Certified Tech Developer
The Ultimate Degree

Swagger

As APIs nos permitem conectar e compartilhar dados entre sistemas diferentes, mas se quisermos nos conectar com APIs externas, como sabemos quais dados são necessários para receber, que tipo de solicitação enviar ou para qual endereço?

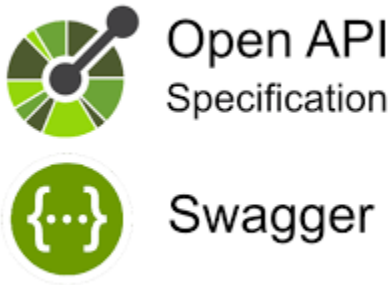
De alguma forma, temos que documentar como usar as APIs que desenvolvemos, se só nós sabemos como nos conectar, não faria sentido criar uma API. **Swagger** nos ajudará a criar a documentação de nossas APIs, para que todos entendam como usá-la.

Mas o que é Swagger

Swagger é um conjunto de ferramentas de código aberto construído em torno da especificação OpenAPI, que nos ajudará a projetar, construir, documentar e consumir APIs REST.

O que é Open Api?

OpenAPI é um padrão para a descrição de APIs. A especificação OpenAPI define o formato no qual elas devem ser descritas, desenvolvidas, testadas e documentadas. Em outras palavras, graças a este padrão, a documentação para a maioria das APIs terá o mesmo formato.



Como funciona a documentação de APIs com Swagger e Spring Boot?

O Swagger gera a documentação automaticamente a partir de nossos controllers, criando uma interface web onde serão listados todos os endpoints da nossa aplicação, compreendendo os dados que eles recebem e também nos permite enviar solicitação (request) desta mesma tela para nossa aplicação. Em nosso caso, usaremos o Swagger por meio do Spring Boot, utilizando uma biblioteca chamada **SpringDoc**. Se quiser conhecer mais sobre o assunto, acesse a documentação oficial.

Conclusão

Concluindo, podemos mencionar que o Swagger realiza de forma automática um trabalho que certamente deixamos para ser realizado no último minuto e que ninguém gostaria de fazer.

Documentar nossas APIs aumentará a velocidade de desenvolvimento em equipes que estão divididas em front-end e back-end. Por outro lado, se desenvolvermos APIs que serão utilizadas em qualquer lugar do mundo, será fundamental ter uma boa documentação para facilitar a comunicação e focar no que realmente importa.

A seguir, sugerimos que você assista a live coding onde exemplificamos o que vimos anteriormente e realizar um exercício para fixar o que aprendemos.

Vamos praticar!

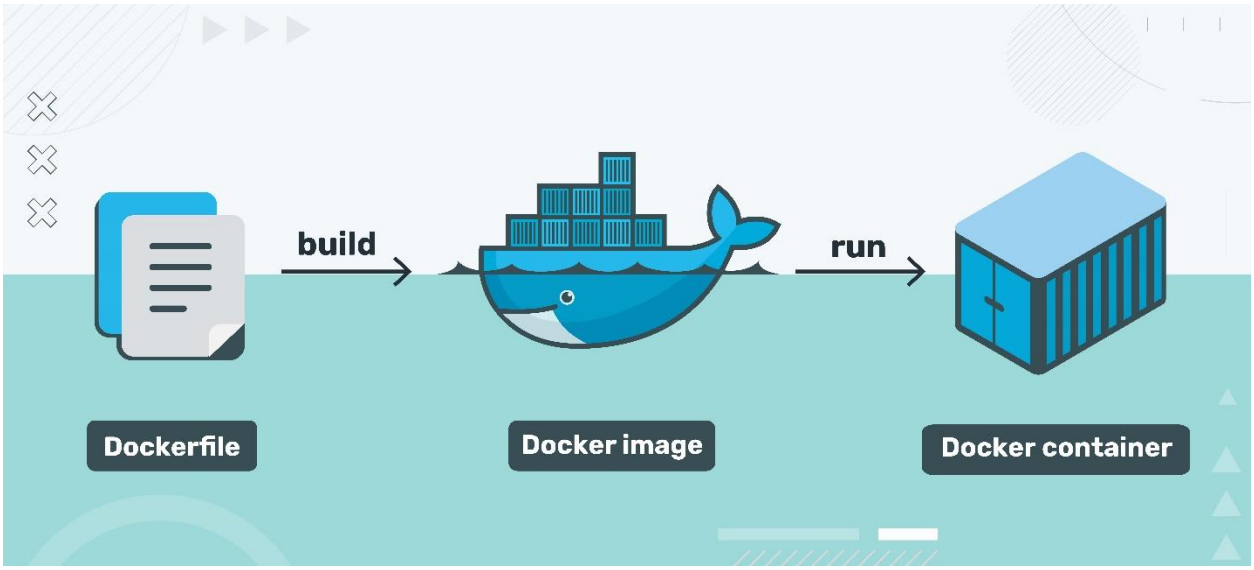
Para praticar o que vimos sobre documentação de APIs com Swagger, sugerimos que você utilize o projeto “Fixture Web” que apresentamos na aula 38 e realize o seguinte:

