03

Atualizando o Spring Boot

Transcrição

[00:00] Olá, pessoal, bem-vindos de volta a mais um treinamento de Spring Boot na Alura. Na aula de hoje, nessa primeira aula, vamos começar a trabalhar no projeto.

[00:11] E nesse terceiro curso de Spring Boot vamos aprender algumas coisas a mais que ficaram faltando nos outros dois treinamento.

[00:19] E para começar, a primeira coisa que precisamos fazer é atualizar a versão do Spring Boot no nosso projeto.

[00:27] Nós gravamos os dois primeiros cursos de Spring Boot na Alura mais ou menos em junho de 2019. Naquela época o Spring Boot estava na versão 2.1.4. Não sei se vocês lembram, nós criamos o projeto no site do Spring Initializr e a versão que tinha a estável, era a 2.1.4.

[00:49] Mas estamos gravando esse terceiro curso de Spring Boot em julho de 2020, e muita coisa mudou desde então. O projeto do Spring não ficou parado no tempo. A galera que faz a manutenção, que está evoluindo o projeto desenvolveu novas funcionalidades, novos recursos, corrigiram bugs. E muitas outras versões foram surgindo.

[01:15] Para não trabalharmos num projeto obsoleto, numa versão muito anterior, a primeira coisa que faremos é atualizar a versão do Spring Boot no nosso projeto. Se entrarmos no site do Spring Initializr, em “start.spring.io”, veremos que a versão que eles sugerem se você for criar um projeto do zero, versão que já vem marcada por padrão é a 2.3.1.

[01:38] A versão atual, no momento de gravação do vídeo desse terceiro curso de Spring Boot, é a 2.3.1. Então estamos muito desatualizados, nosso projeto está na versão 2.1.4. Então a primeira coisa que faremos é atualizar o Spring Boot no projeto.

[01:56] Agora estou no Eclipse, já estou com meu projeto importado. É o mesmo projeto que foi finalizado no segundo curso, o fórum da Alura com todas as funcionalidades que vimos no segundo curso do Spring Boot.

[02:11] Lembra que estamos usando o Maven, então precisamos abrir o `pom.xml`. E lembra que tem aquela tag `<parent>`, e está ali a versão 2.1.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.4.RELEASE</version>
        <relativePath /> <!-- lookup parent from repository
-->
    </parent>
    <groupId>br.com.alura</groupId>
    <artifactId>forum</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>forum</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>
```

```
//código omitido
</project>
```

COPIAR CÓDIGO

[02:23] E temos que trocar para 2.3.1. Vou trocar e salvar. Se você fizer isso pela primeira vez no seu computador o Maven vai detectar que você trocou a versão do projeto, vai baixar todas as novas dependências na internet, então pode ser que demore um pouco. É só você esperar que ele mostra o progresso no canto inferior direito da tela. No meu caso eu já tinha feito isso antes, então foi um pouco mais rápido.

[02:48] Nós estamos atualizando o projeto, na teoria não deveria mudar muita coisa, vamos aproveitar os novos recursos. Mas na prática, quem já trabalhou com desenvolvimento de software já deve ter passado por isso, toda vez que você vai atualizar uma biblioteca em um projeto você tem que tomar muito cuidado.

[03:04] Porque você pode correr um risco de atualizar uma biblioteca, sendo que essa nova versão pode ter gerado uma quebra de compatibilidade. Pode ter alguma funcionalidade, alguma classe, algum método que mudou a assinatura, foi apagado, enfim. Então isso pode gerar um erro de compilação ou uma mudança no comportamento de determinada funcionalidade.

[03:27] Então para fazer essa mudança você tem que tomar muito cuidado, você não pode simplesmente chegar e atualizar. Você tem que pesquisar e fazer isso com calma para ver se não vai gerar problemas.

[03:36] Inclusive, no nosso caso, você pode ver que tem um "X" vermelho no projeto, na parte esquerda da tela. Então no "src/main/java", em "br.com.alura.forum", no pacote "controller" deu erro de compilação. Acabamos de ter um problema no nosso projeto.

[03:53] Vamos abrir, por exemplo, a classe "TopicosController", que é onde está tendo um erro de compilação. E o problema está justamente no nosso métod

`cadastrar`. Ele recebe aquele `TopicoForm`, aquele objeto seguindo o padrão DTO e a anotação `@Valid`, do BIN Validation.

[04:11] Ele está falando que não encontrou a anotação `@Valid`. Então se formos ao início da classe e expandirmos os *imports*, veremos que o pacote dela está correto, é `javax.validation.Valid;`.

[04:24] Além disso, no pacote "form", os nossos *forms* estão com problema também. Todas as anotações do BIN Validation estão com erro de compilação. E ele está dizendo que não encontrou essas classes. E esse é justamente o problema que aconteceu com o nosso projeto.

[04:41] Na versão 2.3.0 do Spring Boot eles fizeram uma mudança e o BIN Validation não vem mais por padrão no projeto. Se você se você quiser utilizar o BIN Validation no seu projeto você tem que adicionar de maneira separada a dependência do BIN Validation.

[04:59] Antes ela vinha automática. Só de você importar o módulo web já vinha embutido o BIN Validation. Agora não vem mais.

[05:06] Inclusive, deixa eu mostrar pra vocês, eu entrei no site do GitHub, na página do projeto do Spring Boot, "github.com/spring-projects/spring-boot", que é o código fonte do Spring Boot; o Spring Boot é uma biblioteca de código fonte aberto.

[05:24] Então ele tem todo o código fonte do Spring Boot, tem embaixo uma documentação explicando o que é o projeto, mas eu quero que vocês subam e deem uma olhada na lateral direita, na parte de *Releases*.

[05:37] Ele está mostrando que a última versão, a última *release* é a 2.3.1. Vamos clicar nela. No caso do Spring Boot, sempre que você quiser atualizar para uma versão, eu recomendo que você entre no site do projeto, veja as notas do *release* e veja o que mudou.

[05:55] Então ele fala quais são as novas funcionalidades, quais são os bugs que foram corrigidos, documentação que foi atualizada, ele te fala tudo que mudou.

[06:05] Em cima da parte “New Features”, logo abaixo do número da versão, se tivesse algum manual de incompatibilidades apareceria um link. Não apareceu porque na verdade vamos usar a versão 2.3.1, mas foi na 2.3.0 que teve essa mudança.

[06:22] Então na barra de endereço vou trocar de 2.3.1 para 2.3.0. E já na versão 2.3.0, ele diz “para fazer o upgrade leia essas instruções”. Ao clicar ele vai descrever as mudanças caso você esteja fazendo upgrade da versão 2.2. Ele explica o que mandou, e ele fala que validação não é mais inclusa no módulo web.

[06:52] Caso você queira utilizar validação, você tem que adicionar essa dependência descrita. Foi isso que aconteceu no nosso projeto.

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation<|artifactId>
</dependency>
```

