

# 27

## JavaServer Pages (JSP)



# OBJETIVOS

- Neste capítulo, você aprenderá:
- O que são JSPs.
- As diferenças entre servlets e JSPs.
- Como criar e implantar JavaServer Pages.
- Como utilizar objetos implícitos e scriptlets da JSP para criar páginas da Web dinâmicas.
- Como especificar informações globais da JSP com diretivas.
- Como utilizar ações para manipular JavaBeans em uma JSP, incluir recursos dinamicamente e encaminhar solicitações para outras JSPs.



- 27.1**    **Introdução**
- 27.2**    **Visão geral das JavaServer Pages**
- 27.3**    **Primeiro exemplo de JSP**
- 27.4**    **Objetos implícitos**
- 27.5**    **Script**
  - 27.5.1**    **Componentes de script**
  - 27.5.2**    **Exemplo de script**
- 27.6**    **Ações-padrão**
  - 27.6.1**    **Ação <jsp:include>**
  - 27.6.2**    **Ação <jsp:forward>**
  - 27.6.3**    **Ação <jsp:useBean>**
- 27.7**    **Diretivas**
  - 27.7.1**    **Diretiva page**
  - 27.7.2**    **Diretiva include**
- 27.8**    **Estudo de caso: Livro de visitas**
- 27.9**    **Conclusão**
- 27.10**   **Internet e recursos Web**



# 27.1 Introdução

- **JavaServer Pages:**
  - Extensão da tecnologia de servlet, que separa a apresentação da lógica do negócio.
  - Simplificam a entrega de conteúdo Web dinâmico.
  - Reutilizam os componentes Java existentes:
    - **JavaBean.**
    - **Bibliotecas de tags personalizadas, que encapsulam funcionalidades complexas.**
- **Classes e interfaces específicas de JSP:**
  - Pacote `javax.servlet.jsp`
  - Pacote `javax.servlet.jsp.tagext`



# 27.2 Visão geral das JavaServer Pages

- **Componentes-chave:**
  - **Diretivas.**
  - **Ações.**
  - **Elementos de script.**
  - **Bibliotecas de tags**



## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **Diretivas:**

- Mensagens para o contêiner de JSP.
  - Isto é, o programa que compila/executa JSPs.
- Permitem que os programadores especifiquem:
  - configurações de página, incluindo conteúdo a partir de outros recursos; e
  - bibliotecas de tags personalizadas para utilização em uma JSP.



## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **Ações:**

- Tags JSP predefinidas que encapsulam funcionalidades.
- Frequentemente realizadas com base nas informações da solicitação do cliente.
- Elas podem ser utilizadas para criar objetos Java para uso em scriptlets de JSP.



## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **Elementos de script:**

- Permitem que programadores insiram código Java em JSPs.
- Realizam o processamento de solicitação:
  - Interagem com elementos **page** e outros componentes para implementar páginas dinâmicas.
- **Scriptlets:**
  - Um tipo de elemento de script.
  - Contêm fragmentos de código.
    - Descrevem a ação a ser realizada em resposta à solicitação de usuário.





## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **Bibliotecas de tags:**
  - Fazem parte do mecanismo de extensão de tag da JSP.
  - Permitem que programadores definam novas tags:
    - Tags encapsulam a funcionalidade complexa.
  - Tags podem manipular o conteúdo da JSP.



## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **JSPs:**

- Semelhantes aos documentos XHTML ou XML padrão.
  - Normalmente incluem marcação de XHTML ou de XML:
    - Tal marcação é conhecida como dado de template fixo ou texto de template fixo .
- Utilizadas quando o conteúdo são principalmente dados de template fixo.
  - Pequenas quantidades do conteúdo geradas dinamicamente.

- **Servlets:**

- Utilizadas quando pequenas quantidades do conteúdo são dados de template fixo.
  - A maior parte do conteúdo gerada dinamicamente.



# Observação de engenharia de software 27.1

---

**O texto literal em uma JSP torna-se literal de string no servlet que representa a JSP traduzida.**

## 27.2 Visão geral das JavaServer Pages (*Continuação*)

- **Quando o servidor recebe a primeira solicitação JSP:**
  - O contêiner JSP traduz uma JSP em um servlet.
    - Trata as solicitações atuais e futuras.
- **Código que representa a JSP:**
  - Colocado no método `_jspService` do servlet.
- **Erros de JSP:**
  - Erros em tempo de tradução:
    - Ocorrem quando JSPs são traduzidas em servlets.
  - Erros em tempo de solicitação:
    - Ocorrem durante o processamento de solicitação.
- **Métodos `jspInit` e `jspDestroy`:**
  - O contêiner os invoca ao inicializar e terminar uma JSP.
  - Definidos nas declarações de JSP.



