



Desenvolvimento para Servidores-II

Tabela de Dados

Neste tópico abordaremos a tag `<h:dataTable>` do JSF que nos permite manipular dados em uma tabela.

Prof. Ciro Cirne Trindade

A tag `h:dataTable`

- A tag `<h:dataTable>` é usada para mostrar uma coleção de dados em uma tabela HTML
- É normalmente utilizada para exibir objetos armazenados em instâncias de `java.util.List` ou `java.sql.ResultSet`

Forma geral

```
<h:dataTable value="#{items}" var="item">
  <h:column>
    <!-- componentes da 1ª coluna -->
    #{item.aPropertyName}
  </h:column>
  <h:column>
    <!-- componentes da próxima coluna -->
    #{item.anotherPropertyName}
  </h:column>
  <!-- adicione mais colunas, se desejar -->
</h:dataTable>
```

Exemplo: tabela de pessoas

- Aplicação aonde é possível cadastrar e depois listar e excluir pessoas
- Uma pessoa é representada por apenas seu nome, sobrenome e e-mail

Classe Pessoa (1/2)

```
package model;
```

```
public class Pessoa {  
    private String nome;  
    private String sobrenome;  
    private String email;  
  
    public Pessoa() {  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
}
```

Classe Pessoa (2/2)

```
    public String getSobrenome() {  
        return sobrenome;  
    }  
  
    public void setSobrenome(String sobrenome) {  
        this.sobrenome = sobrenome;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
  
}
```

Managed bean

PessoaBackBean (1/2)

```
package beans;
```

```
import java.util.ArrayList;
import java.util.List;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;
import model.Pessoa;
```

```
@ManagedBean
```

```
@SessionScoped
```

```
public class PessoaBackBean {
    private Pessoa pessoa = new Pessoa();
    private List<Pessoa> pessoas = new ArrayList<>();

    public PessoaBackBean() { }
    public Pessoa getPessoa() { return pessoa; }
    public void setPessoa(Pessoa pessoa) {
        this.pessoa = pessoa;
    }
}
```



Managed bean

PessoaBackBean (2/2)

```
public List<Pessoa> getPessoas() {  
    return pessoas;  
}  
public void setPessoas(List<Pessoa> pessoas) {  
    this.pessoas = pessoas;  
}  
  
public String cadastrarPessoa() {  
    pessoas.add(pessoa);  
    pessoa = new Pessoa();  
    return null;  
}  
  
}
```


_template.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=UTF-8" />
    <h:outputStylesheet name="./css/default.css"/>
    <h:outputStylesheet name="./css/cssLayout.css"/>
    <title><ui:insert name="title">
      title</ui:insert></title>
  </h:head>
  <h:body>
    <div id="top" class="top">
      <ui:insert name="top">Top</ui:insert>
    </div>
    <div id="content" class="center_content">
      <ui:insert name="content">Content</ui:insert>
    </div>
  </h:body>
</html>
```

index.xhtml (1/2)

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<ui:composition xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    template="._template.xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:p="http://xmlns.jcp.org/jsf/passthrough"
    xmlns="http://www.w3.org/1999/xhtml">

    <ui:define name="title">
        Cadastrar Pessoa
    </ui:define>

    <ui:define name="top">
        <h1>Cadastrar Pessoa</h1>
    </ui:define>

    <ui:define name="content">
        <h:form>
            <h:panelGrid columns="3">
```

Nome :

```
<h:inputText id="nome" label="nome"
    value="#{pessoaBackBean.pessoa.nome}"
    required="true"/>
```

```
<h:message for="nome"/>
```

Sobrenome:

```
<h:inputText id="sobrenome" label="sobrenome"
    value="#{pessoaBackBean.pessoa.sobrenome}"/>
```

```
<h:message for="sobrenome"/>
```