COPIAR CÓDIGO

[06:58] Vou dar um “Ctrl + C” nessa dependência, vou voltar no Eclipse, no `pom.xml`, e na parte de dependências vou colar. Dou um “Ctrl + Shift + F” para formatar. E agora eu tenho um módulo de validação.

[07:15] Antes já vinha embutido por padrão do módulo web e agora você tem que adicionar manualmente. Então vou salvar, ele vai recompilar o projeto. Vou clicar com o botão direito no projeto, escolher “Maven > Update Project”, só para ele atualizar.

[07:31] E perceba que já pararam os erros de compilação. Então cuidado, sempre que você atualizar uma biblioteca no projeto dê uma analisada se não teve nenhum impacto.

[07:41] Então a ideia era só atualizarmos o Spring Boot para pegar uma versão mais recente, mais atual e ver quais foram os impactos; e eu queria mostrar para vocês onde que você consegue pesquisar, no caso do Spring Boot é no GitHub mesmo. Sempre que tem uma nova *release* eles publicam essas notas com as mudanças que foram introduzidas nesta nova versão.

[08:01] Então é importante você ler para ver se não teve nenhum impacto. Eu até coloquei como uma curiosidade a questão de versionamento de software. Geralmente no versionamento de software tem esses números, como a versão 2.3.1, ou 2.3.2. O que significam esses números?

[08:20] O primeiro número é o principal, chamado de Major. É o número que representa uma mudança maior no projeto. Então sempre que tem uma mudança no número, como de 1.0 para 2.0, é uma mudança drástica que pode ter quebra de compatibilidade.

[08:34] O segundo número é o Minor. É quando vai de 2.2 para 2.3, por exemplo. Eu ainda estou na versão 2, só que saiu da 2.2 para a 2.3. Pode ter mudanças que quebrem. Geralmente não vai quebrar a compatibilidade, serão novas funcionalidades, mas às vezes acontece.

[08:50] E o último número é o indicador do Patch. É uma mudança de uma versão 2.3.0 para 2.3.1, por exemplo. Então mudou só o último número, que é o número de Patch. Nesse caso é bem raro ter quebra de compatibilidade.

[09:04] Quando muda esse último número é só correção de bug, algumas correções de segurança, coisa pontual. Não foram novas funcionalidades, nada que vai causar um impacto muito grande.

[09:14] Como no nosso caso saímos da versão 2.1 para a 2.3, nós pulamos 2 Minor versions. Então foi um salto muito grande, no caso acabou quebrando compatibilidade porque o pessoal do Spring Boot deixou opcional aquela parte de validação.

[09:30] Afinal, nem todo projeto tem validação. É bem comum de ter, mas vai que um dia não tenha essa obrigatoriedade, então eles deixaram de fora.

[09:39] Então por hoje esse era o objetivo da nossa aula, ver como atualizar o projeto, ver essas questões de versionamento de software e ver a página do GitHub para acompanhar essas mudanças e ver quais são os impactos.

[09:52] Na próxima aula começaremos de fato a mexer no projeto e implementar as novas funcionalidades que veremos no terceiro curso. Espero vocês no próximo vídeo. Um abraço e até lá.