## Dica de desempenho 27.1

---

**Alguns contêineres de JSP traduzem JSPs em servlets em tempo de instalação. Isso elimina o overhead de tradução para o primeiro cliente que solicita cada JSP.**



## 27.3 Primeiro exemplo de JSP

- **Exemplo simples de JSP (Figura 27.1).**
  - **Demonstra:**
    - **Dados de template fixa (marcação de XHTML).**
    - **Criando um objeto Java (`java.util.Date`).**
    - **Conversão automática de expressão JSP em uma `String`.**
    - **Elemento `meta` para atualizar a página da Web no intervalo especificado.**
  - **Primeira invocação de `clock.jsp`:**
    - **Observe o retardo enquanto:**
      - **O contêiner JSP traduz a JSP em um servlet.**
      - **O contêiner JSP compila o servlet.**
      - **O contêiner JSP executa o servlet.**
    - **Invocações subseqüentes não devem experimentar o mesmo retardo.**



# Resumo

clock.jsp

(1 de 2)

```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.1: clock.jsp -->
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>
9     <meta http-equiv = "refresh" content = "60" />
10    <title>A Simple JSP Example</title>
11    <style type = "text/css">
12      .big { font-family: helvetica, arial, sans-serif;
13            font-weight: bold;
14            font-size: 2em; }
15    </style>
16  </head>
17  <body>
18    <p class = "big">Simple JSP Example</p>
19    <table style = "border: 6px outset;">
20      <tr>
21        <td style = "background-color: black;">
22          <p class = "big" style = "color: cyan;">
23            <!-- expressão de JSP para inserir data/hora -->
24            <%= new java.util.Date() %>
25          </p>
26        </td>
27      </tr>
28    </table>
29  </body>
30 </html>

```

Elemento **meta** que atualiza a página da Web a cada 60 segundos

Cria o objeto **Date** que é convertido em uma **String** implicitamente e exibido no elemento paragraph (p)

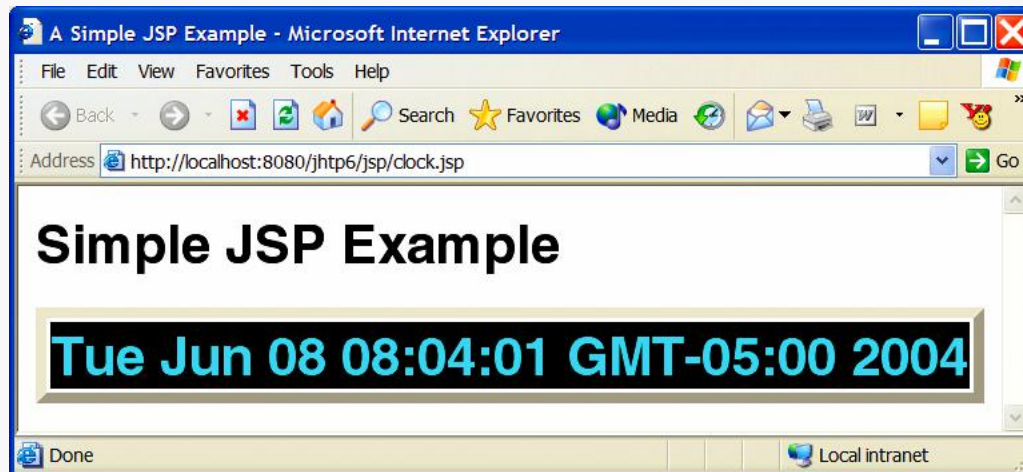
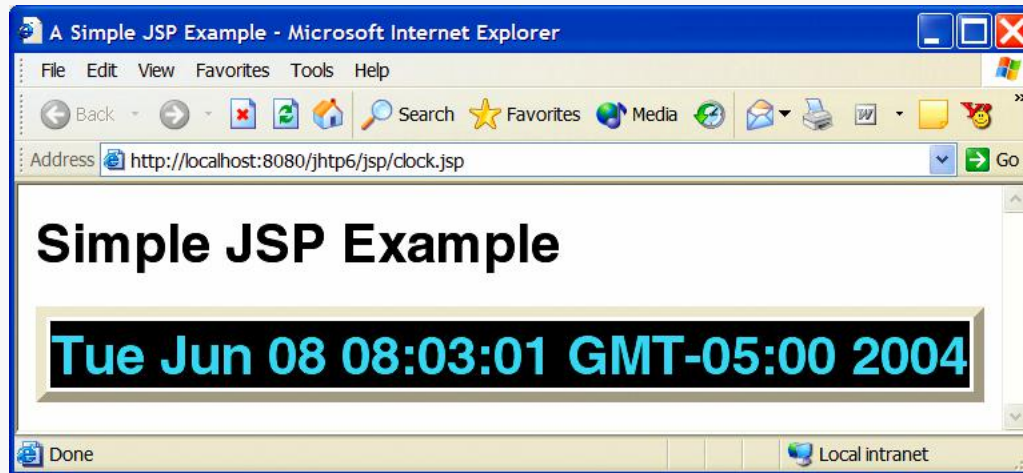