E-mail:

```
<h:inputText p:type="email" id="email"
    label="e-mail"
    value="#{pessoaBackBean.pessoa.email}"/>
```

```
<h:message for="email"/>
```

```
<h:commandButton value="Cadastrar"
    action="#{pessoaBackBean.cadastrarPessoa()}"/>
```

```
<h:commandLink value="Listar pessoas"
    action="/pessoas" immediate="true"/>
```

```
</h:panelGrid>
```

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

```
</ui:define>
```

```
</ui:composition>
```

peessoas.xhtml (1/2)

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<ui:composition xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    template="._template.xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns="http://www.w3.org/1999/xhtml">
    <ui:define name="title">
        Pessoas cadastradas
    </ui:define>

    <ui:define name="top">
        <h1>Pessoas Cadastradas</h1>
    </ui:define>

    <ui:define name="content">
        <h:dataTable value="#{pessoaBackBean.pessoas}"
            var="p">
```

pessoas.xhtml (1/2)

```
<h:column>
    <h:outputText value="#{p.nome}" />
</h:column>
<h:column>
    <h:outputText value="#{p.sobrenome}" />
</h:column>
<h:column>
    <h:outputText value="#{p.email}" />
</h:column>
</h:dataTable>
<h:button value="Voltar" outcome="/index"/>
</ui:define>

</ui:composition>
```

Cabeçalhos, rodapés e títulos

- Cabeçalhos e rodapés de colunas são especificados através da tag `f:facet` no corpo de um `h:column`
- O título da tabela também é especificado através da tag `f:facet`, mas no corpo da tag `h:dataTable`

Exemplo de título, cabeçalho e rodapé

```
<h:dataTable value="#{pessoasBackBean.pessoas}"
              var="p">
  <f:facet name="caption">Pessoas</f:facet>
  ...
  <h:column>
    <f:facet name="header">Nome</f:facet>
    <h:outputText value="#{p.nome}" />
    <f:facet name="footer">Quantidade:
      #{pessoasBackBean.pessoas.size()}
    </f:facet>
  </h:column>
  ...
</h:dataTable>
```

Título da tabela

Cabeçalho da
coluna

Rodapé da
coluna

- A tag `h:dataTable` possui atributos que permitem definir estilos CSS para:
 - A tabela como um todo: `styleClass`
 - Cabeçalhos e rodapés: `headerClass` e `footerClass`
 - Colunas: `columnsClasses`
 - Linhas: `rowsClasses`

Exemplo de estilos (1/2)

```
<h:dataTable value="#{tabelaPessoas.pessoas}"  
    var="p" styleClass="pessoas"  
    captionClass="pessoasCaption"  
    headerClass="pessoasHeader"  
    rowClasses="linhaPar, linhaImpar">  
  
    ...  
</h:dataTable>
```

Exemplo de estilos (CSS) (2/2)

```
.pessoas {  
    border: thin solid black;  
}  
.pessoasHeader {  
    text-align: center;  
    font-style: italic;  
    color: Snow;  
    background: Teal;  
}  
.pessoasCaption {  
    text-align: center;  
    font-style: italic;  
    color: Snow;  
    background: Teal;  
}  
.linhaPar {  
    text-align: left;  
    background: MediumTurquoise;  
}  
.linhaImpar {  
    text-align: left;  
    background: PowderBlue;  
}
```

Removendo uma linha

- É possível remover uma linha da tabela passando um método no atributo `action` de um componente
- Exemplo:

```
<h:commandLink value="Excluir"  
    action="#{pessoaBackBean.deleteRow(p)}/>
```

```
public String deleteRow(Pessoa personToDelete) {  
    pessoas.remove(personToDelete);  
    return null;  
}
```

Referências

- GEARY, David; HORSTMANN, Cay. *Core JavaServer Faces*. 3. ed., Prentice-Hall, 2010.
- ORACLE Corporation. *The Java EE 7 Tutorial*. Disponível em: <https://docs.oracle.com/javaee/7/JEETT.pdf>, 2014.