05

Sobre Spring e Spring Boot

Transcrição

[00:00] Neste vídeo, vou falar um pouco sobre a história do Spring e do Spring Boot. Por isso, é um vídeo opcional. Nele, não vamos implementar nada na nossa API, e, caso você já conheça um pouco a história do Spring e do Spring Boot, ou não tenha interesse, você pode pular esse vídeo normalmente e assistir o próximo.

[00:19] O Spring é um dos frameworks mais antigos do Java. Foi criado em 2002, 2003, e, até hoje, está firme e forte no mercado, cada vez mais popular e sendo utilizado por desenvolvedores Java não só no Brasil, mas no mundo inteiro.

[00:38] O Spring foi desenvolvido por um desenvolvedor chamado Rod Johnson. Naquela época (2000/2002), o pessoal utilizava muito o Java EE para desenvolver aplicações corporativas com Java, ele era até chamado de J2EE. Dentro do J2EE você tinha algumas tecnologias, como RMI, JB, dentre outras, que eram um pouco complexas, e muitos desenvolvedores acabavam sofrendo quando iam desenvolver aplicações grandes, que precisavam ter uma boa performance e boa escalabilidade. Não por culpa do J2EE em si, mas por mal uso, por não conhecer boas práticas.

[01:18] O Rod era um especialista em J2EE. Ele sabia muito bem das armadilhas e como montar uma boa arquitetura: robusta, escalável e de fácil manutenção. Ele teve essa visão, viu essa dificuldade (se baseando nas consultorias que prestava) e teve a ideia de escrever um livro. Então, neste livro, ele escreveu

dicas de como montar uma arquitetura escalável, robusta, performática, usando o J2EE e sem cair nas armadilhas do mercado.

[01:52] O livro que ele escreveu se chama "Expert One-on-One: J2EE Design and Development". É um livro que ficou bem famoso. Além de mostrar esses problemas do J2EE e como você poderia montar uma boa arquitetura, ele também mostrava algumas das dificuldades e coisas ruins do J2EE em si. Ele apresentou no livro dele um código, como se fosse uma biblioteca alternativa. Nesse livro, ele escreveu mais de trinta mil linhas de código.

[02:26] Posteriormente, junto com o livro, foi publicado um fórum onde as pessoas podiam discutir e fazer o download da biblioteca. Várias pessoas começaram a baixar e utilizar nos projetos. Alguns desenvolvedores deram uma ideia para o Rod: por que não pegamos esse código, transformamos em uma biblioteca e lançamos no mercado gratuitamente, para ser uma alternativa ao J2EE?

[02:49] Depois, alguns desenvolvedores se reuniram com ele e eles foram criando essa biblioteca, que foi publicada em 2003, sendo que a versão 1.0 saiu em 2004. Assim nasce o Spring framework.

[03:00] A ideia do Spring era ser uma alternativa a esse modelo complexo do J2EE. O grande foco dele era em simplicidade de código. O coração do Spring foi todo baseado nos padrões de inversão de controle e injeção de dependências. Com isso, o seu código (a sua lógica de negócio, sua regra) não precisava correr atrás das dependências da infraestrutura. Nós invertíamos o controle e a infraestrutura chegava pronta para o seu código, para que ele fosse fácil de manter, desacoplado.

[03:32] Esse foi o coração do Spring. E isso agradou muito os desenvolvedores, que conseguiam implementar seu sistema com classes Java seguindo o padrão POJO, classe simples, sem ter muita dependência de infraestrutura, nem nada muito complexo.

[03:47] Com isso, o Spring acabou se tornando popular, foi evoluindo e sendo desenvolvido também em módulos. Você tinha o módulo central de "*IOC*", inversão de controles, e, em volta dele, você tinha alguns módulos, como MVC, módulo de segurança, de transação. Inclusive, ele até suportava o JavaEE, porque se integra com tecnologias como JPA e como o bean validation, que veremos no curso. Então, era uma alternativa, mas nunca foi um concorrente, nunca quis substituir o J2EE. Sempre quis ser um complemento. Inclusive, ele utiliza muitas das tecnologias do JavaEE.