# Resumo

`clock.jsp`

(2 de 2)

Saída do programa





# Observação de engenharia de software 27.2

---

**As JavaServer Pages são mais fáceis de implementar do que os servlets quando a resposta a uma solicitação de cliente consistir principalmente de marcação que permanece constante entre solicitações.**

# Observação de engenharia de software 27.3

---

**O contêiner de JSPs converte o resultado de cada expressão de JSP em uma string que é enviada para a saída como parte da resposta ao cliente.**

## 27.4 Objetos implícitos

- **Objetos implícitos:**

- Fornece acesso a muitas capacidades de servlet dentro de uma JSP.
- Quatro escopos:
  - *Escopo application:*
    - Objetos possuídos pela aplicação contêiner.
    - Qualquer servlet ou JSP pode manipular esses objetos.
  - *Escopo page:*
    - Objetos que só existem na página em que eles são definidos.
    - Cada página tem sua própria instância desses objetos.



## 27.4 Objetos implícitos (*Continuação*)

- *Escopo request:*
  - Objetos existem pela duração da solicitação do cliente.
  - Os objetos saem de escopo quando a resposta é enviada ao cliente.
- *Escopo session:*
  - Os objetos existem pela duração da sessão de navegação do cliente.
  - Os objetos saem de escopo quando o cliente termina a sessão ou quando o tempo limite de sessão ocorre.



Objeto implícito	Descrição
<i>Escopo application</i> application	Um objeto <code>javax.servlet.ServletContext</code> que representa o contêiner em que a JSP executa.
<i>Escopo page</i> config	Um objeto <code>javax.servlet.ServletConfig</code> que representa as opções de configuração do JSP. Como com os servlets, as opções de configuração podem ser especificadas em um descritor de aplicativo Web.
exception	Um objeto <code>java.lang.Throwable</code> que representa uma exceção que é passada para uma página de erro JSP. Esse objeto está disponível somente em uma página de erro JSP.
out	Um objeto <code>javax.servlet.jsp.JspWriter</code> que escreve texto como parte da resposta para uma solicitação. Esse objeto é utilizado implicitamente com as expressões e ações JSP que inserem conteúdo de string em uma resposta.
page	Um <code>Object</code> que representa a referência <code>this</code> para a instância JSP atual.

**Figura 27.2 |** Objetos implícitos JSP. (Parte 1 de 2.)

Objeto implícito	Descrição
<b>pageContext</b>	Um objeto <code>javax.servlet.jsp.PageContext</code> que fornece programadores em JSP com acesso aos objetos implícitos discutidos nessa tabela.
<b>response</b>	Um objeto que representa a resposta ao cliente e, normalmente, é uma instância de uma classe que implementa <code>HttpServletResponse</code> (pacote <code>javax.servlet.http</code> ). Se um protocolo diferente de HTTP for utilizado, esse objeto é uma instância de uma classe que implementa <code>javax.servlet.ServletResponse</code> .
<i>Escopo request</i>	
<b>request</b>	Um objeto que representa a solicitação de cliente e, normalmente, é uma instância de uma classe que implementa <code>HttpServletRequest</code> (pacote <code>javax.servlet.http</code> ). Se um protocolo diferente de HTTP for utilizado, esse objeto é uma instância de uma subclasse de <code>javax.servlet.ServletRequest</code> .
<i>Escopo session</i>	
<b>session</b>	Um objeto <code>javax.servlet.http.HttpSession</code> que representa as informações de sessão de cliente se essa sessão tiver sido criada. Esse objeto é disponível apenas em páginas que participam de uma sessão.

**Fig. 27.2 | Objetos implícitos JSP. (Parte 2 de 2.)**

## 27.5 Script

- **Script:**
  - Conteúdo dinamicamente gerado.
  - Insere o código Java e a lógica na JSP utilizando scripts.



## 27.5.1 Componentes de script

- **Componentes de script JSP:**

- Scriptlets são blocos de código delimitados por `<% e %>`.
- Comentários de JSP:
  - Comentários na JSP (delimitados por `<%-- e --%>`).
  - Comentários XHTML (delimitados por `<!-- e -->`).
  - Comentários do Java (delimitados por `// e /* e */`).
- Expressões (delimitadas por `<%= e %>`).
- Declarações (delimitadas por `<%! e %>`).
- Sequências de escape.





