01

Introdução

Transcrição

Você pode fazer o DOWNLOAD (https://s3.amazonaws.com/caelum-online-public/jsf-primefaces/stages/capitulo-2.zip) do projeto inicial. Esse projeto foi criado no primeiro curso sobre JSF.

Configurando o ambiente

Se você já tem o ambiente pronto, pois assistiu ao primeiro treinamento sobre JSF, você pode pular essa parte e seguir com o curso, caso você não tenha assistido o treinamento, vamos ajudá-lo a configurar o seu ambiente.

Nesse treinamento, utilizaremos como IDE a versão **Java EE** do **Eclipse**, você pode baixá-lo aqui (https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-java-ee-developers). Após o download, você pode extrair o arquivo e começar a utilizá-lo. Como servidor, utilizaremos o **Tomcat 8** (https://tomcat.apache.org/download-80.cgi), basta baixá-lo e extrair o zip. Além disso, também utilizaremos o **MySQL** como banco de dados, mas para a aplicação funcionar, devemos criar o *database* `livrariadb` . Abra o console do MySQL e digite:

```
mysql> create database livrariadb;
```

Com o *database* criado, vamos criar o *server* no Eclipse. Na aba **Servers**, clique no link disponibilizado para criar um novo servidor. Na tela que abrir, selecione a versão 8 do Tomcat e clique em *Next >*. Clique em *Browse...*, selecione o diretório onde o Tomcat foi extraído e clique em *Finish*.

O próximo passo é importar o projeto inicial, você pode baixá-lo aqui (https://s3.amazonaws.com/caelum-online-public/jsf-primefaces/stages/capitulo-2.zip). Com o download feito, importe o projeto no Eclipse, clicando em *Import -> Existing Projects into Workspace*, marque a opção *Select archive file*, selecione o zip baixado e clique em *Finish*.

Por último, não esqueça de associar a aplicação com o Tomcat, clicando nele com o botão direito, e selecionando *Add and Remove...*. Depois basta selecionar o projeto, clicar em *Add >* e depois em *Finish*.

Ótimo, configuramos o ambiente e estamos prontos para começar a utilizar o Primefaces.

Testando a aplicação

Podemos agora testar a aplicação, acessando o link http://localhost:8080/livraria/livro.xhtml (http://localhost:8080/livraria/livro.xhtml). Repare que somos redirecionados para a página de login, pois precisamos estar logados para poder acessar as páginas da nossa aplicação. Tente fazer o login com qualquer usuário e senha, assim o JPA criará as tabelas para nós e poderemos inserir um usuário válido no banco.

Há um script que vocês podem baixar [aqui][5], nele há comandos SQL para popular o nosso banco. Execute esses comandos e volte para a aplicação, agora podemos fazer o login e acessar a aplicação utilizando o usuário **nico.steppat@caelum.com.br** e a senha **12345** (ou com algum outro usuário que você tenha criado).

Após isso, podemos dar uma olhada no código da aplicação. Na pasta `WebContent`, temos três páginas, **autor**, **livro** e **login**, todos eles são arquivos `xhtml` e utilizam o `_template.xhtml`. Na pasta `WEB-INF` temos as bibliotecas e os arquivos `faces-config.xml`, que é o arquivo de configuração do JSF, e `web.xml`.

Cada tela possui um *bean* associado, que delega para os respectivos modelos, que são entidades (as tabelas do banco são criadas baseadas nesses modelos).

No pacote `dao`, temos um `DAO` genérico e um específico para o usuário, que faz a query de autenticação, para descobrir se o usuário está no banco, além de ter uma classe para popular o banco e a classe `JPAUtil`, para inicializar o JPA.

Por fim, no pacote `util`, temos os dois *phases listener*, o LogPhaseListener monitora o ciclo da vida, as fases do JSF e o Autorizador, que faz com que a nossa aplicação sempre passe pelo login.

Com tudo configurado e apresentada a nossa aplicação, começaremos com o Primefaces a partir do próximo video!