



**PUCPR**  
GRUPO MARISTA



---

# Web Client: AJAX & JSF

Disciplina de ADS: **Tecnologia para Desenvolvimento Web**



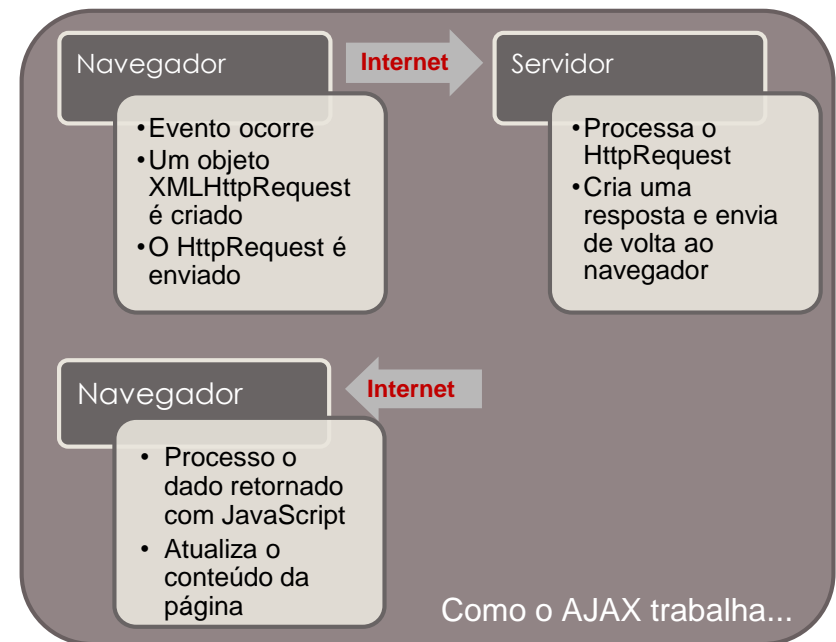
# Sumário – Web Client: AJAX

- AJAX
- JSF Tags para AJAX
- JSF e AJAX – Exemplo Simples
  - Criar Projeto
  - Criar e Editar JSF Page
  - Criar e Editar Managed Bean



# AJAX

- Sigla para **Asynchronous JavaScript And XML**, ou JavaScript e XML assíncrono
- Técnica para criar aplicações Web mais rápidas e interativas
- Permite que as **alterações ocorram em uma página da Web sem a necessidade da completa atualização da página.**
- O JavaScript é usado para implementar componentes e **funcionalidade AJAX** em páginas Web



# Usando AJAX com JSF

## ➤ tag f:ajax

- O Ajax é uma técnica do JavaScript para para **enviar e receber dados do servidor** de forma **assíncrona**, permitindo atualizar **partes da página da web**, sem recarrega total
- Em um aplicações **Facelets**, podemos usar a Tag **<f:ajax>** para **adicionar a funcionalidade Ajax** a qualquer componente UI da interface, **sem codificação ou configuração adicionais**.
- No exemplo a seguir, o **comportamento Ajax** é adicionado a um componente de entrada incluindo a **tag core** **f: ajax**:

```
<h:inputText value="#{bean.message}">  
    <f:ajax />  
</h:inputText>
```

# JSF Tags para AJAX – alguns exemplos

Tag <f: ajax> & Descrição
<b>Execute:</b> Uma lista de Ids, separados por espaços, para componentes que devem ser incluídos na solicitação do Ajax.
<b>Disable:</b> Se verdade, o comportamento do Ajax será aplicado a qualquer componente pai ou filho. Se falso, o comportamento do Ajax será desativado.
<b>Listener:</b> Uma expressão EL para um método em um bean de apoio a ser chamado durante a solicitação do Ajax.
<b>Render:</b> Uma lista de IDs separados por espaços para componentes que serão atualizados após um request.Render do Ajax
<b>Onerror:</b> O nome de uma função de retorno de chamada JavaScript que será chamada se houver um erro durante a solicitação do Ajax.