## Erro comum de programação 27.1

---

**Colocar um comentário de JSP ou um comentário de XHTML dentro de um scriptlet é um erro de sintaxe em tempo de tradução que impede que a JSP seja traduzida adequadamente.**

# Observação de engenharia de software 27.4

---

**As JSPs não devem armazenar informações de estado do cliente em variáveis de instância. Em vez disso, devem utilizar objeto `session` implícito de JSP. Para informações adicionais sobre como utilizar o objeto `session`, visite o tutorial sobre o J2EE da Sun em [java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html).**

Literal	Seqüência de escape	Descrição
<code>&lt;%</code>	<code>&lt;\%</code>	A seqüência de caractere <code>&lt;%</code> normalmente indica o começo de um scriptlet. A seqüência de escape <code>&lt;\%</code> coloca os caracteres literais <code>&lt;%</code> na resposta ao cliente.
<code>%&gt;</code>	<code>%\&gt;</code>	A seqüência de caractere <code>%&gt;</code> normalmente indica o fim de um scriptlet. A seqüência de escape <code>%\&gt;</code> coloca os caracteres literais <code>%&gt;</code> na resposta ao cliente.
<code>'</code> <code>"</code> <code>\</code>	<code>\'</code> <code>\"</code> <code>\\</code>	Como com literais de string em um programa Java, as seqüências de escape para os caracteres <code>'</code> , <code>"</code> e <code>\</code> permitem que esses caracteres apareçam em valores de atributo. Lembre-se de que o texto literal em uma JSP torna-se literais de string no servlet que representa a JSP traduzida.

**Figura 27.3 | Seqüências de escape de JSP.**

## 27.5.2 Exemplo de script

- **Demonstra as capacidades básicas dos scripts.**
  - Respondendo a solicitações **get**.



# Observação de engenharia de software 27.5

---

**Os scriptlets, as expressões e os dados de template fixo podem ser misturados em uma JSP para criar diferentes respostas baseadas nas informações em uma solicitação.**

# Resumo

## welcome.jsp

(1 de 3)

Linhas 17-23

Linha 19

Linha 26

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 27.4: welcome.jsp -->
6  <!-- JSP que processa uma solicitação "get" contendo dados. -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9
10   <!-- seção de cabeçalho do documento -->
11   <head>
12     <title>Processing "get" requests with data</title>
13   </head>
14
15   <!-- seção de corpo do documento -->
16   <body>
17     <% // inicia o scriptlet
18
19     String name = request.getParameter("name");
20
21     if ( name != null )
22     {
23       <%= name %>
24
25     <h1>
26       Hello <%= name %>, <br />
27       Welcome to JavaServer Pages!
28     </h1>
29   </body>
30 </html>

```

Scriptlet  
utilizado para  
inserir código  
Java

Utiliza o objeto  
implícito **request**  
para obter o parâmetro

Expressão JSP



# Resumo

welcome.jsp

(2 de 3)

Scriptlets utilizados para  
inserir código Java

Linhas 30-33 e Linhas

45-49

