



Monitoramento com Spring Boot Actuator

Transcrição

[00:00] Na última aula nós terminamos a parte de segurança utilizando token. Na aula de hoje vamos aprender um pouco sobre monitoramento. Se você tem uma API REST que tem dezenas, centenas de clientes é importante ter um monitoramento, ainda mais se você estiver trabalhando com arquitetura de micro serviços, em que você tem vários serviços que você tem que gerenciar.

[00:40] Pensando nisso, o pessoal do Spring Boot criou um projeto chamado actuator. É uma API que tem endpoints que devolvem um JSON com informações sobre a API. É uma facilidade para conseguir monitorar e acompanhar o andamento da nossa API.

[01:12] Como ele é um módulo, vamos precisar adicioná-lo ao nosso projeto. Precisamos adicionar a dependência do Spring Boot actuator. Vou salvar. O Maven vai baixar as dependências. Só de ter o actuator como dependência do projeto, ele já vem habilitado por padrão.

[01:45] Vou rodar nossa aplicação. Por padrão, ele vai criar um novo endpoint na nossa aplicação, que é o /actuator. Se dispararmos uma requisição GET para /actuator, ele vai devolver um JSON com algumas informações sobre a aplicação. Vou testar pelo Postman mesmo. Ele me deu um 403, porque como está com Spring security, é um novo endereço, precisamos liberar o acesso naquele método onde configuramos as URLs.

[02:54] Para testes, vou colocar um permitAll, mas quando você for colocar sua API em produção, a ideia é que você remova isso, porque esse endpoint devo informações sensíveis sobre a aplicação, você não quer deixar isso aberto para

qualquer pessoa. Vai ser só para sua equipe, a equipe de infraestrutura da sua empresa.

[03:25] Vou disparar de novo a requisição. Ele me dá um 200 e devolve um JSON com alguns links, que são sub-recursos de coisas que você pode acessar do actuator. Por exemplo, o self, o health, para saber informações sobre a saúde da sua API, o info, que devolve informações sobre a API.

[04:07] Vamos disparar uma requisição para o /health. Deu proibido, porque esse é um sub-endereço. Vou voltar no configuration e colocar /actuator/**. Salvo e testo de novo. A única coisa que o health me devolve é um “up”, indicando que a aplicação está de pé.

[04:47] Vamos testar o /info. Vem vazio, porque por padrão ele não devolve nada. Conseguimos configurar no Spring para ele devolver informações da API e para expor outros endpoints. Vamos fazer isso. Como é uma configuração, fica no application.properties. Eu vou colar algumas propriedades. Você não precisa decorar isso. No exercício você vai pegar esse arquivo pronto. Mas só para entender, na primeira linha estou dizendo para o health sempre mostrar todos os detalhes, não só o up. Na segunda, quero que ele inclua outros endpoints com outras informações sobre a nossa API. E as duas últimas servem para exibir informações sobre nossa aplicação.

[06:15] No valor, passamos a sintaxe @project.name@. Isso é para falar para o Spring Boot que é para puxar as informações do arquivo .xml. No .xml, no topo do arquivo, tem as informações de groupId, artifactId, etc. Conseguimos falar para ele puxar todas essas informações e jogar no actuator.

[06:52] Vou salvar, ele vai reiniciar. Vou disparar de novo a requisição para o /info. Agora veio com as informações que coloquei para ele expor no application properties. Vamos disparar a requisição para o health. Ao invés de devolver só o status up ou down, ele devolve mais alguns detalhes. São informações que conseguimos descobrir, habilitar, para expor no endpoint do actuator.

[08:04] Tudo que ele devolve é no formato JSON. O actuator não tem interface gráfica, não mostra uma tela, um dashboard com gráficos e coisas do gênero. Ele só vai devolver o JSON com essas informações. Então, na verdade, a equipe de desenvolvimento da API só vai ativar o actuator, configurar o que queremos e não queremos expor, e passar o endereço para o pessoal que vai cuidar da parte de infraestrutura. Esse pessoal já tem ferramentas que vão disparar requisições para esse endereço, vão puxar o JSON, montar um gráfico, algo do gênero.

[08:52] No geral, na área de TI você tem o pessoal responsável pela parte de operações. Eles é que vão consumir o endpoint para fazer o monitoramento, configurar alertas, e-mails e coisas do gênero. A única coisa que nós fazemos é liberar o actuator para expor as informações da nossa API. Nós não precisamos implementar essas informações na mão. O próprio actuator já tem a lógica que expõe as informações automaticamente.

[09:20] Esse é o actuator. É uma maneira de você expor informações da sua API para que alguém consiga puxar esses dados e fazer o monitoramento. No próximo vídeo vou mostrar uma ferramenta que podemos utilizar para consumir essas informações e exibir uma tela com gráficos, de maneira visual, caso sua equipe de operações não saiba como consumir. Ou você mesmo pode usar para fazer o monitoramento, ter uma interface gráfica de administração dessas informações.