01

Primefaces na página de livros

Transcrição

Começando daqui? Você pode fazer o DOWNLOAD (https://s3.amazonaws.com/caelum-online-public/jsf-primefaces/stages/capitulo-4.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Primefaces na página de livros

Começaremos a alterar a página `livro.xhtml`, mas essa tarefa será dividida em dois capítulos. Neste capítulo, focaremos no formulário de cadastro de livros e no próximos alteraremos a tabela de exibição dos livros cadastrados.

Antes de utilizar o Primefaces na página, não podemos nos esquecer de declarar o seu *namespace* dentro da tag `<html>`:

```
xmlns:p="http://primefaces.org/ui"
```

E vamos mexer no que já foi visto no treinamento, adicionar o componente `<h:outputPanel>` e alterar o `<fieldset>`, `<h:outputLabel>`, `<h:inputText>` e o `<h:messages>`. Você pode dar um simples `CTRL+F` e substituir esses componentes pelos os do Primefaces. Vamos trocar o `panelGrid` também, para utilizar o do Primefaces:

```
<ui:define name="titulo">
    <p:outputPanel>Novo Livro</p:outputPanel>
</ui:define>

<ui:define name="conteudo">
    <h:form>

        <p:messages id="messages" />

        <p:fieldset legend="Dados do Livro">
            <p:panelGrid columns="2">

                <p:outputLabel value="Titulo:" for="titulo" />
                <p:inputText id="titulo" value="#{livroBean.livro.titulo}"
                    required="true" requiredMessage="Título obrigatório"
                    validatorMessage="Título não pode ser superior a 40">
                    <f:validateLength maximum="40" />
                    <f:ajax event="blur" render="messages" />
                </p:inputText>

                <p:outputLabel value="ISBN:" for="isbn" />
                <p:inputText id="isbn" value="#{livroBean.livro.isbn}"
                    validator="#{livroBean.comecaComDigitoUm}" />

                <p:outputLabel value="Preço:" for="preco" />
                <p:inputText id="preco" value="#{livroBean.livro.preco}" />

                <p:outputLabel value="Data de Lançamento:" for="dataLancamento" />
```

```
                <p:inputText id="dataLancamento"
                    value="#{livroBean.livro.dataLancamento.time}">
                    <f:convertDateTime pattern="dd/MM/yyyy"
                        timeZone="America/Sao_Paulo" />
                </p:inputText>

            </p:panelGrid>

        </p:fieldset>

        <p:fieldset legend="Dados do Autor">

            <p:outputLabel value="Selecione Autor:" for="autor" />
            <h:selectOneMenu value="#{livroBean.autorId}" id="autor">
                <f:selectItems value="#{livroBean.autores}" var="autor"
                    itemLabel="#{autor.nome}" itemValue="#{autor.id}" />
            </h:selectOneMenu>
            <h:commandButton value="Gravar Autor"
                action="#{livroBean.gravarAutor}">
                <f:ajax execute="autor" render="tabelaAutores" />
            </h:commandButton>

            <br />

            <h:commandLink value="Cadastrar novo autor"
                action="#{livroBean.formAutor}" immediate="true" />

            <h:dataTable value="#{livroBean.autoresDoLivro}" var="autor"
                id="tabelaAutores">
                <h:column>
                    <h:outputText value="#{autor.nome}" />
                </h:column>
                <h:column>
                    <h:commandLink value="X" action="#{livroBean.removerAutorDoLivro(autor)}"/>
                </h:column>
            </h:dataTable>
        </p:fieldset>
        <h:commandButton value="Gravar" action="#{livroBean.gravar}">
            <f:ajax execute="@form" render="@form :formTabelaLivros:tabelaLivros" />
        </h:commandButton>
    </h:form>
    <!-- tabela de exibição dos autores -->
</ui:define>
```

Máscara de input

Agora, temos três campos que precisam de um tratamento especial, o **ISBN**, que tem um formato específico, logo poderíamos colocar uma máscara para facilitar o usuário; o **Preço**, pois não podem ser digitadas letras nesse campo e a **Data de Lançamento**, pois queremos colocar um calendário para auxiliar o usuário.

Olhando o [*ShowCase* de Inputs][1] do Primefaces, vemos que nele há um [InputMask][2], um [InputNumber][3] e um [Calendar][4], que atendem aos nossos campos. Vamos começar pelo `InputMask`, que se chama `<p:inputMask>` e tem um atributo chamado `mask`, no qual definimos a máscara que queremos, no nosso caso será **999-9-99-999999-9**, que é o padrão do ISBN, onde **9** representa um número qualquer. Então o `input` do ISBN ficará assim:

```
<p:outputLabel value="ISBN:" for="isbn" />
<p:inputMask id="isbn" value="#{livroBean.livro.isbn}"
    validator="#{livroBean.comecaComDigitoUm}" mask="999-9-99-999999-9" />
```

Repare que agora o `input` do ISBN só aceita números e num formato específico. Agora veremos o preço, o `InputNumber`, mas há um problema, ele só existe a partir da versão **5.3.4** e a última versão disponibilizada para a comunidade (no momento da criação deste curso) é a versão **5.3**, então infelizmente não podemos utilizá-lo (caso a versão **5.4** já esteja disponibilizada para download, você pode baixá-la e testar esse componente).

Então pularemos para o `Calendar`, veja que há vários modos de como utilizá-lo, vamos utilizar o **Popup**. Basta trocar `p:inputText` por `p:calendar` e adicionar o atributo `mask="true"`, pois também queremos uma máscara nesse campo:

```
<p:outputLabel value="Data de Lançamento:" for="dataLancamento" />
<p:calendar id="dataLancamento" value="#{livroBean.livro.dataLancamento.time}"
    pattern="dd/MM/yyyy" timeZone="America/Sao_Paulo" mask="true" />
```

Com isso, os dados do livros estão terminados, para completar faltam os dados do autor.