

Artigo disponível em: <http://www.devmedia.com.br/javaxserver-faces-2-2-aumento-de-produtividade/30282>

JavaServer Faces 2.2: aumento de produtividade

Component Tags do JSF 2

JSF 2 possui uma variedade de componentes. Cada componente possui por trás uma classe `UIComponent`. Esses componentes são exibidos como elementos HTML.

Veremos neste artigo os componentes mais utilizados, os novos componentes adicionados e como eles são exibidos em HTML.

1. Componente `h:commandButton`

Exemplo de Uso:

```
<h:commandButton value="Próximo" action="proximo"/>
```

Saída HTML:

```
<input type="submit" name="j_idt8" value="Próximo" />
```

2. Componente `h:commandLink`

Exemplo de Uso:

```
<h:commandLink value="meuLink" action="proximo"/>
```

Saída HTML:

```
<script type="text/javascript"
src="/components/faces/javax.faces.resource/jsf.js?ln=javax.faces&stag
e=Development">
</script>
<a href="#"
onclick="mojarra.jsfcljs(document.getElementById('j_idt6'),
{'j_idt10':'j_idt10'}, '');return false">
meuLink
</a>
```

3. Componente h:dataTable

Exemplo de Uso:

```
<h:dataTable value="#{meuBean.lista}" var="p">
    <h:column>
        <f:facet name="header">Id</f:facet>
        #{p.id}
    </h:column>
    <h:column>
        <f:facet name="header">Nome</f:facet>
        #{p.nome}
    </h:column>
</h:dataTable>
```

Saída HTML:

```
<table>
    <thead>
        <tr>
            <th scope="col">Id</th>
            <th scope="col">Nome</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td> 1 </td>
            <td> João </td>
        </tr>
        <tr>
            <td> 2 </td>
            <td> Carlos </td>
        </tr>
        <tr>
            <td> 3 </td>
            <td> Francisco </td>
        </tr>
    </tbody>
</table>
```

4. Componente h:form

Exemplo de Uso:

```
<h:form> ... </h:form>
```

Saída HTML:

```
<form id="j_idt6" name="j_idt6" method="post"
      action="/components/faces/index.xhtml"
      enctype="application/x-www-form-urlencoded">
    ...
</form>
```

5. Componente h:graphicImage

Exemplo de Uso:

```
<h:graphicImage value="/images/teste.png"/>
```

Saída HTML:

```

```

6. Componente h:inputHidden

Exemplo de Uso:

```
<h:inputHidden value="valorEscondido"/>
```

Saída HTML:

```
<input type="hidden" name="j_idt22" value="valorEscondido" />
```

7. Componente h:inputSecret

Exemplo de Uso:

```
<h:inputSecret value="#{meuBean.senha}"/>
```

Saída HTML:

```
<input type="senha" name="j_idt24" value="" />
```

8. Componente h:inputText

Exemplo de Uso:

```
<h:inputText id="meuInputTextExemplo" value="#{meuBean.inputText}"/>
```

Saída HTML:

```
<input id="meuInputTextExemplo" type="text" name="meuInputTextExemplo" />
```

9. Componente h:inputTextArea

Exemplo de Uso:

```
<h:inputTextarea value="#{meuBean.inputTextarea}"/>
```

Saída HTML:

```
<textarea name="j_idt27">
  </textarea>
```

10. Componente h:button

Exemplo de Uso:

```
<h:button value="meuButton" outcome="next"/>
```

Saída HTML:

```
<input type="button"
onclick="window.location.href='/components/faces/next.xhtml'; return
false;" value="meuButton" />
```

11. Componente h:link

Exemplo de Uso:

```
<h:link value="Próximo" outcome="proximo">
    <f:param name="teste" value="testeValor"/>
</h:link>
```

Saída HTML:

```
<a href="/components/faces/next.xhtml?teste=testeValor">
    Próximo
</a>
```

12. Componente h:body

Exemplo de Uso:

```
<h:body>
    Teste do Corpo
</h:body>
```

Saída HTML:

```
<body>
    Teste do Corpo
</body>
```

13. Componente h:outputFormat

Exemplo de Uso:

```
<h:outputFormat value="Olá {0}">
    <f:param value="João"/>
</h:outputFormat>
```

Saída HTML:

Olá João

14. Componente h:outputLabel

Exemplo de Uso:

```
<h:outputLabel for="inputTextId" value="meuLabelExemplo"/>
<h:inputText id="inputTextId" value="meuInputTextExemplo"/>
```