```
30 <% // continua scriptlet
31
32 } // fim do if
33 else {
34
35 %> <!-- fim de scriptlet para inserir de dados de template fixa --%>
36
37 <form action = "welcome.jsp" method = "get">
38 <p>Type your first name and press Submit</p>
39
40 <p><input type = "text" name = "firstName" />
41 <input type = "submit" value = "Submit" />
42 </p>
43 </form>
44
45 <% // continua scriptlet
46
47 } // fim do else
48
49 %> <!-- fim do scriptlet --%>
50 </body>
51
52 </html> <!--fim do documento XHTML -->
```

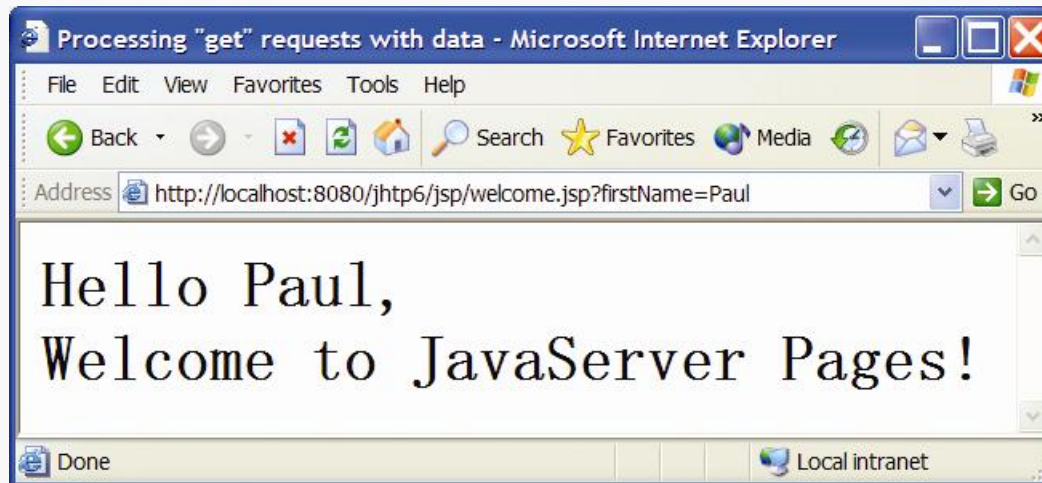
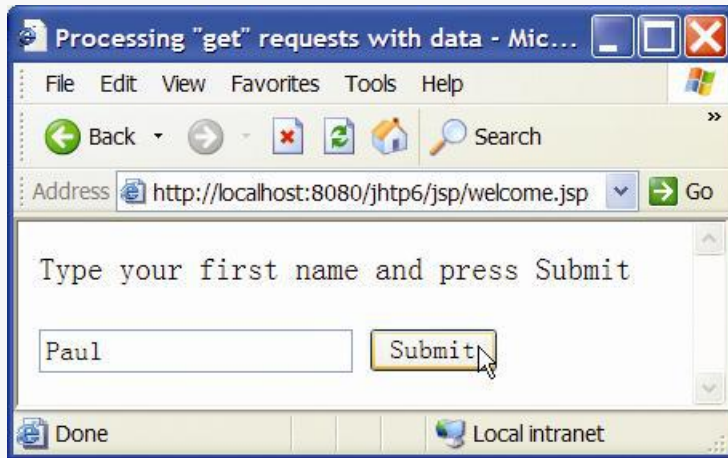


# Resumo

welcome.jsp

(3 de 3)

Saída do programa





## Dica de prevenção de erro 27.1

---

**Às vezes, é difícil depurar erros em uma JSP, porque os números de linha informados por um contêiner de JSPs normalmente referenciam o servlet que representa a JSP traduzida, não os números de linha da JSP original. Os ambientes de desenvolvimento de programas permitem que JSPs sejam compiladas no ambiente, assim você pode ver as mensagens de erro de sintaxe. Essas mensagens incluem a instrução no servlet que representa a JSP traduzida, o que pode ser útil na determinação do erro.**

---



## Dica de prevenção de erro 27.2

---

**Muitos contêineres de JSP armazenam o código-fonte para os servlets que representam as JSPs traduzidas. Por exemplo, o diretório de instalação do Tomcat contém um subdiretório chamado `work` em que você pode localizar o código-fonte para os servlets traduzidos pelo Tomcat. Lembre-se, a partir do que foi discutido no Capítulo 26, de que os arquivos de log localizados no subdiretório `logs` do diretório de instalação do Tomcat também são úteis para determinar os erros.**



## Dica de prevenção de erro 27.3

---

**Sempre coloque a chave de fechamento para a instrução `if` e a instrução `else` no mesmo scriptlet.**



## 27.6 Ações-padrão

- **Ações-padrão de JSP:**
  - **Fornecem acesso a tarefas comuns realizadas em uma JSP:**
    - **Incluindo conteúdo a partir de outros recursos.**
    - **Redirecionando solicitações para outros recursos.**
    - **Interagindo com JavaBeans.**
  - **Os contêineres de JSP processam ações em tempo de solicitação.**
  - **Delimitadas por `<jsp:ação>` e `</jsp:ação>`.**



Ação	Descrição
<code>&lt;jsp:include&gt;</code>	Inclui dinamicamente outro recurso em uma JSP. Quando a JSP executa, o recurso referenciado é incluído e processado.
<code>&lt;jsp:forward&gt;</code>	Encaminha processamento de solicitação para outra JSP, servlet ou página estática. Essa ação termina a execução da JSP atual.
<code>&lt;jsp:plugin&gt;</code>	Permite que um componente plug-in seja adicionado a uma página no formulário de um elemento <code>object</code> específico do navegador ou elemento de HTML <code>embed</code> . No caso de um applet Java, essa ação permite ao navegador fazer download e instalar o Java Plug-in, se ele ainda não tiver sido instalado no computador de cliente.
<code>&lt;jsp:param&gt;</code>	Utilizado com as ações <code>include</code> , <code>forward</code> e <code>plugin</code> para especificar pares nome-valor adicionais de informações a ser utilizadas por essas ações.