- Adicionar a biblioteca do SpringDoc ao POM;
- Acessar a URL configurada para visualizar a interface do Swagger;
- Ao executar o projeto, testar a API a partir desta mesma interface;
- Adicionar as anotações necessárias para indicar que os campos “id” e “nome”, dentro da entidade torneio não podem ser null.

Mão à obra!

Implantar APIs no Docker

Como sabemos, para criar um executável de uma aplicação Java, precisamos apenas executar o Maven Install, e ele criará um arquivo .jar que funcionará em qualquer computador que tenha o Java instalado. No entanto, pode ser ineficiente, pois pode nos trazer diversos problemas de incompatibilidade. Portanto, se usarmos o Docker, temos a possibilidade de abstrair dos problemas de execução da aplicação em diferentes computadores, vamos ver como podemos construir as imagens e executar os contêineres.



Convidamos você a assistir ao vídeo para conhecer mais sobre o Docker e ver como podemos construir as imagens e executar os contêineres.

Não tem o Docker instalado?

Você provavelmente já tem o Docker instalado desde quando estudou Infraestrutura I, mas se trocou de máquina, teve que desinstalá-lo ou por qualquer outro motivo não tenha o Docker instalado. Deixamos para você novamente os guias de instalação:

- Guia de instalação para Mac;
- Guia de instalação para Linux;
- Guia de instalação para Windows.