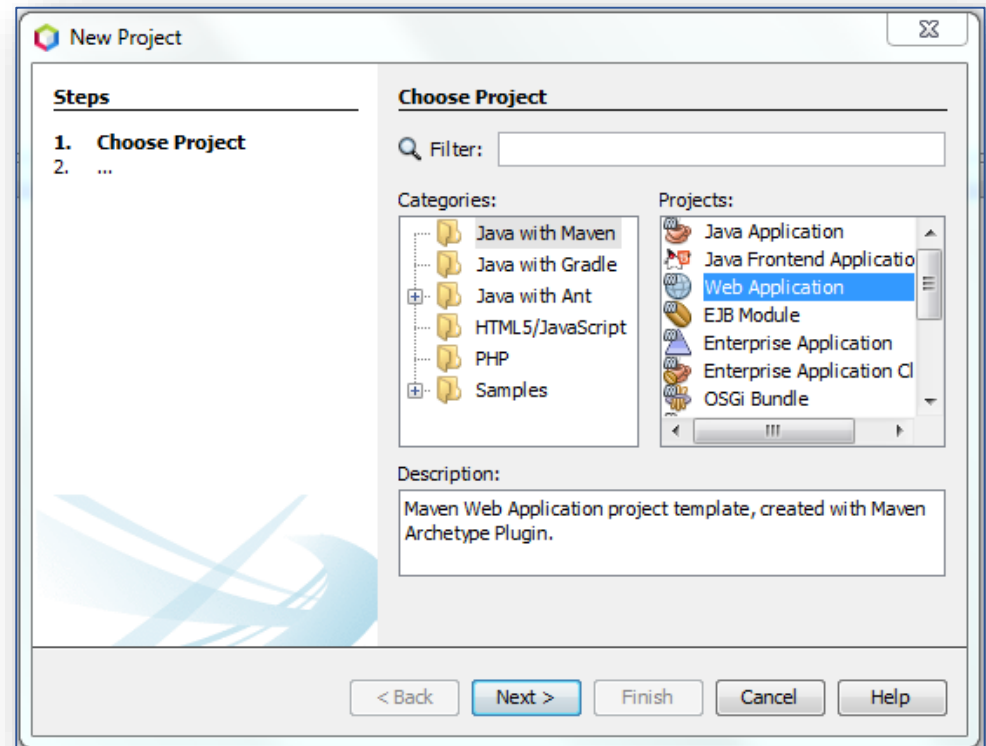
# JSF e AJAX – Exemplo Simples

## ➤ Tutorial Hello World

- No **JSF**, codificar o **AJAX** é como codificar uma tag HTML normal, ou seja, bem simples.
- Neste tutorial, apresentamos o exemplo de **hello world**, com um botão que, quando clicado, faz **uma solicitação Ajax** em vez de enviar o todo o formulário.

## ➤ Primeira Ação = Criar Projeto

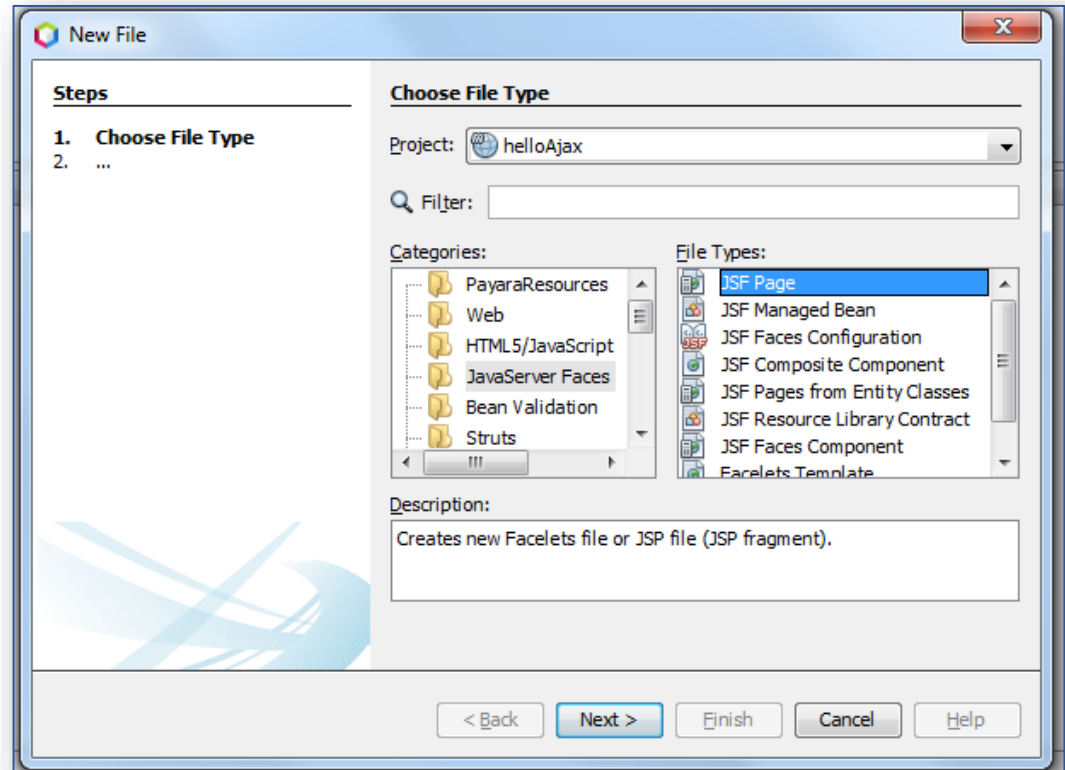
- Na barra de menu, vá em **File > New Project** e crie um aplicação **Java with Maven > Web Application**
- Digite no nome do projeto **helloAjax**