**Figura 27.5 | Ações-padrão de JSP. (Parte 1 de 2.)**



Ação	Descrição
<i>Manipulação do JavaBean</i> <b>&lt;jsp:useBean&gt;</b>	Especifica que o JSP utiliza uma instância de JavaBean (isto é, um objeto da classe que declara o JavaBean). Essa ação especifica o escopo do objeto e atribui um ID (isto é, um nome de variável) que os componentes de script podem utilizar para manipular o bean.
<b>&lt;jsp:setProperty&gt;</b>	Configura uma propriedade na instância JavaBean especificada. Um recurso especial dessa ação é a correspondência automática de parâmetros de solicitação para propriedades de bean do mesmo nome.
<b>&lt;jsp:getProperty&gt;</b>	Obtém uma propriedade na instância de JavaBean especificada e converte o resultado em uma string para saída na resposta.

**Figura 27.5 | Ações-padrão de JSP. (Parte 2 de 2.)**



## 27.6.1 Ação <jsp:include>

- Ação <jsp:include>:
  - Permite que o conteúdo dinâmico seja incluído em uma JSP.
  - Mais flexível que a diretiva include:
    - Requer mais overhead quando o conteúdo da página muda frequentemente.



## Observação de engenharia de software 27.6

---

**De acordo com a especificação de JavaServer Pages 2.0, um contêiner de JSP tem a permissão de determinar se um recurso incluído com a diretiva `include` foi alterado. Se tiver sido alterado, o contêiner pode recompilar a JSP que incluiu o recurso. Entretanto, a especificação não fornece um mecanismo para indicar uma alteração em um recurso incluído para o contêiner.**



## Dica de desempenho 27.2

---

**A ação `<jsp:include>` é mais flexível do que a diretiva `include`, mas requer mais overhead quando o conteúdo de página muda com frequência. Utilize a ação `<jsp:include>` somente quando o conteúdo dinâmico for necessário.**

## Erro comum de programação 27.2

---

**Especificar em uma ação `<jsp:include>` uma página que não faz parte do mesmo aplicativo Web é um erro de tempo de solicitação — a ação `<jsp:include>` não incluirá nenhum conteúdo.**



Atributo	Descrição
<b>page</b>	<b>Especifica o caminho URI relativo do recurso a incluir. O recurso deve fazer parte do mesmo aplicativo Web.</b>
<b>flush</b>	<b>Especifica se o objeto implícito <code>out</code> deve ser esvaziado antes de <code>include</code> ser executado. Se <code>true</code>, <code>JspWriter out</code> é esvaziado antes da inclusão, daí você não pode mais encaminhar para outra página posteriormente. O valor padrão é <code>false</code>.</b>

**Figura 27.6 | Atributos da ação `<jsp:include>`.**



# Resumo

## include.jsp

(1 de 3)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.7: include.jsp -->
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8
9   <head>
10     <title>Using jsp:include</title>
11
12     <style type = "text/css">
13       body
14       {
15         font-family: tahoma, helvetica, arial, sans-serif;
16       }
17
18       table, tr, td
19       {
20         font-size: .9em;
21         border: 3px groove;
22         padding: 5px;
23         background-color: #dddddd;
24       }
25     </style>
26   </head>
27
```