Saída HTML:

```
<label for="inputTextId">
    meuLabelExemplo
</label>
<input id="inputTextId" type="text" name="inputTextId"
value="meuInputTextExemplo" />
```

15. Componente h:outputLink

Exemplo de Uso:

```
<h:outputLink value="proximo.xhtml">
    meuOutputExemplo
</h:outputLink>
```

Saída HTML:

```
<a href="proximo.xhtml">
    meuOutputExemplo
</a>
```

16. Componente h:outputText

Exemplo de Uso:

```
<h:outputText value="meuOutputText" />
```

Saída HTML:

```
meuOutputText
```

17. Componente h:outputScript

Exemplo de Uso:

```
<h:outputScript library="scripts" name="main.js" />
```

Saída HTML:

```
<script type="text/javascript"
src="/components/faces/javax.faces.resource/main.js">
```

18. Componente h:outputStylesheet

Exemplo de Uso:

```
<h:outputStylesheet library="stylesheets" name="main.css" />
```

Saída HTML:

```
<link type="text/css" rel="stylesheet" href
="/components/faces/javax.faces.resource/main.css?ln=stylesheets" />
```

19. Componente h:panelGrid

Exemplo de Uso:

```
<h:panelGrid columns="2">
    <h:outputLabel for="inputText1" value="meuLabel1"/>
    <h:inputText id="inputText1" value="#{meuBean.inputText}"/>
    <h:outputLabel for="inputText2" value="meuLabel2"/>
    <h:inputText id="inputText2" value="#{meuBean.inputText}"/>
</h:panelGrid>
```

Saída HTML:

```
<table>
    <tbody>
        <tr>
            <td>
                <label
for="inputText1">meuLabel1</label>
            </td>
            <td>
                <input
type="text" name="inputText1" />
            </td>
        </tr>
        <tr>
            <td>
                <label
for="inputText2">meuLabel2</label>
            </td>
            <td>
                <input
type="text" name="inputText2" />
            </td>
        </tr>
    </tbody>
</table>
```

20. Componente h:selectBooleanCheckbox

Exemplo de Uso:

```
<h:selectBooleanCheckbox value="#{meuBean.selectBooleanCheckbox}"/>
```

Saída HTML:

```
<input type="checkbox" name="j_idt52" />
```

21. Componente h:selectOneRadio

Exemplo de Uso:

```
<h:selectOneRadio value="#{meuBean.selectBooleanCheckbox}">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectOneRadio>
```

Saída HTML:

```
<table>
    <tr>
        <td>
            <input type="radio" checked="checked"
name="j_idt54" id="j_idt54:0" value="branco" />
            <label for="j_idt54:0">Branco</label>
        </td>
        <td>
            <input type="radio" checked="checked"
name="j_idt54" id="j_idt54:1" value="preto" />
            <label for="j_idt54:1">Preto</label>
        </td>
        <td>
            <input type="radio" checked="checked"
name="j_idt54" id="j_idt54:2" value="azul" />
            <label for="j_idt54:2">Azul</label>
        </td>
    </tr>
</table>
```

22. Componente h:selectOneListbox

Exemplo de Uso:

```
<h:selectOneListbox value="#{meuBean.selectBooleanCheckbox}">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectOneListbox>
```

Saída HTML:

```
<select name="j_idt59" size="3">
    <option value="branco" selected="selected">Branco</option>
    <option value="preto" selected="selected">Preto</option>
    <option value="azul" selected="selected">Azul</option>
</select>
```

23. Componente h:selectOneMenu

Exemplo de Uso:

```
<h:selectOneMenu value="#{meuBean.selectBooleanCheckbox}">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectOneMenu>
```

Saída HTML:

```
<select name="j_idt64" size="1">
    <option value="branco" selected="selected">Branco</option>
    <option value="preto" selected="selected">Preto</option>
    <option value="azul" selected="selected">Azul</option>
</select>
```

24. Componente h:selectManyCheckbox

Exemplo de Uso:

```
<h:selectManyCheckbox value="#{meuBean.selectManyCheckbox}">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectManyCheckbox>
```

Saída HTML:

```
<table>
    <tr>
        <td>
            <input
value="branco" type="checkbox" />
            <label
class="">Branco</label>
        </td>
        <td>
            <input
value="preto" type="checkbox" />
            <label
class="">Preto</label>
        </td>
        <td>
            <input
value="azul" type="checkbox" />
            <label
class="">Azul</label>
        </td>
    </tr>
</table>
```


