



Enviando token JWT no Swagger

Transcrição

[00:00] Nós já temos configurada a documentação pelo Swagger. Vimos que ele listou todos os controllers, os endpoints. Testamos alguns. Mas não conseguimos fazer a exclusão, porque tenho o campo para digitar o id, porém tomamos um 403, porque esse endpoint precisa de autenticação. Nós até geramos o token, só que não tem um lugar para informar esse token. Nessa aula vamos configurar um campo para conseguir digitar o cabeçalho do authorization e digitar o token.

[00:45] Voltando ao Eclipse. Como essa é uma configuração do Swagger, ela fica na classe Swagger configurações. Dentro dessa classe, nós tínhamos configurado qual o pacote base, qual a url raiz, e para ele ignorar a classe usuário. Logo na linha que estamos ignorando a classe usuário temos que adicionar mais uma para configurar um parâmetro global.

[01:10] Eu vou colar o código, porque é meio chato. A ideia é que existe o método chamado globalOperationParameters, em que conseguimos adicionar parâmetros globais. Ou seja, é um parâmetro que quero que o Swagger apresente em todos os endpoints. Esse método recebe uma lista com parâmetros. Eu fiz a chamada para o arrays.asList, mas no caso só estamos passando um único parâmetro. Para passar o parâmetro, precisamos dar new nessa classe, que é um builder, onde vamos construir como vai ser esse parâmetro.

[01:54] Eu setei o nome do parâmetro, que é authorization, coloquei uma description para aparecer na documentação, e falei que é o header para envi

o token para o JSON web token. Depois, tenho que dizer qual é o tipo do parâmetro, que no caso é uma string. Qual o tipo de parâmetro, no nosso caso é um cabeçalho. O parâmetro é opcional, porque tem endereços que não precisam dele, e o builder para construir o objeto para nós. Feito isso é só salvar, ele vai reiniciar o projeto.

[02:36] Se eu for até o site do Swagger, ele carregou normalmente. Vamos simular aquele fluxo. Vou até o AutenticacaoController, na requisição POST, clicar no try it out, preencher com os dados, mandar executar. Ele devolveu o token. Vou copiar o token, minimizar o AutenticacaoController, expandir o tópicos controller e aquele método DELETE para fazer o teste. Já apareceu além do campo do id o authorization. Ele até coloca entre parênteses que é um header.

[03:28] Se eu clicar agora no try it out, além de ter o campo para fornecer o id da url, tem um campo para passar o cabeçalho authorization. Tenho que colocar o bearer, que é o tipo, e vou colar o token gerado. Agora, se eu executar, dá código 200. Ele executou com sucesso.

[03:53] Vamos fazer um teste. Vou passar um token inválido e mandar executar. Ele me dá um 403. Se eu não levar o cabeçalho e executar também dá 403. Está funcionando. Já tínhamos implementado o esquema de pegar o token do cabeçalho e autenticar o cliente na requisição.

[04:14] Com isso, consigo também fazer testes na API, pelo Swagger, passando o cabeçalho de autenticação. Mesmo os endpoints que precisam de autenticação consigo fazer o teste. Isso fica disponível para os clientes que forem ler a documentação.

[04:43] Com isso, terminamos esse capítulo de documentação. Conforme vocês viram, existe o Swagger, que é a ferramenta para fazer a documentação, e o Spring fox Swagger, que é a biblioteca para documentar uma API Java com Spring, utilizando o Swagger. A documentação fica por esse site. Tem um layout bacana, ele identifica nossos controllers, tem toda a documentação dos

endpoints, quais os parâmetros, o que ele devolve, e dá até para fazer testes direto pela interface. É uma ferramenta bem interessante e importante para quem está construindo uma API REST.

[05:23] Espero que vocês tenham gostado. Vejo vocês no próximo vídeo.