

Olá, tudo bem?

Nesta videoaula, você conhecerá o significado do Spring MVC, estudará sobre a inclusão de uma nova dependência no arquivo pom.xml (pom ponto xml), a partir do repositório central do Maven; e aprenderá a configurar e integrar o Spring MVC ao seu projeto. Então, preparados? Ótimo, pois vamos iniciar nossa jornada pelo framework mais famoso do mundo Java.

Antes de estudar a aplicação prática do Spring MVC, você conhecerá uma breve história sobre a origem do Spring, aprenderá sobre os módulos que o compõem e observará alguns exemplos de onde ele é aplicado atualmente. Por isso, é importante que você saiba que o Spring é um framework de código aberto (open source), criado por Rod Johnson e direcionado à plataforma Java. Além disso, ele foi lançado em dezembro de 2013 e tem como principal característica utilizar uma arquitetura baseada em inversão de controle (IoC) e injeção de dependência, a fim de facilitar a criação de instâncias e a inclusão de novos comportamentos para construção da classe, em uma aplicação orientada a objetos. Portanto, isso reduz acoplamento e facilita manutenção do código.

O ecossistema do Spring é composto por módulos voltados a aplicações web (Spring MVC), persistência de dados (Spring Data), segurança (Spring Security) e mensageria (Spring AMQP), por exemplo. Caso você deseje obter mais informações acerca dos módulos que fazem parte do ecossistema do Spring, acesse o link <https://spring.io/projects> (https dois ponto barra barra spring ponto io barra projects).

Continuando com o nosso assunto principal, talvez você nunca tenha imaginado, mas, por trás daquelas atrações que o Netflix te proporciona, no conforto da sua casa, há várias soluções que foram implementadas utilizando o Spring, como acessar sua conta ou efetuar login, processar o pagamento da sua mensalidade e, até mesmo, na transmissão daquele filme que você tanto gosta.

Com isso, você pode observar que há uma enorme variedade de aplicações do Spring. Contudo, nesta videoaula, você estudará com mais detalhes o módulo do Spring direcionado a criação de aplicações web, isto é, o Spring Web MVC ou simplesmente Spring MVC. Atualmente, ele se encontra na versão 5.2.6.RELEASE (cinco ponto dois ponto seis ponto release) e tem como objetivo implementar uma arquitetura - modelo de organização e comportamento das classes - baseada no padrão de projeto MVC (model-view-controller), além de mapear os responsáveis por processar as requisições que chegam ao servidor.

Portanto, o Spring MVC possui um interceptador de requisições chamado DispatcherServlet. Ele é um servlet que analisa as requisições que chegam ao servidor de aplicação Tomcat e as repassa ao controller que processa os dados e os envia, à camada de persistência, o model para ser armazenado na base de dados. Assim, para indicar que o processamento da requisição foi bem

sucedido, uma resposta é enviada, por parte do controller, ao DispatcherServlet, indicando para quem ele deve encaminhar o retorno. Novamente, o DispatcherServlet identifica para quem ele deve enviar uma resposta, e, após o envio dela, finaliza o fluxo de execução.

#Audiodescrição: A imagem mostra um fluxograma que ilustra a ação do interceptador de requisições, chamado “DispatcherServlet”. À esquerda do fluxo, está escrita a palavra “Requisição”, que é ligada por uma seta a um retângulo representando o “Dispatcher Servlet”. Desse retângulo, sai uma seta diagonal para cima apontando para o retângulo representando “Handler Mapping”; outra seta bidirecional para baixo conecta com um retângulo representando “View”; uma seta diagonal para um retângulo representando “View Resolver”; e uma seta continua o fluxo, conectando a “Controller”. Do retângulo “Controller”, sai uma seta até outro retângulo representando “Model”. Dele, sai uma seta até um cilindro abaixo, representado “DataBase”. De DataBase, sai uma seta para cima que retorna por todas as outras etapas até o “Dispatcher Servlet”, gerando uma “Resposta”. Entre a etapa “Controller” e “Dispatcher Servlet”, a seta que liga essas duas etapas do fluxo de retorno traz seu próprio retângulo, com a informação “Model and View”.

Até aqui, você pode conhecer um pouco da história do Spring e entender seu fluxo de funcionamento. Agora, você aprenderá a encontrar a dependência no repositório central do Maven, associada ao Spring MVC, e, em seguida, configurar seu projeto.