# JSF e AJAX – Exemplo Simples

## ➤ Segunda Ação = Criar JSF Page

- Com o botão direito do mouse sobre o projeto **helloAjax**, crie um novo **File > New > Other ...** E escolha **JavaServer Faces > JSF Page**
- Clique **Next**
- Digite no nome do arquivo **helloAjax.xhtml**



# JSF e AJAX – Exemplo Simples

## ➤ Terceira Ação = Alterar a JSF Page

- Altere o código fonte do arquivo criado, como indicado ao lado →

Inclui os componentes do JSF para suporte ao AJAX

Botão JSF habilitado para chamar o AJAX

Quando o botão é clicado, ele fará uma solicitação Ajax para o servidor, em vez de enviar o formulário inteiro.

### helloAjax.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core">
  <h:body>
    <h3>JSF 2.0 + Ajax Hello World Example</h3>
    <h:form>
      <h:inputText id="name"
        value="#{helloBean.name}"></h:inputText>
      <h:commandButton value="Welcome Me">
        <f:ajax execute="name" render="output" />
      </h:commandButton>
      <h2><h:outputText id="output"
        value="#{helloBean.sayWelcome}" /></h2>
    </h:form>
  </h:body>
</html>
```



# JSF e AJAX – Exemplo Simples

## ➤ Detalhe do helloAjax.html

### ➤ Na tag `<f:ajax>`:

`execute = "name"` - Indica que o componente `form` com `id = "name"` será enviado para o servidor para processamento.

`render = "output"` - Após o pedido do Ajax, ele atualizará o componente com um `id = "output"`.

Após a conclusão da solicitação do Ajax, o componente `<h:outputText>` com o `id = "output"` será atualizado.

## helloAjax.xhtml - detalhe

```
...  
<h:commandButton value="Welcome Me">  
    <f:ajax execute="name" render="output" />  
</h:commandButton>  
<h:outputText id="output" value="#{helloBean.sayWelcome}" />  
...
```

Chamada ao **Managed Bean** correspondente, feita pelo Ajax, quando é feita a renderização do `id = "output"`.

# JSF e AJAX – Exemplo Simples

- **Quarta Ação = Criar o Managed Bean**
  - Com o botão direito do mouse sobre o projeto **helloAjax**, crie um novo **File > New > Other ...** E escolha **JavaServer Faces > JSF Managed Bean**
  - Configure como indicado ao lado ➔
  - Clique **Finish**

**New JSF Managed Bean**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

☐ Add data to configuration file

Configuration File:

Name:

Scope:

< Back   Next >   Finish   Cancel   Help

# JSF e AJAX – Exemplo Simples

## ➤ Quinta Ação = Alterar o JSF Managed Bean

- Altere o código fonte do arquivo criado, como indicado ao lado →

Quando o botão no **helloAjax.xhtml** é clicado, é feita uma solicitação **Ajax** que passa o valor da caixa de texto para o processamento do servidor.

Depois disso, o componente **outputText** no **helloAjax.xhtml** é atualizado com o valor enviado pelo método **getSayWelcome ()**, sem a “atualização completa de página”.

### Controller.helloBean.xhtml

```
package controller;

import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import java.io.Serializable;

@Named(value = "helloBean")
@SessionScoped
public class helloBean implements Serializable {

    private String name;

    public helloBean() {
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSayWelcome() {
        //check if null?
        if ("".equals(name) || name == null) {
            return "";
        } else {
            return "Ajax message : Welcome " + name;
        }
    }