[04:28] Ele foi evoluindo, foram surgindo novos módulos. Enquanto isso, o JavaEE também foi evoluindo. Foram surgindo outras tecnologias, o EJB, que era o grande coração do J2EE, que dava muito problema, muita dor de cabeça. Foram surgindo outras tecnologias e frameworks no mercado, como JSF, Wicket, Vaadin, depois o CDI, dentre outras coisas. Mas aí, o JavaEE acabou dando uma esfriada, tendo alguns problemas com a comunidade e ficando para trás. Em 2013, 2014, o pessoal do Spring criou o Spring Boot, que foi um projeto que revolucionou o desenvolvimento para Java e que fez o Spring alavancar de novo no mercado.

[05:13] A ideia do Spring Boot é que você consiga desenvolver uma aplicação sem o uso de um container, sem ser o desenvolvimento tradicional, em que você precisa ter sua aplicação e roda ela dentro de um servidor. Quando você termina de desenvolver, você gera um WAR e joga ele dentro de um servidor. A ideia do Spring Boot é inverter esse processo. Temos um servidor embutido e rodamos a aplicação em um `main`. Conforme vamos ver no curso, o nosso projeto vai ser inicializado por uma classe `main`.

[05:40] Com isso, não preciso mais gerar um WAR. Posso gerar o build da minha aplicação como sendo um JAR, que é muito mais leve e mais simples de ser executado. Isso acabou indo de encontro com ideias fortes no mercado, como a de desenvolvimento de micro serviços. O pessoal utiliza muito o Spring Boot para montar micro serviços, como o API REST. Se você tem uma API REST, para desenvolver com o Spring Boot é muito simples, porque, além do

contêiner ser embutido, o Spring Boot também simplificou uma coisa que era muito chata no Spring, que é a parte de configuração.

[06:18] Quando o Spring começou, toda a parte de configuração era feita via `.xml`. você tinha um `.xml` gigantesco configurando zilhões de coisas. Depois eles evoluíram e trouxeram o suporte para anotações, mas, mesmo assim, você tinha que criar várias classes com vários beans configurados com anotações. Ficava algo muito complexo. Para você criar um projeto do zero, você gastava um tempão só para fazer a parte de configuração.

[06:47] A ideia do Spring Boot é que muitas coisas já vêm configuradas por padrão para você. Você consegue criar um projeto e inicializá-lo de uma maneira muito rápida, muito produtiva, o que atraiu as empresas a utilizarem o Java.

[06:58] O Spring Boot foi evoluindo, e hoje está firme e forte no mercado como principal framework utilizado por desenvolvedores Java para construir aplicações REST, em micro serviços e em containers. Esses usos vão ao encontro da ideia de usar o Spring Boot (uma aplicação Java leve, simples) rodando dentro de um container, usando o Docker, por exemplo. Essa evolução acabou condizendo com as tendências de mercado. Graças a isso, o Spring disparou. O Boot foi o que fez o Spring disparar no mercado.

[07:36] O Java EE acabou decaindo. Não morreu, mas caiu em desuso. Hoje, é bem difícil ver um projeto sendo desenvolvido com Java EE. O pessoal prefere utilizar o Spring, Spring Boot e Spring Cloud, que é um conjunto de bibliotecas que foram utilizadas para montar micro serviços e ter toda a infraestrutura de micro serviços.

[07:58] Um grande "*case*" é o da Netflix, que tem vários projetos que o Spring Boot simplificou muito. Essa é a história do Spring e do Spring Boot. Por isso, recomendamos fortemente para você que é um desenvolvedor Java e que, necessariamente, vai trabalhar com desenvolvimento de aplicações Java, vai montar uma API REST e trabalhar com micro serviços: o Spring Boot e o Spring

Cloud são ferramentas indispensáveis. É o que está sendo utilizado no mercado, tanto no Brasil quanto no exterior.

[08:22] Neste treinamento, vamos aprender um pouco sobre o Spring Boot, vamos desenvolver nossa API REST e você vai se inteirar em como ele é utilizado para esse objetivo. Valeu por assistir esse vídeo. No próximo a gente continua a trabalhar na nossa aplicação.