

# YourCodeLab.com



## Utilização de Ajax

Prof. Msc. Alexssander Siqueira



# Conteúdo da Aula

- Definição de Ajax
- Bibliotecas Ajax
- Utilização de componentes JSF



# Definição de Ajax

- *Asynchronous Javascript and XML* (Ajax), é uma tecnologia que utiliza objetos *HttpRequest* para realizar chamadas assíncronas a partir do navegador.
- Logo, o usuário possui uma experiência mais rica devido as atualizações e renderizações de dados em tela de forma dinâmica.



# Bibliotecas Ajax

- A definição de JSF não contempla a integração de chamadas Ajax em seus componentes.
- Logo, surgiram frameworks com esse propósito:
  - ✓ Richfaces
  - ✓ Primefaces
  - ✓ Icefaces
- Esta disciplina utiliza o **Primefaces** por ser o mais popular.



# Utilização de Ajax

- O componente `<h:inputText>` está ligado a um campo de texto, o qual servirá como base para execução da Action de consulta.

```
<h:inputText id="searchName" styleClass="form-control"
    value="#{customerManagedBean.customer.name}">
    <p:ajax event="keyup"
        listener="#{customerManagedBean.searchByNameAction}"
        update="customerTable" process="@all" />
</h:inputText>
```



# Utilização de Ajax

- O componente `<p:ajax>` é acionado quando uma tecla é pressionada (**event = "keyup"**). O atributo *listener* indica a Action que será invocada (**searchByNameAction**).

```
<h:inputText id="searchName" styleClass="form-control"
    value="#{customerManagedBean.customer.name}">
    <p:ajax event="keyup"
        listener="#{customerManagedBean.searchByNameAction}"
        update="customerTable" process="@all" />
</h:inputText>
```



# Utilização de Ajax

- Por fim, deve-se indicar qual componente da tela será atualizado a cada busca (**update = "customerTable"**).
- O atributo *process* define a cadeia de invocação das chamadas de Ajax. O valor *@all* garante a execução plena.

```
<h:inputText id="searchName" styleClass="form-control"
    value="#{customerManagedBean.customer.name}">
    <p:ajax event="keyup"
        listener="#{customerManagedBean.searchByNameAction}"
        update="customerTable" process="@all" />
</h:inputText>
```



# Utilização de Ajax

- O componente `<h:dataTable>` que será atualizado precisa estar contido pelo componente `<p:panel>` `</p:panel>` onde o atributo **id** deve ser equivalente ao informado no slide anterior.

```
<p:panel id="customerTable">
  <h:dataTable styleClass="table table-striped"
    <h:column>
      <f:facet name="header">
        <h:outputLabel value="Id"/>
      </f:facet>
      <h:outputText value="#{c.id}"/>
    </h:column>
    <h:column>
      <f:facet name="header">
        <h:outputLabel value="Name"/>
      </f:facet>
      <h:outputText value="#{c.name}"/>
    </h:column>
  </p:panel>
```





# Utilização de Ajax

- O código da Action `searchByNameAction` é bem simples, onde o nome digitado é buscado na base de dados através de uma chamada de serviço.

```
public void searchByNameAction() {  
    System.out.println("Searching...");  
    this.listCustomer = customerService.findCustomerByName(customer.getName());  
}
```



## Referência

- No projeto YourJSFExamples é possível visualizar este código no Branch version\_03.

[https://github.com/YourCodeLabExamples/YourJSF/tree/version\\_03](https://github.com/YourCodeLabExamples/YourJSF/tree/version_03)

YourCodeLab.com

Questions???

alexssander\_as@hotmail.com