicamente o swagger oferece um conjunto de ferramentas que fazem uso do OpenAPI

Ferramentas



O swagger pode ser dividido em 3 grandes ferramentas

- Editor (já usamos)
- UI (também já usamos)
- Code generator

code
gen

Gerando código
usando openAPI

Data model



É possível gerar um modelo de dados usando os schemas do OpenAPI e os convertendo para o pydantic (live de python # 165)

```
pip install datamodel-code-generator
```

Exemplo de uso



```
datamodel-codegen --input api.yaml --output model.py
```



Também é possível gerar uma API completa usando o gerador do FastAPI

```
pip install fastapi-code-generator
```


Exemplo de uso



```
fastapi-codegen --input api.yaml --output app
```

Mais
ferramentas?

ecossi
stema

Python + OpenAPI?



- Django
 - django-openapi
 - Esquema de rotas + documentação
 - drf
 - drf-swagger
 - drf-yasg
- Flask
 - flasgger
 - flask-pydantic-spec

Python + OpenAPI?



- FastAPI
 - Suporte nativo
- APISpec
 - Uso genérico, usa marshmallow
- Clientes
 - openapi-python-client

Mais ferramentas



Uma outra ferramenta que eu gosto BASTANTE é o ReDoc, pois ela gera documentações estáticas.

```
yarn add redoc-cli  
yarn run redoc-cli -o doc.html spec.yml
```

Mais ferramentas podem ser acessadas aqui: **<https://openapi.tools/>**



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

