

Criação do projeto

Transcrição

Bem-vindo ao curso de SpringMVC! Neste curso estudaremos o SpringMVC e alguns dos seus módulos para resolver problemas reais. Neste curso criaremos um projeto semelhante a **Casa do Código**, no qual há várias peculiaridades que desenvolveremos com o SpringMVC.

SpringMVC é um framework Java e como todo projeto java, temos que configurar nosso ambiente de desenvolvimento para conseguirmos desenvolver o projeto. Neste caso, usaremos o **JBoss Forge**, um framework criado pela **JBoss** para a criação da estrutura inicial do projeto.

Antes de continuarmos, baixe o JBoss Forge no site [forge.jboss.org \(http://forge.jboss.org/\)](http://forge.jboss.org/), descompacte em um diretório de sua preferência e renomeie o diretório do JBoss Forge para **forge**. Este último passo foi feito para deixar mais simples a interação com o JBoss Forge, pois o usaremos através do terminal.

Com o terminal aberto, navegue até o diretório onde está o JBoss Forge (pasta **forge** renomeada no passo anterior) e execute o **JBoss** dentro de `bin/forge`. Este executável é um terminal disponibilizado pelo JBoss Forge para que possamos configurar a estrutura inicial do projeto.

Atenção: O passo anterior pode demorar um pouco para ser finalizado, pois o JBoss irá carregar uma série de itens para que possamos usá-lo.

Neste momento, já dentro do terminal do JBoss, podemos digitar e executar o seguinte comando para criar o projeto: `project-new --named casadocodigo`. Este comando indica para o JBoss que queremos um novo projeto (`project-new`) e que este projeto terá o nome **casadocodigo** (`--named casadocodigo`).

Observação: Nas versões mais recentes do JBoss o comando para criar um novo projeto mudou para `new-project`, ou seja, utilize o comando `new-project --named casadocodigo` para criar o projeto caso encontre o erro: `***ERROR*** No such command: project-new`.

```
*** \*\*\*Success\*\*\* ** Project named 'casadocodigo' has been created!"
```

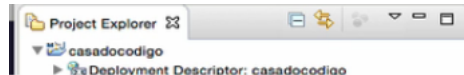
Sucesso! Nosso projeto inicial está criado!

Observação: Se verificar no site do JBoss Forge, notará que o projeto criado por este framework, é um projeto **Maven-based**, ou seja, um projeto baseado no **Maven**. Note que na pasta onde o JBoss Forge há uma nova pasta com o nome do projeto.

Podemos agora abrir o eclipse e fazer os seguintes passos: Abrir a ferramenta de import, escolher que queremos importar um projeto Maven e selecionar o projeto criado com o JBoss Forge!

1. Clique com o direito no project explorer do Eclipse e vá até a opção de import e selecione import.
2. Na caixa de filtro, digite `maven` para que o eclipse filtre as opções de importe e selecione `Existing Maven Projects`.
3. Em `Root Directory`, selecione `Browse` e navegue pelas pastas até chegar na pasta do projeto (`casadocodigo`).
Selecione a pasta do projeto e clique em `Open`. Após isso note que o eclipse reconheceu o projeto e já exibe na listagem o `pom.xml` do projeto. Clique em `Finish`!

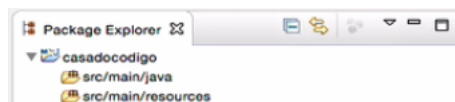
Nosso projeto agora é listado no eclipse, com a estrutura padrão criada com o JBoss Forge. Seu projeto deve estar parecido com a imagem abaixo:



Para deixarmos o projeto com um fluxo um pouco mais simples de se trabalhar e visualizar, vamos mudar a perspectiva do eclipse de **Java EE** para **Java**. Podemos fazer isso acessando os menus:

Window -> Perspective -> Open Perspective -> Other

selecionando a perspectiva **Java** e clicando em **ok** note como a estrutura do projeto é exibida de forma mais simples.



Experimente explorar as pastas do projeto!

Com a estrutura inicial do projeto pronta, podemos começar a criar algumas páginas para podermos visualizar o projeto funcionando. Clique na pasta **webapp** do projeto e use o atalho **ctrl + n** para criarmos um novo arquivo. Na caixa de seleção que aparece, digite **html** para criarmos um arquivo HTML. Selecione a opção **HTML File**. Em **File name** digite **index.html** e clique em **Finish**.

O código do **index.html** deve estar parecido com este:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>

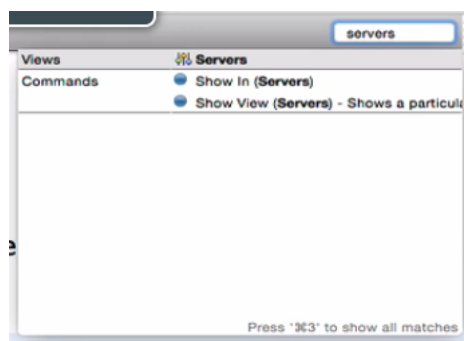
</body>
</html>
```

Modifique o título, coloque algum código no corpo do HTML, para visualizar depois como ficou. Algo como:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Livros de Java, Android, iPhone, PHP, Ruby e Muito mais - Casa Do Código</title>
</head>
<body>
    <h1>Oi</h1>
</body>
</html>
```

Com este último passo, estamos prontos para vermos nosso projeto funcionando, mas temos que configurar mais uma coisa, **o servidor!**

Baixe o Tomcat (versão 7) no site da [Apache \(http://tomcat.apache.org/download-70.cgi\)](http://tomcat.apache.org/download-70.cgi) e descompacte em um diretório de sua preferência. Após isso, no eclipse use a barra de pesquisa na parte superior direita e procure por `servers`. Abra esta aba.



Quando a aba de servers do eclipse for aberta, clique com o direito do mouse na área da aba aberta e selecione `new -> server`. Há uma lista de servidores que podemos usar, vamos selecionar o `Apache -> Tomcat V0.7 Server` e clique em `Next`. O próximo passo é selecionar o diretório onde nosso Tomcat está instalado, clique em `Browse` e navegue pelas pastas até encontrar a pasta do Tomcat e selecione esta pasta e clique em `Finish`.

Nosso servidor deve ser listado na aba servers do eclipse. Devemos agora adicionar nosso projeto no servidor. Clicando com o botão direito sobre o servidor e selecionando a opção `add/remove` visualizamos nosso projeto na listagem. Para adicioná-lo ao servidor, basta selecionar o projeto e clicarmos em `add` e depois em `Finish`. Podemos iniciar então o servidor clicando com o direito novamente sobre o servidor e selecionando a opção `start`.

Nosso projeto já está rodando no servidor, para verificar, basta acessar com um navegador a url:

`localhost:8080/casadocodigo`.

Apesar de funcionar, não é esse nosso real objetivo, queremos nossa lista de livros igual a casa do código!

Até o momento, não usamos nada do SpringMVC e neste ponto que sabemos que está tudo funcionando perfeitamente, podemos começar a usar o SpringMVC de fato!

