

## Um tema na aplicação inteira

### Transcrição

Na [documentação][5] do Primefaces, mostra como resolver esse problema do tema não ser persistido entre a navegação das páginas. Nós podemos dizer qual o tema que queremos através de um `context-param` no arquivo `web.xml`, mas também podemos dizer que o tema é dinâmico utilizando *expression language*, o usuário que irá escolher o tema de sua preferência. Então, lá no `web.xml`, adicione:

```
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>#{temaBean.tema}</param-value>
</context-param>
```

### Guardando a configuração no bean

Repare que a *expression language* chama o `tema`, da classe `TemaBean`, então vamos criar essa classe. Ela terá o atributo `tema`, com seus *getters* e *setters*, já pré-inicializado como `vader`, ou seja, a aplicação iniciará com o design desse tema. Lembrando que esse *bean* deverá ser anotado com `@ManagedBean` e ele terá que sobreviver durante várias requisições, ele tem que ser válido durante a sessão, logo ele também será anotado com `@SessionBean`:

```
@ManagedBean
@SessionScoped
public class TemaBean {

    private String tema = "vader";

    public String getTema() {
        return tema;
    }

    public void setTema(String tema) {
        this.tema = tema;
    }
}
```

Reiniciando o servidor e testando a aplicação, vemos que agora a aplicação já inicia com o tema `vader`. E navegando entre as páginas, vemos que o tema continua. Mas se mudarmos para `aristo` e trocarmos de página, o tema volta para o `vader`, então o problema original persistiu! Isso porque temos que mudar mais uma coisa. Se o usuário troca o tema através do *switcher*, precisamos reconfigurar o atributo do `TemaBean`. Para isso, vamos definir o atributo `value` do `themeSwitcher`, dizendo que ele é o tema do `TemaBean`:

```
<p:themeSwitcher id="basic" style="width:165px" value="#{temaBean.tema}">
  <f:selectItem itemLabel="Choose Theme" itemValue="" />
  <f:selectItem itemLabel="Aristo" itemValue="aristo" />
  <f:selectItem itemLabel="Vader" itemValue="vader" />
</p:themeSwitcher>
```

Só que por padrão, o `themeSwitcher` não utiliza ajax, então vamos adicionar a tag `f:ajax` para ele passar a fazê-lo:

```
<p:themeSwitcher id="basic" style="width:165px" value="#{temaBean.tema}">
  <f:selectItem itemLabel="Choose Theme" itemValue="" />
  <f:selectItem itemLabel="Aristo" itemValue="aristo" />
  <f:selectItem itemLabel="Vader" itemValue="vader" />
  <f:ajax />
</p:themeSwitcher>
```

Ótimo, agora tudo funciona como o esperado! Só que reparem que o *switcher* aparece também na página de login, para isso não acontecer vamos adicionar o atributo `rendered` no formulário, como já fizemos anteriormente:

```
<h:form rendered="#{usuarioLogado != null}">
  <p:themeSwitcher id="basic" style="width:165px" value="#{temaBean.tema}">
    <f:selectItem itemLabel="Choose Theme" itemValue="" />
    <f:selectItem itemLabel="Aristo" itemValue="aristo" />
    <f:selectItem itemLabel="Vader" itemValue="vader" />
    <f:ajax />
  </p:themeSwitcher>
</h:form>
```

Agora vamos colocar o *switcher* no lado direito da página, pois ele não é um elemento tão importante da página a ponto de dividir espaço com o menu. Então vamos colocá-lo do lado do logo, adicionando um `panelGrid` de duas colunas para agrupar esses dois componentes e colocando um estilo no formulário, através do atributo `style`, para jogá-lo para a direita:

```
<h:panelGrid columns="2">
  <h:graphicImage library="img" name="logo.png" />

  <h:form rendered="#{usuarioLogado != null}" style="position:absolute; right: 10px">
    <p:themeSwitcher id="basic" style="width:165px" value="#{temaBean.tema}">
      <f:selectItem itemLabel="Choose Theme" itemValue="" />
      <f:selectItem itemLabel="Aristo" itemValue="aristo" />
      <f:selectItem itemLabel="Vader" itemValue="vader" />
      <f:ajax />
    </p:themeSwitcher>
  </h:form>
</h:panelGrid>
```