# Resumo

include.jsp

(2 de 3)

```

28 <body>
29   <table>
30     <tr>
31       <td style = "width: 160px; text-align: center">
32         <img src = "images/logotiny.png"
33           width = "140" height = "93"
34           alt = "Deitel & Associates, Inc. Logo" />
35       </td>
36       <td>
37         <!-- inclui banner.html neste JSP -->
38         <jsp:include page = "banner.html"
39           flush = "true" />
40       </td>
41     </tr>
42     <tr>
43       <td style = "width: 160px">
44         <!-- inclui toc.html neste JSP -->
45         <jsp:include page = "toc.html" flush = "true" />
46       </td>
47       <td style = "vertical-align: top">
48         <!-- inclui clock2.jsp neste JSP -->
49         <jsp:include page = "clock2.jsp"
50           flush = "true" />
51       </td>
52     </tr>
53   </table>
54 </body>
55 </html>

```

Utiliza a ação JSP  
**include** para incluir  
banner.html

Utiliza a ação JSP  
**include** para incluir  
toc.html

Utiliza a ação JSP  
**include** para incluir  
clock2.jsp

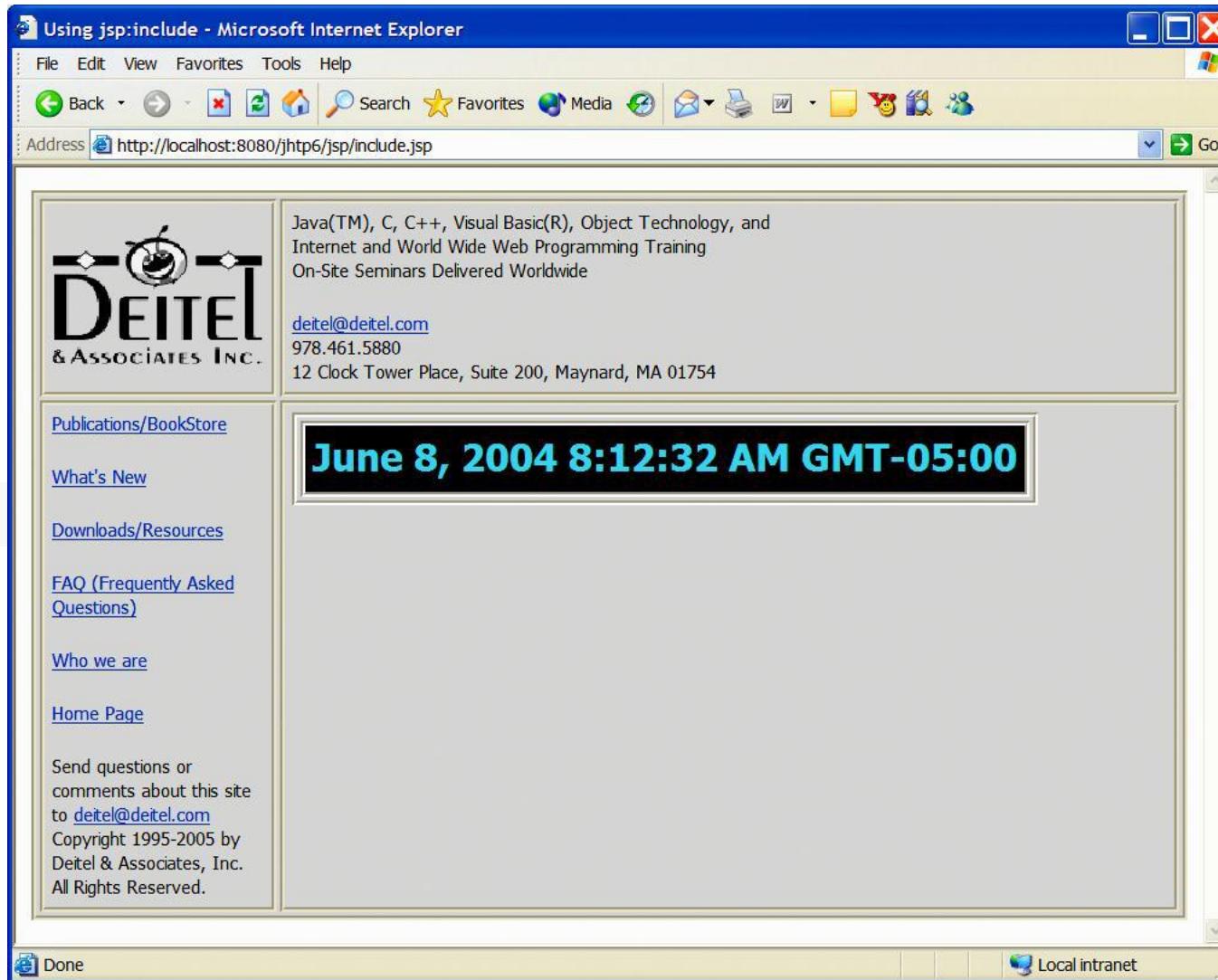


# Resumo

include.jsp

(3 de 3)

Saída do programa



# Resumo

banner.html

```
1 <!-- Fig. 27.8: banner.html -->
2 <!-- banner para incluir em outro documento -->
3 <div style = "width: 580px">
4     <p>
5         Java(TM), C, C++, Visual Basic(R),
6         Object Technology, and <br /> Internet and
7         World wide Web Programming Training <br />
8         On-Site Seminars Delivered worldwide
9     </p>
10    <p>
11        <a href = "mailto:deitel@deitel.com">deitel@deitel.com</a>
12        <br />978.461.5880<br />12 Clock Tower Place, Suite 200,
13        Maynard, MA 01754
14    </p>
15 </div>
```



