

# Desenvolvimento Web II

## Aula 13 - Introdução ao Primefaces

# Apresentação

---

Olá, seja bem-vindo(a) à nossa aula sobre o Primefaces. O Primefaces é um conjunto de componentes que poderá deixar suas aplicações mais bonitas e rápidas, uma vez que possui Ajax embutido. Consequentemente, os sistemas de informações desenvolvidos com o JSF e Primefaces serão mais fáceis de serem utilizados fornecendo aos usuários uma boa experiência de utilização.

Nesta aula, conheceremos o que é o Primefaces e como configurá-lo em uma aplicação JSF. Por fim, criaremos uma aplicação básica para sintetizar o conhecimento adquirido nesta aula.

## Objetivos

- Conhecer a suíte de componentes do Primefaces;
- Aprender como utilizar o Primefaces em uma aplicação JSF.

# O que é Primefaces?

O Primefaces é uma suíte de componentes de código aberto com várias extensões para serem utilizados com o framework JSF. Com esses componentes suas páginas XHTML (views) poderão conter, por exemplo: campos de texto que auto completam as informações de acordo com a entrada do usuário utilizando Ajax; gráficos para exibição de informações; calendários com exibição de pop-ups; dentre diversos outros. A Figura 1 apresenta alguns desses componentes.

**Figura 01** - Exemplos de componentes do Primefaces.



Fonte: <http://primefaces.org/>

Portanto, como características dessa suíte de componentes, temos:

- Um rico conjunto de componentes (editores, autocompletar, gráficos, calendários, etc);
- Ajax embutido nesses componentes, garantindo mais agilidade nas aplicações JSF;
- Componentes leves, configuração através de um Jar (veremos mais a frente);
- Um kit para criar aplicações móvel web, ou seja, aplicações web responsivas;
- Vários temas para deixar nossas aplicações visualmente mais bonitas;

- Uma extensiva documentação;
- Uma comunidade de desenvolvedores ativa.

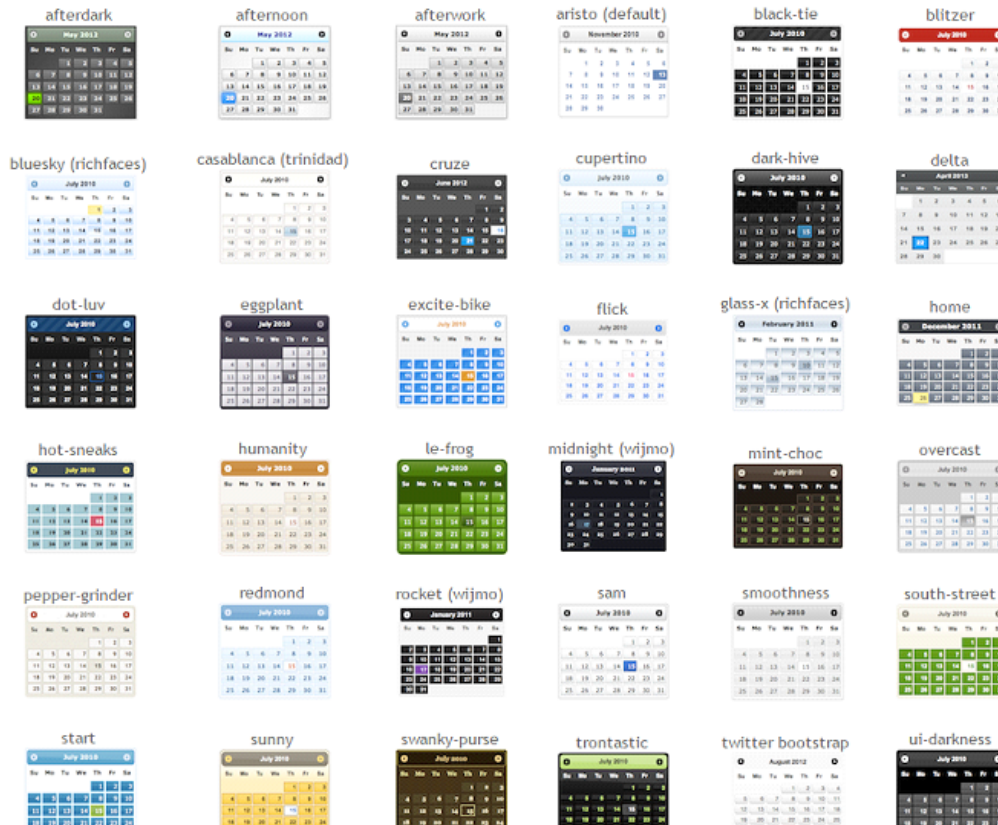
Essas características fazem com que o desenvolvimento de aplicações JSF com Primefaces se torne mais atrativo. Preparados para aprender a configurá-lo em suas aplicações JSF? Então, vamos lá!