Para encontrar a dependência no repositório central do Maven, você deve abrir seu navegador de internet e acessar o link <https://mvnrepository.com> (https dois pontos barra barra mvnrepository ponto com). Na barra de busca do repositório central do Maven, digite Spring MVC e, depois, acione o botão search ou pesquisar. Feito isso, uma lista de resultados com todas as dependências relacionadas ao Spring será apresentada a você.

#Audiodescrição: A imagem apresenta uma captura de tela do Maven Repository. No campo de busca da página, está escrito “spring mvc”. Logo abaixo, à esquerda, há os menus “Repository”, “Group” e “Category” com suas respectivas funções. À direita, há uma lista com os resultados da pesquisa.

Agora, escolha a opção Spring Web MVC. Após essa ação, uma lista de resultados com todas as versões disponíveis, até o momento, será exibida para você.

#Audiodescrição: A imagem mostra uma captura de tela da página do Maven Repository, após ter selecionado a opção “Spring Web MVC” na página de busca. À esquerda, há um gráfico ascendente, representando os artefatos indexados. Logo abaixo desse gráfico, há uma lista de “Popular Categories”. À direita da tela, é exibida uma lista de releases do Spring Web MVC, cada um com sua versão, repositório, número de usos e data de publicação.

Ainda no repositório central do Maven, escolha a opção 5.2.6.RELEASE (cinco ponto dois ponto seis ponto release). Essa é a versão mais recente do Spring MVC. Após essa ação, uma página web será mostrada a você com informações sobre a dependência que está buscando. Nela, conterà os dados sobre licença de uso, a data de publicação, o mantenedor e muito mais.

#Audiodescrição: A imagem mostra a captura de tela do Maven Repository, que mostra as informações sobre a Spring Web MVC 5.2.6 Release. À esquerda, há um gráfico ascendente, representando os artefatos indexados. Logo abaixo desse gráfico, há uma lista de “Popular Categories”. À direita da tela, há os dados sobre licença de uso, a data de publicação, o mantenedor, entre outras. Abaixo dessas informações, temos as abas “Maven”, “Gradle”, “SBT”, “Ivy”, “Grape”, “Leiningen” e “Buildr”, dentre as quais a aba “Maven” está aberta, possuindo uma caixa de texto com o seguinte trecho de código:

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>pring-webmvc</artifactId>
  <version>5.2.6.RELEASE</version>
</dependency>
```

Então, navegue até a aba com o nome Maven. Nela, você encontrará o conteúdo que precisa para adicionar o Spring MVC ao seu projeto.

O próximo passo consiste em fazer a inclusão da dependência do Spring MVC em seu projeto. Para isso, você deve copiar o conteúdo da caixa de texto da aba Maven e, depois, colar, dentro da tag dependencies, o seu arquivo pom.xml (pom ponto xml), que se encontra na raiz do projeto CatalogoDeFilmes.

Em seguida, você deve aguardar o gerenciador de dependências, o Maven, se encarregar de fazer o download da biblioteca do Spring MVC e incluí-la em seu projeto. Lembre-se de que tudo isso ocorrerá de forma transparente, em segundo plano, dentro do Eclipse.

Após a inclusão da dependência do Spring MVC, o seu pom.xml (**pom ponto xml**) vai conter, entre as tags dependencies, uma nova biblioteca chamada spring-webmvc (**spring webmvc**) na versão 5.2.6.RELEASE (**cinco ponto dois ponto seis ponto release**) do grupo org.springframework (**org ponto springframework**), que será incluído no seu projeto a partir do Maven.

#Audiodescrição: A imagem mostra o código do exemplo 1.

Pronto! O gerenciador de dependências Maven está configurado. Aos poucos, todo esse processo se tornará familiar para você, por isso, você não deve ficar tenso. Calma. É comum a busca de dependências na central de repositório do Maven para um projeto que se está construindo, seja de algo simples, como uma biblioteca para formatação de data; ou um framework como o Spring.

A partir desse ponto, você deve focar na configuração do Spring MVC, para que ele possa se comunicar com o seu projeto e gerenciar as requisições vindas dos clientes. Assim como as respostas a serem enviadas.

No arquivo web.xml (**web ponto xml**), dentro das tags web-app (**web app**), que se encontra no diretório WebContent/WEB-INF (**webcontent barra web inf**) do seu projeto, você deve adicionar as tags de criação do servlet DispatcherServlet do Spring MVC. Ele é como o próprio nome diz, um Servlet que tem o objetivo de gerenciar e mapear quem são os responsáveis por processar as requisições que chegam ao servidor.

Na tag servlet do arquivo web.xml (**web ponto xml**), você tem a descrição do DispatcherServlet, através de nome, parâmetros de inicialização e classe responsável por gerenciar e mapear quem são os responsáveis por processar as requisições que chegam ao servidor. Além disso, você tem a tag servlet-mapping (**servlet mapping**) que indica qual a URL que esse Servlet irá atender.

#Audiodescrição: A imagem mostra o código do exemplo 2.

Para finalizar o processo de configuração, você irá informar ao Spring MVC qual o seu contexto de execução. De forma prática, deve-se instruir o Spring a encontrar pacotes, classes e views, que ele precisa administrar na inicialização do servidor.

No arquivo *web.xml* (**web ponto xml**), especificamente nas configurações do `DispatcherServlet`, você encontra a tag `init-param` (**init param**), e nela, o parâmetro `contextConfigLocation` que recebe como valor `/WEB-INF/spring-context.xml` (**barra web inf barra spring context ponto xml**). Este arquivo, o *spring-context.xml* (**spring context ponto xml**), contém um conjunto de propriedades que direcionam o Spring MVC a criar seu contexto de execução, ou seja, os pacotes, as classes e as views, da qual ele precisa administrar.

No arquivo *spring-context.xml* (**spring context ponto xml**), estará descrito onde se encontram os pacotes da qual o Spring MVC precisa gerenciar, a partir da tag `context:component-scan` base-package (**context dois ponto component scan espaço base package**). Além disso, declara-se o `InternalResourceViewResolver`, o encarregado por informar onde as views estão localizadas, ou seja, em `/WEB-INF/views/` (**barra web inf barra views barra**) e qual a extensão delas, que, neste caso, é `.jsp` (**ponto jsp**).

#Audiodescrição: A imagem mostra o código do exemplo 3.

Agora, que você realizou toda a configuração, deve-se realizar a inicialização do Spring MVC. Para isso, você vai utilizar o servidor Tomcat no Eclipse, e navegar até o menu Run, escolhendo a opção Run on Server. Em seguida, o Eclipse irá inicializar o servidor de aplicação, que é o Tomcat.

#Audiodescrição: A imagem mostra uma captura de tela do Eclipse, inicializando o Spring e o `DispatcherServlet`. No topo, temos as abas "Markers", "Properties", "Servers", "Data Source Explorer", "Snippets", "Console" e "Progress", dentre as quais a aba "Console" está aberta. Nela, são exibidos os dados da inicialização em letras vermelhas.

Pronto! A partir do console do Eclipse, você pode ver a inicialização do Spring e do `DispatcherServlet`, ambos estão em execução e prontos para o trabalho. Até aqui, você aprendeu como configurar o Spring MVC e conheceu seu interceptador de requisições o `DispatcherServlet`. Tudo isso formará a base para você entender o funcionamento da arquitetura do Spring MVC e como ele trata as requisições que chegam no servidor. Na próxima videoaula, você irá conhecer o *controller*, um dos recursos disponíveis no Spring MVC.

Até a próxima aula e bons estudos!

Referências

Apostila Java para Desenvolvimento Web. Disponível em: <<https://www.caelum.com.br/apostila-java-web/spring-mvc/>>. Acesso em: 01 jun 2020.

BRITO, Michelli. **Spring MVC e o Dispatcher Servlet**. Disponível em: <<https://medium.com/@michellibrito/spring-mvc-e-o-dispatcher-servlet-15bd5575e2d8>>.

Acesso em: 01 jun 2020.

Spring 4 MVC HelloWorld Tutorial – Annotation/JavaConfig Example. Disponível em: <<http://websystique.com/springmvc/spring-4-mvc-helloworld-tutorial-annotation-javaconfig-full-example/>>. Acesso em: 01 jun 2020.

Spring MVC Example. Disponível em: <<https://www.journaldev.com/14476/spring-mvc-example/>>. Acesso em: 01 jun 2020.

Spring MVC Interview. Disponível em: <<https://www.baeldung.com/spring-mvc-interview-questions>>. Acesso em: 01 jun 2020.

Web on Sevlet Stack. Disponível em: <<https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>>. Acesso em: 01 jun 2020.

What is dispatcher servlet in spring. Disponível em: <<https://stackoverflow.com/questions/2769467/what-is-dispatcher-servlet-in-spring>>. Acesso em: 01 jun 2020.