

Aviso legal

Este trabalho é licenciado sob a Creative Commons Atribuição-NãoComercial-SemDerivações 4.0 Internacional (CC BY-NC-ND 4.0). Para ver uma cópia desta licença, por favor visite a página <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt>



Serviços para Aplicações Web e Móveis

Swagger UI

Fábio Marques (fabio@ua.pt)
Mestrado em Informática Aplicada



Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

Introdução

Swagger

Swagger UI

A documentação da API tem um papel crítico tanto para os programadores que a desenvolvem como para os que a consomem.

- **Desenvolvedores da API:** A documentação é uma ferramenta de comunicação que permite a transmissão de informação aos programadores que consomem a API. Fornece orientações claras utilizar a API, incluindo endpoints, métodos, formatos de pedidos e respostas, etc.
- **Consumidores da API:** Os programadores que consomem API dependem muito da documentação para saber como interagir com a API.

Uma documentação abrangente poupa tempo e esforço ao fornecer instruções e exemplos claros sobre como fazer pedidos, interpretar respostas, tratar erros e aproveitar os vários recursos da API. Uma boa documentação pode acelerar significativamente o processo de desenvolvimento.

Introdução

Swagger

Swagger UI

- é uma ferramenta de código aberto que ajuda a documentar, projetar e consumir serviços da Web RESTful.
- simplifica o desenvolvimento de API, da sua documentação e o seu teste
- é um conjunto de ferramentas relacionadas com a especificação **OpenAPI**¹
- consiste em três componentes principais:
 - **Swagger Editor**: Uma ferramenta de edição que permite escrever especificações OpenAPI.
 - **Swagger UI**: Uma interface de utilizador que disponibiliza a documentação da API.
 - **Swagger Codegen**: Uma ferramenta que gera código de cliente, servidor e documentação a partir de uma especificação OpenAPI.

¹Especificação que fornece um conjunto de propriedades para descrever uma API REST.

- **Padronização:** Assegura a consistência e adesão às melhores práticas do desenho de API.
- **Documentação:** Gera automaticamente a documentação da API a partir da especificação OpenAPI, mantendo-a sincronizada com as alterações ao código.
- **Interatividade:** Permite testar a API diretamente a partir da documentação.
- **Geração de clientes:** Simplifica a geração de código para várias linguagens de programação.
- **Testes:** Facilita o teste da API com ferramentas e integrações incorporadas.

- **Nomenclatura consistente:** Utilizar convenções de nomenclatura consistentes para caminhos, parâmetros e campos de resposta.
- **Controle de versão:** Incluir a versão da API no URL para suportar a compatibilidade com versões anteriores e facilitar a gestão de versões.
- **Utilizar componentes reutilizáveis:** Utilizar referências para reutilizar definições de esquema, parâmetros e respostas.
- **Manter a simplicidade:** Evitar complexidade desnecessária e manter a documentação simples e fácil de entender.

- **Documentar minuciosamente:** Incluir descrições detalhadas para cada caminho, parâmetro, resposta e código de estado.
- **Controlo de versões:** Utilizar o controlo de versões para gerir alterações na documentação e colaborar com a equipa de forma eficaz
- **Testar a especificação:** Utilizar ferramentas como Swagger UI ou Postman para validar a especificação da API e garantir que se comporta como esperado.

Introdução

Swagger

Swagger UI

Instalação

```
npm i swagger-ui-express  
npm i js-yaml
```

Utilização do swagger

```
const swaggerUI = require('swagger-ui-express');  
var fs = require('fs');  
var jsyaml = require('js-yaml');  
var spec = fs.readFileSync('swagger.yaml', 'utf8');  
swaggerDocument = jsyaml.load(spec);  
  
app.use('/api-docs', swaggerUI.serve, swaggerUI.setup(swaggerDocument));
```

Campo	Tipo	Descrição
<code>openapi*</code>	string	Número da versão da especificação
<code>info*</code>	Objeto Info	Metadados com informação sobre a API
<code>paths</code>	Objeto Paths	Caminhos e operações que estão disponíveis na API
<code>security</code>	Objeto Security Requirement	Declaração de quais os mecanismos que podem ser utilizados na API

```
openapi: "3.1.0" // latest version
info: ...
paths: ...
```

Campo	Tipo	Descrição
title*	string	O título da API
summary	string	Um pequeno resumo da API
description	string	A descrição da API
termsOf Service	string	Um URL para os termos de serviço da API
contact	Objeto Contact	A informação do contacto da API
license	Objeto License	A informação sobre a licença
version*	string	A versão do documento OpenAPI

```
info:  
  title: "Tasks Application"  
  description: "Small academic example"  
  version: "1.0.0"
```

Paths

Campo	Tipo	Descrição
/path*	Objeto Path Item	Um caminho relativo para um endpoint individual

Path Item

Campo	Tipo	Descrição
\$ref	string	Permite a referência para a definição deste item
summary	string	Um pequeno resumo que se aplica a todas as operações deste caminho
description	string	Descrição que se aplica a todas as operações deste caminho
get ... trace	Objeto Operation	A definição da operação aplicada a este caminho

Campo	Tipo	Descrição
summary	string	Um pequeno resumo do que a operação realiza
description	string	Descrição do comportamento da operação
requestBody	Objeto Request Body	O corpo do pedido que é aplicável a esta operação
responses	Objeto Responses	Lista de possíveis respostas que podem ser devolvidas como resultado da execução da operação

```
/tasks:
  get:
    description: Returns all tasks from the system that the user has access to
    responses:
      '200':
        description: A list of tasks.
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                properties:
                  id:
                    type: number
                ...
```


Campo	Tipo	Descrição
description	string	Breve descrição do corpo do pedido
content*	Map[string, Objeto Media Type]	O conteúdo do corpo do pedido
required	boolean	Indica se o corpo do pedido é obrigatório. Falso por omissão

```
description: task to add to the system
content:
  'application/json':
    schema:
      $ref: '#/components/schemas/Task'
    examples:
      task:
        summary: Task Example
        externalValue: 'https://foo.bar/examples/task-example.json'
```

Campo	Tipo	Descrição
default	Response Object Reference Object	Documenta as respostas que não são contempladas nas respostas para os códigos HTTP
HTTP Status Code	Response Object Reference Object	Descreve a resposta esperada para o código de estado HTTP

```
'200':  
  description: a task to be returned  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/Task'  
default:  
  description: Unexpected error  
  content:  
    application/json:  
      schema:  
        $ref: '#/components/schemas/ErrorModel'
```

Campo	Tipo	Descrição
<code>description*</code>	string	Descrição da resposta
<code>content</code>	Map[string, Objeto Media Type]	Contém a descrição da resposta

```
description: A complex object array response
content:
  application/json:
    schema:
      type: array
      items:
        $ref: '#/components/schemas/VeryComplexType'
```

Editor Swagger

Documentação Swagger

Especificação OpenAPI