## Configurando o Primefaces em uma Aplicação JSF

---

Para configurar o Primefaces nas suas aplicações JSF é necessário realizar o download da biblioteca `primefaces-6.2.jar`, no link <http://www.primefaces.org/downloads> na seção “Community Downloads”, e incluí-lo junto às demais bibliotecas existentes em seu projeto. Além disso, o Primefaces possui um conjunto de temas que modificam as cores e formas dos seus campos. A Figura 2 apresenta alguns desses temas que podem ser encontrados no link <http://primefaces.org/themes>. Dessa forma, vamos realizar também o download dos temas, **all-themes-1.0.10.jar**, no link <http://repository.primefaces.org/org/primefaces/themes/all-themes/1.0.10/>. Esse arquivo contém todos os temas de código abertos disponibilizados para serem utilizados com essa suíte de componentes, como pode ser observado na Figura 2.

**Figura 02** - Exemplos de temas que podem ser utilizados no Primefaces.



Já estamos com os arquivos jar do Primefaces: **primefaces-5.1.jar**, e dos temas: **all-themes-1.0.10.jar**. Como o Primefaces é uma suíte de componentes para serem utilizados com o JSF, precisaremos também da biblioteca do framework JSF. Você deve realizar o download do arquivo **javafx.faces-2.2.8.jar** no link <https://maven.java.net/content/repositories/releases/org/glassfish/javafx.faces/2.2.8/>.

Pronto! Agora já estamos com os três arquivos jar necessários para criar nossa primeira aplicação web com JSF e Primefaces. Vamos iniciar criando um projeto dinâmico web chamado HelloPrime. Ao criar o projeto, ficaremos com a estrutura definida na Figura 3.

**Figura 03** - Estrutura do projeto HelloPrime.



Agora, vamos colocar nossos três arquivos jar: **primefaces-5.1.jar**, **all-themes-1.0.10.jar** e **javax.faces-2.2.8.jar** dentro da pasta lib, cujo caminho é **WebContent\WEB-INF\lib\**. Dessa forma, já incluímos as bibliotecas necessárias para utilização do JSF e Primefaces nas nossas aplicações web. E o que precisamos fazer em seguida? Precisamos definir nosso web.xml, apresentado na Listagem 1, incluindo a Servlet do JSF e qual tema queremos utilizar no Primefaces.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xmlns="http://java.sun.com/xml/ns/javaee"
4 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5 http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
6
7     <display-name>HelloPrime</display-name>
8     <welcome-file-list>
9         <welcome-file>index.jsf</welcome-file>
10    </welcome-file-list>
11
12    <context-param>
13        <param-name>primefaces.THEME</param-name>
14        <param-value>start</param-value>
15    </context-param>
16
17    <servlet>
18        <servlet-name>Faces Servlet</servlet-name>
19        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
20        <load-on-startup>1</load-on-startup>
21    </servlet>
22    <servlet-mapping>
23        <servlet-name>Faces Servlet</servlet-name>
24        <url-pattern>*.jsf</url-pattern>
25    </servlet-mapping>
26
27 </web-app>
```

**Listagem 1** - Código do web.xml da aplicação HelloPrime

Observe, na Listagem 1, que a escolha do tema se dá através da configuração do parâmetro “primefaces.THEME”, cujo valor é o escolhido. Podemos escolher dentre os vários temas apresentados anteriormente na Figura 2. Para a nossa aplicação, utilizaremos o “start”. Que tal você testar outros na sua implementação?

Continuando, já temos as libs e o web.xml definidos. Vamos então criar a página index.xhtml, apresentada na Listagem 2, para exibir nosso primeiro componente na aplicação HelloPrime.

```

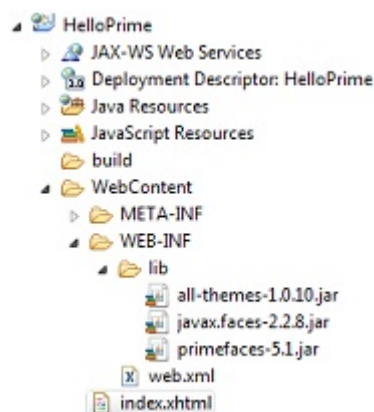
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml"
3 xmlns:h="http://java.sun.com/jsf/html"
4 xmlns:f="http://java.sun.com/jsf/core"
5 xmlns:p="http://primefaces.org/ui">
6   <h:head>
7     <title>Editor de textos</title>
8   </h:head>
9   <h:body>
10    <h1> Editando textos com o Prime</h1>
11    <hr/>
12    <p:editor />
13    <hr/>
14  </h:body>
15 </html>

```

**Listagem 2** - Página index.xhtml da aplicação HelloPrime

Observe na Listagem 2 que estamos definindo na Tag HTML o namespace “p” com a URL <http://primefaces.org/ui>. Isso é necessário para que possamos utilizar os componentes do Primefaces nas nossas páginas. Ainda na Listagem 2, observe que estamos utilizando o componente editor de textos do Primefaces (p:editor). Entenderam? Bem nosso projeto deve estar com a estrutura definida na Figura 4.

**Figura 04** - Estrutura completa do projeto HelloPrime.



Pronto! Agora é só adicionar o projeto HelloPrime aos projetos configurados para o Tomcat, iniciar o mesmo e acessar o endereço <http://localhost:8080/HelloPrime/>. Fazendo isso, você verá a tela apresentada na Figura 5.

**Figura 05** - Projeto HelloPrime com JSF e Primefaces em execução.



Vamos incrementar nosso exemplo com uma segunda página que utiliza o componente `p:autoComplete`: um campo que autocompleta as informações digitadas pelos usuários. Nesse exemplo, criaremos um `autoComplete` de países. Primeiramente, vamos criar uma `ManagedBean` denominada `PaisesMBean`. Esse `ManagedBean`, apresentado na Listagem 3, contém um atributo denominado “pais” e um método “completar”. Assim que iniciarmos a digitação do nome do país, o método `completar` será executado para apresentar o nome completo.



