



Habilitando o Spring Security

Transcrição

[00:00] Na última aula, configuramos a parte de cache, para dar uma melhorada na performance. Na aula de hoje, vamos aprender um pouco sobre segurança, porque até então temos nossa API funcionando, mas não tem segurança nenhuma. Qualquer pessoa que souber os endereços dos endpoint consegue disparar requisições para ler os tópicos, cadastrar, excluir, alterar, fazer qualquer coisa.

[00:30] No geral, em uma aplicação costumamos ter um controle de acesso. Quero restringir quem pode disparar requisições para a API, quem tem autorização para acessar determinados recursos. Nesta aula, vamos aprender sobre essa parte de segurança.

[00:45] O Spring Boot possui um módulo focado somente nessa parte de segurança, que cuida da parte de autenticação e autorização, que é o Spring security. O primeiro passo para começarmos a proteger nossa API é baixar o módulo do Spring security e adicionar como uma dependência do nosso projeto.

[01:04] Vou abrir nosso pom.xml. Vamos ter que adicionar mais uma dependência. Vou pegar a do cache, copiar, colar, e trocar o final para Spring-Boot-starter-security, que é o módulo de segurança do Spring Boot. Ele vai baixar as dependências.

[01:28] Como funciona a parte de segurança? Só preciso da dependência. O próximo passo seria configurar. Lembra que em um projeto com Spring Boot fazemos as configurações no application.properties. Só que como essa parte de

segurança é um pouco complicada, temos muitas configurações, muita coisa dinâmica, não vamos utilizar o `application.properties`. Essa parte de segurança fazemos toda via classe Java.

[02:07] Eu vou precisar criar uma classe onde todas as configurações de segurança estarão. Vou criar uma nova classe, colocar no pacote `security`, criado dentro do pacote `config`, só para colocarmos coisas relacionadas a segurança dentro desse pacote. E vou criar uma classe chamada, por exemplo, `securityConfigurations`. A ideia é que dentro dessa classe estarão todas as configurações de segurança do nosso projeto.

[02:43] É uma classe Java, não tem nada a ver com Spring. Tenho que habilitar a parte do Spring security. Para fazer isso, fazemos na própria classe. Existe uma anotação chamada `@EnableWebSecurity`. Como essa é uma classe que tem configurações, precisamos colocar a anotação `@Configuration`. O Spring vai carregar e ler as configurações que estiverem dentro dessa classe.

[03:20] Além disso, vamos ter que herdar essa classe de outra classe do Spring chamada `web security configurer adapter`. Essa classe tem alguns métodos para fazer as configurações que vamos sobrescrever posteriormente.

[03:38] É isso. Nós colocamos a dependência do Spring security no projeto, criamos a classe, anotada com `@EnableWebSecurity`, com `@Configuration`. Dentro, depois, vamos colocar as configurações de segurança. Por enquanto está vazio, mas só de ter feito isso já habilitamos a parte de segurança. Por padrão, o Spring bloqueia todo acesso à nossa API. Tudo está restrito até que eu faça a configuração e libere o que precisa ser liberado.

[04:05] Já podemos testar. No Postman, vou tentar disparar uma requisição para aquele nosso endpoint/tópicos, que era o endpoint que trazia uma lista com todos os tópicos do projeto. Ele não voltou a lista. Devolveu o código 401. Não tenho autorização. Ou seja, o Spring security está habilitado e o padrão dele é bloquear tudo. Com isso, terminamos nessa aula. No próximo vídeo

vamos começar a fazer as configurações, para ensinar ao Spring o que é público e o que não é.