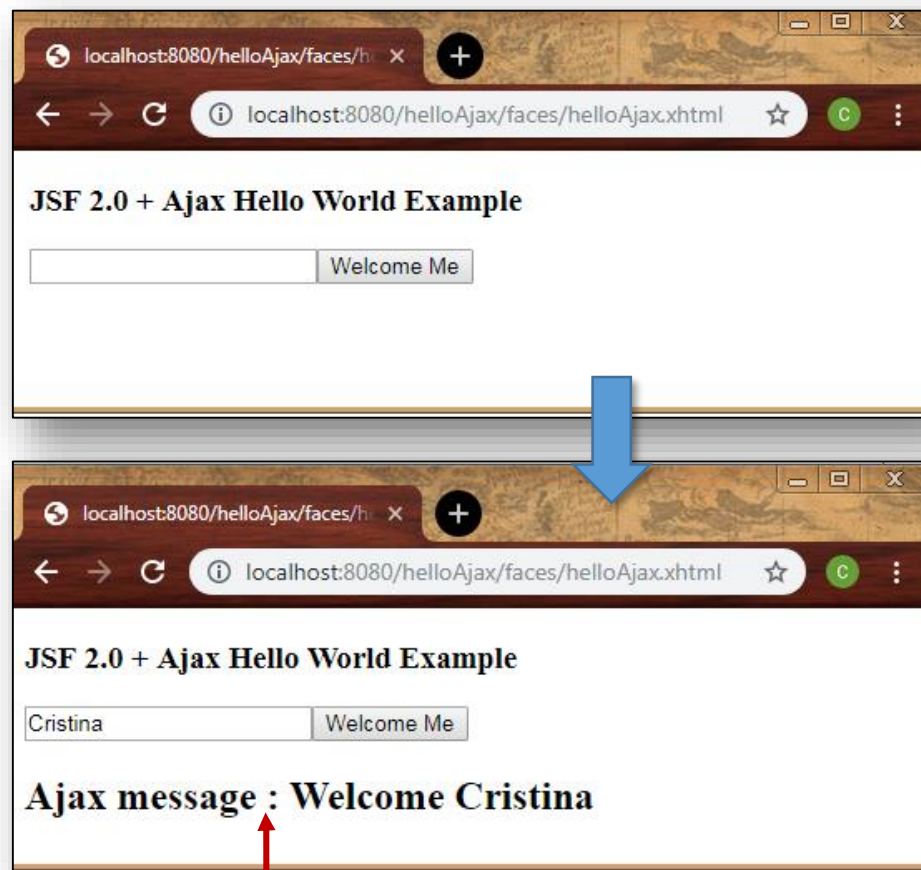
}
```

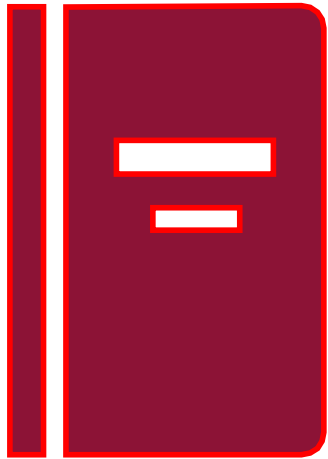
# JSF e AJAX – Exemplo Simples

## ➤ Sexta Ação = Executar a JSF Page

- Com o botão direito do mouse sobre o projeto **helloAjax**, selecione o **Clean and Build**
- Com o botão direito do mouse sobre o **helloAjax.xhtml**, selecione **Run File**

Mensagem retornada pelo AJAX de forma assíncrona: apenas um **<div>** da página web foi atualizado.





# Referências

# Referências

- JSF 2.0 + Ajax hello world example
  - <https://www.mkyong.com/jsf2/jsf-2-0-ajax-hello-world-example/>
- Using Ajax Functionality with JavaServer Faces Technology
  - <https://javaee.github.io/tutorial/jsf-ajax002.html>
- JSF - Ajax
  - [https://www.tutorialspoint.com/jsf/jsf\\_ajax.htm](https://www.tutorialspoint.com/jsf/jsf_ajax.htm)



**PUCPR**  
GRUPO MARISTA

**PROFESSOR CONTEUDISTA**

**Cristina Verçosa Pérez Barrios de Souza, Profa. Dra.**

© PUCPR. Todos os direitos reservados.

Nenhum texto pode ser reproduzido sem prévia autorização.