```

1 import java.util.ArrayList;
2 import java.util.List;
3
4 import javax.faces.bean.ManagedBean;
5 import javax.faces.bean.SessionScoped;
6
7 /**
8  * Managed Bean para autocompletar países.
9  * @author itamir.filho
10  *
11  */
12 @ManagedBean
13 @SessionScoped
14 public class PaisesMBean {
15
16     private String pais;
17
18     /**
19      * Método para
20      * @param pais
21      * @return
22      */
23     public List<String> completar(String pais) {
24         List<String> resultado = new ArrayList<String>();
25
26         List<String> paises = new ArrayList<String>();
27         paises.add("BRASIL");
28         paises.add("ESTADOS UNIDOS");
29         paises.add("ALEMANHA");
30         paises.add("ARGENTINA");
31
32         for (String p : paises) {
33             if(p.startsWith(pais.toUpperCase()))
34                 resultado.add(p);
35         }
36         return resultado;
37     }
38
39     public String getPais() {
40         return pais;
41     }
42
43     public void setPais(String pais) {
44         this.pais = pais;
45     }
46 }

```

**Listagem 3** - Código da classe PaisesMBean

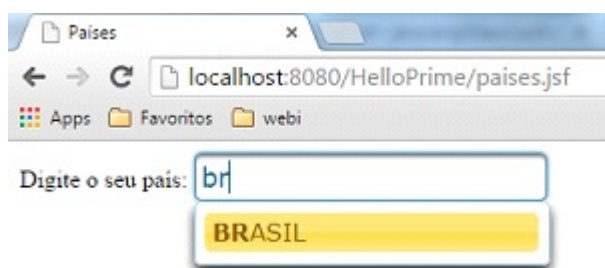
Criado o ManagedBean, vamos criar a página países.xhtml. Essa página, que é apresentada na Listagem 4, utiliza o componente autocomplete do Primefaces (p:autoComplete). Observe, ainda na Listagem 4, que referenciamos o método de completar (*completeMethod*) e a variável (value) definidos no ManagedBean PaísesMBean. Essa é a ligação entre a view e o controller.

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml"
3 xmlns:h="http://java.sun.com/jsf/html"
4 xmlns:f="http://java.sun.com/jsf/core"
5 xmlns:p="http://primefaces.org/ui">
6   <h:head>
7     <title>Países</title>
8   </h:head>
9   <h:body>
10    <h:form>
11      Digite o seu país:
12      <p:autoComplete value="#{paísesMBean.país}" completeMethod="#{paísesMBean.completa
13    </h:form>
14  </h:body>
15 </html>
```

**Listagem 4** - Código da página países.xhtml

Após a criação da classe PaísesMBean e da página países.xhtml, agora é só iniciar o Tomcat, e acessar o endereço <http://localhost:8080/HelloPrime/paises.jsf>. Ao fazer isso, você verá a tela apresentada na Figura 6.

**Figura 06**



Com isso, chegamos ao fim da nossa aula. E então, gostou do Primefaces? Viu como é fácil utilizá-lo em aplicações JSF? Existem diversos componentes do Primefaces que você poderá utilizar em suas aplicações. Para conhecer mais sobre eles, consulte as demonstrações em: <http://www.primefaces.org/showcase/index.xhtml>.

# Resumo

---

Nesta aula, definimos o que é o Primefaces e quais as características dessa suíte de componentes. Em seguida, para você aprender como configurá-lo em uma aplicação JSF, criamos uma aplicação denominada "HelloPrime". Durante o desenvolvimento dessa aplicação, foi apresentado o que é necessário em termos de bibliotecas (jars) e configurações para que você possa utilizar o Primefaces em aplicações JSF. Além disso, ainda nessa aplicação, apresentamos dois componentes do Primefaces bastante utilizados: um editor de texto e um campo autocomplete.

# Referências

---

JQUERY. Disponível em: <<http://jquery.com/>>. Acesso em: 1 ago. 2012.

W3c. Disponível em: <<http://www.w3.org/>>. Acesso em: 1 ago. 2012.

W3school. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 1 ago. 2012.

\_\_\_\_\_. **Ajax**. Disponível em: <<http://www.w3schools.com/Ajax/>>. Acesso em: 1 ago. 2012.