

Demo do Primefaces

Transcrição

Com o ambiente pronto e configurado, queremos estilizar a nossa aplicação, por exemplo, o ISBN tem um formato específico, então queremos colocar uma máscara no campo dele para facilitar o usuário a digitá-lo; colocar um calendário no campo de data de lançamento, entre outras funcionalidades... Mas aí você pode me falar que pode usar JavaScript, JQuery, CSS... Tudo isso para melhorar a aplicação. Mas aqui é um curso sobre JSF, a ideia é que um componente faça isso para nós!

O JSF por padrão só faz o básico "arroz com feijão", se quisermos algo a mais, precisamos utilizar componentes que estendem as funcionalidades do JSF. E uma das bibliotecas de componentes JSF que mais se destaca no mercado, é o **Primefaces** (<http://www.primefaces.org/>). Podemos acessar o seu site, e lá há uma sessão **Demo** (<http://www.primefaces.org/showcase/index.xhtml>), que demonstra todos os componentes disponíveis (e como utilizá-los) diretamente no navegador.

Mas antes de começar a experimentar esses componentes, vamos configurar o nosso projeto para utilizar o **Primefaces**. Primeiramente, temos que fazer o seu download, acessando esse [link](http://www.primefaces.org/downloads) (<http://www.primefaces.org/downloads>) e procurando pela sessão *Community Downloads*. Lá baixaremos a versão mais recente, que na data de criação deste curso, é a versão **5.3**. Feito o download, copie o arquivo `jar` para dentro da pasta `WebContent/WEB-INF/lib`.

Declaração do namespace

Pronto, agora podemos começar, e vamos começar pelo início, pela página `login.xhtml`. Como Primefaces é uma biblioteca de componentes, precisamos declarar o seu *namespace*, e o prefixo geralmente utilizado é o `p`, então vamos seguir esse padrão. Basta adicionar dentro da tag `<html>`, juntamente com os outros *namespaces*:

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

Os primeiros componentes do Primefaces

A ideia agora é utilizar cada vez mais os componentes do Primefaces. Vamos "primefacear" todo o conteúdo do formulário, começando pelo título "Login", ele passará a ser um `<p:outputPanel>`. Depois, o *fieldset* e *legend*, agora teremos um `<p:fieldset>`, que já tem um atributo `legend`, então podemos substituí-los. Após isso, a ideia é substituir todos os componentes padrões do JSF, que utilizam o prefixo `h`, pelo prefixo `p`, ou seja, o Primefaces não só define componentes novos, como também redefine os componentes padrões do JSF. A razão dele fazer isso é que o Primefaces dá "algo a mais" para esses componentes, pode ser que o componente já faça ajax, ou que já venha com um CSS aplicado, etc. Ou seja, não precisamos reaprender a API.

Então nosso formulário ficará assim:

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

<ui:define name="conteudo">
```

```

<p:messages globalOnly="true" />

<h:form id="login">
    <p:fieldset legend="Login">
        <h:panelGrid columns="3">

            <p:outputLabel value="Email:" for="email" />
            <p:inputText id="email" value="#{loginBean.usuario.email}" required="true">
                <f:passThroughAttribute name="type" value="email" />
            </p:inputText>
            <p:message for="email" id="messageEmail" />

            <p:outputLabel value="Senha:" for="senha" />
            <p:inputText id="senha" value="#{loginBean.usuario.senha}" required="true" >
                <f:passThroughAttribute name="type" value="password" />
            </p:inputText>
            <p:message for="senha" id="messageSenha" />

            <p:commandButton value="Efetue Login" action="#{loginBean.efetuaLogin}" />
        </h:panelGrid>
    </p:fieldset>
</h:form>
</ui:define>

```

Podemos acessar a página de login, e perceber que já há uma diferença notável na mesma. Podemos até efetuar login para ver se tudo continua funcionando normalmente... Mas não está! Ficamos clicando infinitamente no botão e nada acontece. O que está acontecendo?

Ajax por padrão

O botão do Primefaces já faz ajax por padrão, que não é um comportamento do `commandButton` padrão. Então, já que ele faz ajax por padrão, precisamos definir o que queremos submeter e quais partes da tela queremos renderizar, o que não estamos fazendo no momento, já que atualmente, quando clicamos no botão para efetuar login, só estamos submetendo o botão e não os campos do formulário. Então vamos mudar esse comportamento, para isso o Primefaces tem dois atributos, o `process` e o `update`. No `process` dizemos quais componentes queremos submeter, no nosso caso queremos submeter todo o formulário, então utilizaremos `@form`; e no `update` dizemos o que queremos atualizar na página, e no nosso caso também é o formulário, logo também terá como valor `@form`. Então o `commandButton` ficará assim:

```

<p:commandButton value="Efetue Login" action="#{loginBean.efetuaLogin}" update="@form" process='

```

Agora já podemos tentar fazer o login, mas opa... A nossa senha está aparecendo! Isso quer dizer que o `passThroughAttribute` não está funcionando, portanto vamos removê-lo e consultar novamente o [demo](http://www.eclipse.org/downloads/) (<http://www.eclipse.org/downloads/>) do Primefaces para ver se ele nos oferece algum componente para substituí-lo. E no menu **Input**, vemos que há um **Password**, que nos oferece diversas opções, vamos utilizar o `password` com `feedback`, e olhando como utilizá-lo, vemos que, no nosso caso, basta trocar `inputText` por `password` e adicionar o atributo `feedback="true"`:

```

<p:password id="senha" value="#{loginBean.usuario.senha}" feedback="true" required="true" />

```

Com isso, tudo volta a funcionar corretamente. Então o primeiro passo está concluído, o formulário de login já está utilizando o Primefaces. Nos próximos capítulos vamos focar nos outros formulários, então até lá!

O que aprendemos?

- Objetivo dos Primefaces é estender os componentes JSF
- Como importar a lib Primefaces
- Primefaces redefine todos os componentes da especificação JSF (por exemplo `h:inputText` se torna `p:inputText`)
- Usar os primeiros componentes específicos como `p:fieldset` ou `p:password`
- Comandos do Primefaces usa AJAX por padrão