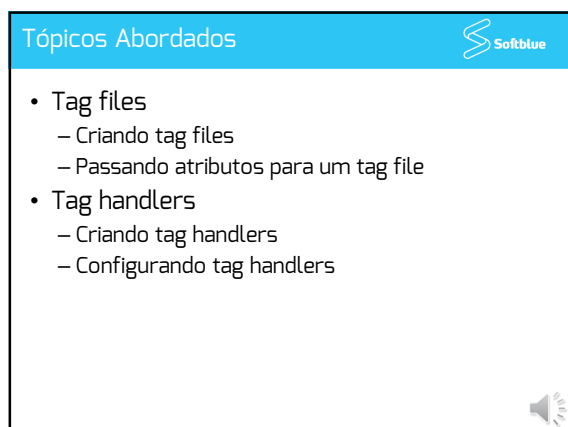
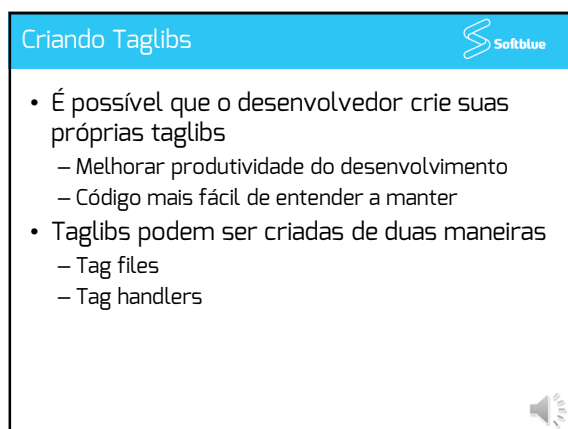




1



2



3

Tag Files

- Fragmentos de arquivos JSP
- Possuem a extensão **.tag**
- Estão localizados no diretório **WEB-INF/tags**

image.tag

```
<CENTER>
<IMG src="java.png"><BR>
</CENTER>
```

Este tag file adiciona uma imagem centralizada na página

4

Usando um Tag File

exemplo.jsp

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>

<html>
<body>
<t:image />
</body>
</html>
```

O *image* referencia o nome do tag file (*image.tag*)

5

Passando Atributos para Tag Files

image.tag

```
<%@ attribute name="imgName"
required="true"
rtexprvalue="true"
%>
<CENTER>
<IMG src="${imgName}">
</CENTER>
```

A diretiva **attribute** define os atributos da tag

imgName é substituído pelo atributo passado

exemplo.jsp

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>

<html>
<body>
<t:image imgName="java.png" />
</body>
</html>
```

Passagem de atributo

6

Informações no Corpo das Tags

image.tag

```
<%@ attribute name="imgName"
required="true"
rtexprvalue="true"
%>
<CENTER>
<IMG src="{imgName}"><BR>
<jsp:doBody />
<CENTER>
```

Imprime o corpo da tag sendo chamada

exemplo.jsp

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>
<html>
<body>
<t:image imgName="java.png">
  Logotipo do Java
</image>
</body>
</html>
```

Corpo da tag

7

Tag Handlers

- Tag files são interessantes quando não há necessidade de envolver código Java
- Nestes casos, é melhor usar um tag handler
- O **Tag Handler** é uma classe Java que representa uma taglib
- A classe estende **SimpleTagSupport** e sobreescreve o método **doTag()**

8

Criando um Tag Handler

```
public class DateTag extends SimpleTagSupport {
    private String pattern;

    public void doTag() throws JspException, IOException {
        Date d = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern);
        getJspContext().getOut().print(sdf.format(d));
    }

    public void setPattern(String pattern) {
        this.pattern = pattern;
    }
}
```

Envio do dado para a response

Setter invocado pelo container

9

Configurando o Tag Handler

- A configuração é feita em um arquivo **.tld** (tag library descriptor), dentro de **WEB-INF**

```
mytags.tld<taglib>
<tlb-version>1.0</tlb-version>
<short-name>MyCustomTags</short-name>
<uri>myTags</uri>
<tag>
  <description>Formata uma data</description>
  <name>date</name>
  <tag-class>tag.DateTag</tag-class>
  <body-content>empty</body-content>
  <attribute>
    <name>pattern</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
</tag>
</taglib>
```

10

Usando o Tag Handler

```
exemplo.jsp<%@ taglib prefix="t" uri="myTags" %>
<html>
<body>
  <t:date pattern="dd/MM/yyyy" />
</body>
</html>
```

O container chama o método `setPattern()` no handler

Referencia a URI do arquivo `mytags.tld`

11

Tag Handler e o Corpo da Tag

```
public class MyTag extends SimpleTagSupport {
    public void doTag() throws JspException, IOException {
        <code>getJspContext().invoke(null);</code>
    }
}
```

Imprime o conteúdo do corpo da tag na response

É possível passar um `Writer` como parâmetro

12

- Atributo das tags: **rtexprvalue**

Valor	Significado
true	Strings literais e outras expressões (como EL e scriptlets) são aceitas como valor para o atributo
false	Apenas strings literais são aceitas como valor para o atributo

- Atributo do corpo das tags: **body-content**

Valor	Significado
empty	A tag não pode ter corpo
scriptless	O corpo da tag não pode conter scriptlets (mas aceita expressões EL)
tagdependent	O corpo da tag é interpretado como texto