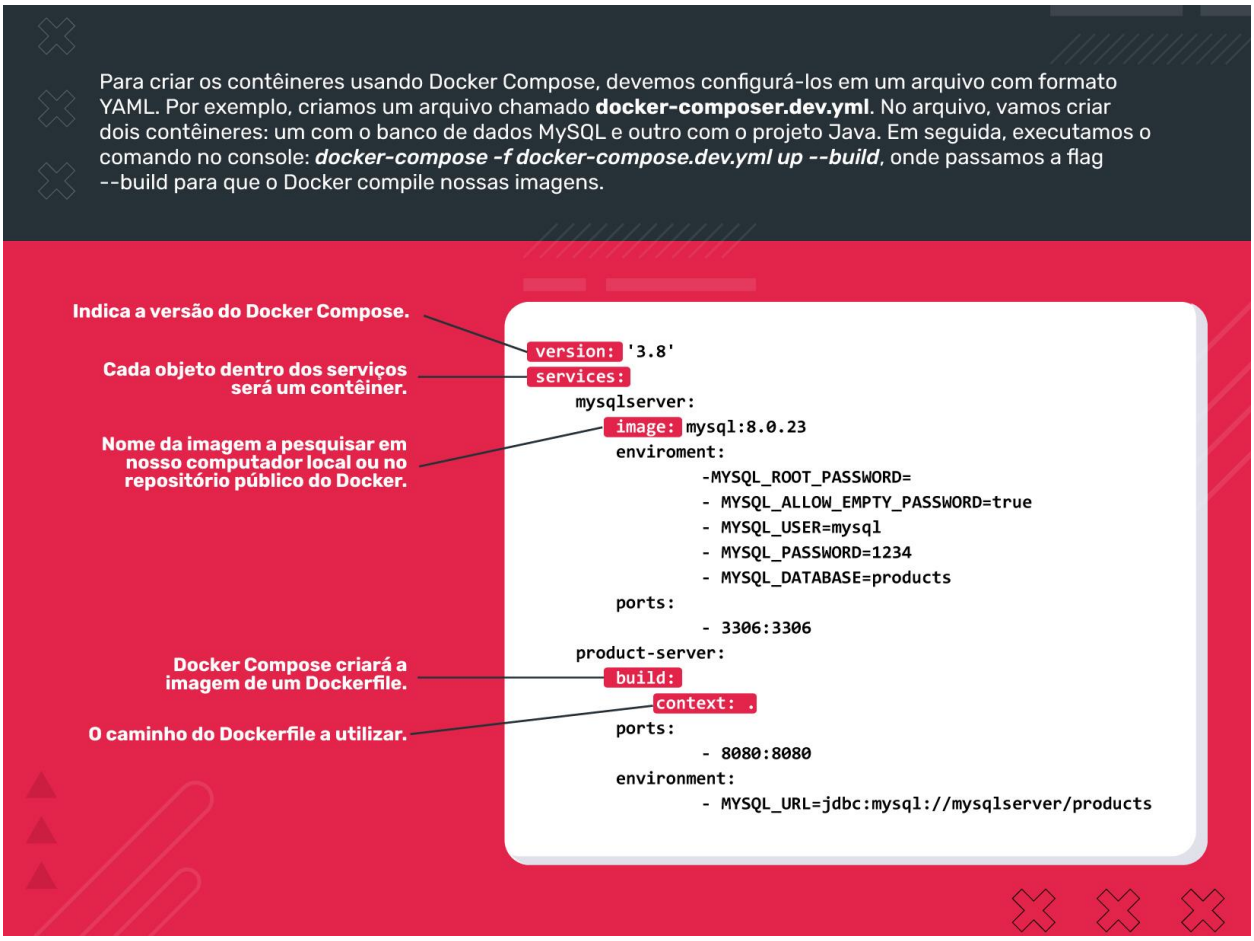
Processo de implantação

A partir da imagem que apresentamos anteriormente, veremos em detalhes o papel de cada uma de suas partes. Vamos lá!

Docker Compose

Os comandos de Docker e os Dockerfiles são muito úteis quando queremos criar contêineres individuais e não precisamos que eles tenham comunicação com outros contêineres.No entanto, se para implantar nossa aplicação precisarmos de mais de um contêiner, por exemplo, um contêiner para a aplicação Java e outro para o banco de dados ou muitos outros contêineres se trabalharmos com uma arquitetura de microsserviços, isso tornará muito mais difícil de gerenciar à medida em que escalamos nosso sistema.

Para resolver isso, o Docker nos fornece o **Docker Compose**, usamos essa ferramenta quando precisamos gerenciar vários contêineres, pois nos permitirá, por exemplo, iniciar todos os contêineres ao mesmo tempo, interrompê-los, etc., como se estivéssemos manipulando apenas um.



Dockerizando nossas aplicações

Para este desafio vamos trabalhar com o projeto “Fixture web” que usamos na aula 49. O que vamos fazer? Vamos executar nossa aplicação Java e o banco de dados Mongo com Docker.

Para fazer isso, temos que:

- Criar o arquivo Dockerfile para construir a imagem de nosso aplicativo em Java.
- Criar Docker-compose para executar nossa imagem e a imagem do Mongo.

Por último, teste se está funcionando corretamente utilizando o Postman ou a partir do Swagger.