25. Componente h:selectManyListbox

Exemplo de Uso:

```
<h:selectManyListbox value="#{meuBean.selectBooleanCheckbox}"
size="2">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectManyListbox>
```

Saída HTML:

```
<select name="j_idt74" size="2">
    <option value="branco" selected="selected">Branco</option>
    <option value="preto" selected="selected">Preto</option>
    <option value="azul" selected="selected">Azul</option>
</select>
```

26. Componente h:selectManyMenu

Exemplo de Uso:

```
<h:selectManyMenu value="#{meuBean.selectBooleanCheckbox}">
    <f:selectItem itemValue="branco" itemLabel="Branco"/>
    <f:selectItem itemValue="preto" itemLabel="Preto"/>
    <f:selectItem itemValue="azul" itemLabel="Azul"/>
</h:selectManyMenu>
```

Saída HTML:

```
<select name="j_idt81" multiple="multiple" size="1">
    <option value="red">Branco</option>
    <option value="green">Preto</option>
    <option value="blue">Azul</option>
</select>
<input type="hidden" name="javax.faces.ViewState"
id="j_idl:javax.faces.ViewState:0" value="-
6389029500114305749:3536902027553202703" autocomplete="off" />
```

27. Outros Componentes

A tag h:inputFile é um novo componente adicionado no JSF 2.2. Este componente permite que um arquivo seja carregado.

Este componente pode ser usado dentro de um h:form e ligado a um bean que tem um campo do tipo javax.servlet.http.Part.

Segue na Listagem 19 um exemplo

Listagem 19. Exemplo de um formulário utilizando h:inputFile.

```
<h:form enctype="multipart/form-data">
    <h:inputFile value="#{fileUploadBean.file}"/><br/>
    <h:commandButton value="Carregar Arquivo"/><p/>
</h:form>
```

Neste código temos um formulário HTML que é criado através do elemento `h:form`. O formulário deve ter como valor `"multipart/form-data"` no atributo `enctype`. Este é um tipo MIME padrão que deveria ser usado para submeter formulários que contêm arquivos, dados que não sejam ASCII e dados binários.

No código acima também temos o atributo `value` que é ligado ao bean. Um exemplo de um bean para o código acima é dado na Listagem 20.

Listagem 20. Exemplo de um bean para carregar o arquivo.

```
@Named
@RequestScoped
public class FileUploadBean {
    private Part file;
    public Part getFile() {
        return file;
    }

    public setFile(Part file) {
        this.file = file;
    }
}
```

O arquivo será carregado quando o botão “Carregar Arquivo” for clicado.

O upload do arquivo também pode ser empacotado em uma tag .

Segue o exemplo na Listagem 21.

Listagem 21. Exemplo utilizando `f:ajax`.

```
<h:form enctype="multipart/form-data" prependId="false">
    <h:inputFile value="#{fileUploadBean.file}"/><br/>
    <h:commandButton value="Upload">
        <f:ajax onevent="statusUpdate"/>
    </h:commandButton>
</h:form>
<textarea id="status" cols="40" rows="10" readonly="readonly"/>
```

Este código é similar ao código mostrado anteriormente com a diferença que o fora do `h:form` é um espaço reservado para a exibição do status do upload do arquivo. Além disso, o `h:commandButton` tem uma tag aninhada. Esta tag invoca o método `statusUpdate` definido separadamente num JavaScript definido, conforme a Listagem 22.

Listagem 22. Exemplo de JavaScript para mostra o status.

```
<br>var statusUpdate = function statusUpdate(data) {<br>    var statusArea = document.getElementById("status");<br>    var text = statusArea.value;<br>    text = text + "Nome: " + data.source.id;<br><br>    if (data.type === "event") {<br>        text = text + " Event: " + data.status + " ";<br>    }<br>    statusArea.value = text;<br>}<br>
```

As últimas versões do JSF têm aumentado significativamente a produtividade das equipes de desenvolvimento, tornando mais simples e prático desenvolver em cima dessa plataforma. A última versão do JSF, a versão 2.2 em especial, teve um incremento de funcionalidades e uma atualização significativa em alguns pontos de diversos outros componentes para fornecer uma maior produtividade aos desenvolvedores.

Bibliografia

[1]Josh Juneau. Java EE 7 Recipes: A Problem-Solution Approach. Apress, 2013.

[2]Josh Juneau. Introducing Java EE 7: A Look at What's New. Apress, 2013.

[3]Arun Gupta. Java EE 7 Essentials. O'Reilly, 2013.