




# Documentação da API

■ Course	 <b>Spring Boot 3</b>
☀ Confidence	Somewhat Confident
📅 Next Review	@9 de agosto de 2023
🕒 Last Edited	@8 de agosto de 2023 10:34
🔄 Curso	Spring Boot 3: Documente teste e prepare uma API para o deploy

## Tópicos

- Documentando com SpringDoc
- Para saber mais: OpenAPI Initiative
- JWT na documentação
- Para saber mais: personalizando a documentação
- Vantagens da documentação
- Faça como eu fiz: documentação da API
- O que aprendemos?

## Para saber mais: OpenAPI Initiative

A documentação é algo muito importante em um projeto, principalmente se ele for uma API Rest, pois nesse caso podemos ter vários clientes que vão precisar se comunicar com ela, necessitando então de uma documentação que os ensinem como realizar essa comunicação de maneira correta.

Por muito tempo não existia um formato padrão de se documentar uma API Rest, até que em 2010 surgiu um projeto conhecido como **Swagger**, cujo objetivo era ser uma especificação *open source* para design de APIs Rest. Depois de um tempo, foram desenvolvidas algumas ferramentas para auxiliar pessoas desenvolvedoras a implementar, visualizar e testar suas APIs, como o *Swagger UI*, *Swagger Editor* e *Swagger Codegen*, tornando-se assim muito popular e utilizado ao redor do mundo.

Em 2015, o Swagger foi comprado pela empresa **SmartBear Software**, que doou a parte da especificação para a fundação Linux. Por sua vez, a fundação renomeou o projeto para **OpenAPI**. Após isso, foi criada a **OpenAPI Initiative**, uma organização focada no desenvolvimento e evolução da especificação OpenAPI de maneira aberta e transparente.

A OpenAPI é hoje a especificação mais utilizada, e também a principal, para documentar uma API Rest. A documentação segue um padrão que pode ser descrito no formato yaml ou JSON, facilitando a criação de ferramentas que consigam ler tais arquivos e automatizar a criação de documentações, bem como a geração de códigos para consumo de uma API.

Você pode obter mais detalhes no [site oficial da OpenAPI Initiative](#).

## Para saber mais: personalizando a documentação

Vimos no vídeo anterior que é possível personalizar a documentação gerada pelo SpringDoc para a inclusão do token de autenticação. Além do token, podemos incluir outras informações na documentação que fazem parte da especificação OpenAPI, como, por exemplo, a descrição da API, informações de contato e de sua licença de uso.

Tais configurações devem ser feitas no objeto `OpenAPI`, que foi configurado na classe `SpringDocConfigurations` de nosso projeto:

```
@Bean
public OpenAPI customOpenAPI() {
```

```

return new OpenAPI()
    .components(new Components()
        .addSecuritySchemes("bearer-key",
            new SecurityScheme()
                .type(SecurityScheme.Type.HTTP)
                .scheme("bearer")
                .bearerFormat("JWT")))
    .info(new Info()
        .title("Voll.med API")
        .description("API Rest da aplicação Voll.med, contendo as funcionalidades de CRUD de médicos e de paciente")
        .contact(new Contact()
            .name("Time Backend")
            .email("backend@voll.med")))
    .license(new License()
        .name("Apache 2.0")
        .url("http://voll.med/api/licenca"));
}

```

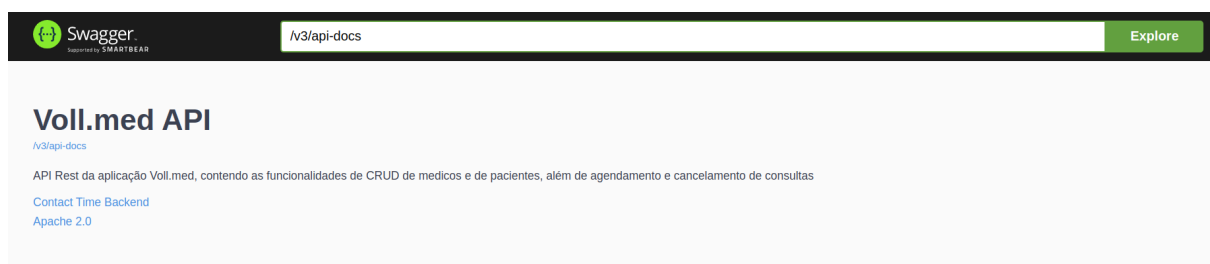
Usando os imports:

```

import io.swagger.v3.oas.models.info.Contact;
import io.swagger.v3.oas.models.info.Info;
import io.swagger.v3.oas.models.Components;
import io.swagger.v3.oas.models.OpenAPI;
import io.swagger.v3.oas.models.info.License;
import io.swagger.v3.oas.models.security.SecurityScheme;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

```

No código anterior, repare que após a configuração do token JWT foram adicionadas as informações da API. Ao entrar novamente na página do Swagger UI, tais informações serão exibidas, conforme demonstrado na imagem a seguir:



Para saber mais detalhes sobre quais informações podem ser configuradas na documentação da API, consulte a [especificação OpenAPI](#) no site oficial da iniciativa.

## Vantagens da documentação

Ao longo desta aula, aprendemos a utilizar a biblioteca SpringDoc para a geração da documentação da nossa API.

Considerando o que foi ensinado, escolha as alternativas que indicam as vantagens de se gerar uma documentação automatizada de uma API Rest.

- Padronização do código da API.
  - A geração da documentação não padroniza automaticamente o código de uma API.
- Possibilidade de testar a API.
  - A documentação gerada seguindo o padrão OpenAPI permite que ferramentas, como o Swagger UI, possam ser utilizadas para realizar testes na API.
- Facilidade para clientes se integrarem a ela.

- A documentação facilita muito a vida de quem deseja se integrar à uma API Rest, pois expõe as suas funcionalidades e como elas devem ser consumidas.
  - Melhorar a produtividade do desenvolvimento.
    - Não necessariamente a produtividade vai aumentar com a geração da documentação.
- 

## O que aprendemos?

### Nesta aula, você aprendeu como:

- Adicionar a **biblioteca SpringDoc** no projeto para que ela faça a geração automatizada da documentação da API;
  - Analisar a documentação do SpringDoc para entender como realizar a sua configuração em um projeto;
  - Acessar os endereços que disponibilizam a documentação da API nos formatos yaml e html;
  - Utilizar o **Swagger UI** para visualizar e testar uma API Rest;
  - Configurar o JWT na documentação gerada pelo SpringDoc.
-