# Resumo

toc.html

```
1 <!-- Fig. 27.9: toc.html -->
2 <!-- conteúdo a incluir em outro documento -->
3
4 <p><a href = "http://www.deitel.com/books/index.html">
5     Publications/BookStore
6 </a></p>
7
8 <p><a href = "http://www.deitel.com/whatsnew.html">
9     What's New
10 </a></p>
11
12 <p><a href = "http://www.deitel.com/books/downloads.html">
13     Downloads/Resources
14 </a></p>
15
16 <p><a href = "http://www.deitel.com/faq/index.html">
17     FAQ (Frequently Asked Questions)
18 </a></p>
19
20 <p><a href = "http://www.deitel.com/intro.html">
21     Who we are
22 </a></p>
23
24 <p><a href = "http://www.deitel.com/index.html">
25     Home Page
26 </a></p>
27
28 <p>Send questions or comments about this site to
29     <a href = "mailto:deitel@deitel.com">
30         deitel@deitel.com
31     </a><br />
32     Copyright 1995-2005 by Deitel & Associates, Inc.
33     All Rights Reserved.
34 </p>
```





# Resumo

clock2.jsp ciLinha  
14iLinhas 17-  
20iLinha 25

```

1  <!-- Fig. 27.10: clock2.jsp -->
2  <!-- data e hora a incluir em outro documento -->
3
4  <table>
5      <tr>
6          <td style = "background-color: black;">
7              <p class = "big" style = "color: cyan; font-size: 3em;
8                  font-weight: bold;">
9
10                 <!-- script para determinar local de cliente e --%>
11                 <!-- formatar data de maneira correspondente --%>
12                 <%
13                     // obtém o local do cliente
14                     java.util.Locale locale = request.getLocale();
15
16                     // obtém DateFormat para o Locale do cliente
17                     java.text.DateFormat dateFormat =
18                         java.text.DateFormat.getInstance(
19                             java.text.DateFormat.LONG,
20                             java.text.DateFormat.LONG, locale );
21
22                 <%-- fim do script --%>
23
24                 <!-- saída da data --%>
25                 <%= dateFormat.format( new java.util.Date() ) %>
26             </p>
27         </td>
28     </tr>
29 </table>

```

Utiliza o método `getLocale`  
e do objeto

Invoca o método estático  
`getInstance` de  
`DateFormat` para obter um objeto  
`DateFormat` para o `Locale`  
especificado

Formata `Date` com o  
`DateFormat`  
especificado



## 27.6.2 Ação <jsp:forward>

- **Ação <jsp:forward>:**
  - Permite que uma JSP encaminhe uma solicitação a recursos diferentes.
    - Encaminha solicitações a recursos no mesmo contexto.
- **Ação <jsp:param>:**
  - Especifica pares nome-valor de informações.
    - Os pares nome-valor são passados para outras ações.



# Observação de engenharia de software 27.7

---

**Ao utilizar a ação `<jsp:forward>`, o recurso para o qual a solicitação será encaminhada deve estar no mesmo contexto (aplicativo Web) da JSP que recebeu originalmente a solicitação.**

# Resumo

## forward1.jsp

(1 de 2)

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 27.11: forward1.jsp -->t
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8  <head>
9      <title>Forward request to another JSP</title>
10 </head>
11 <body>
12     <% // inicia o scriptlet
13
14         String name = request.getParameter( "firstName" );
15
16         if ( name != null )
17         {
18             <%-- fim do scriptlet para inserir de dados de template fixa --%>
19
20             <jsp:forward page = "forward2.jsp">
21                 <jsp:param name = "date"
22                     value = "<%= new java.util.Date() %>" />
23             </jsp:forward>
24
25             <% // continua scriptlet
26
27             } // fim do if
28             else
29             {
30                 <%-- fim do scriptlet para inserir de dados de template fixa --%>

```

Utiliza o objeto implícito **request** para obter o parâmetro

Encaminha a solicitação a **forward2.jsp**



# Resumo

## forward1.jsp

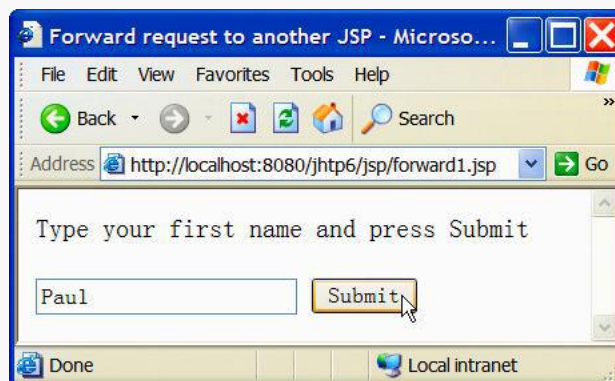
(2 de 2)

Saída do programa

```

31 <form action = "forward1.jsp" method = "get">
32     <p>Type your first name and press Submit</p>
33
34     <p><input type = "text" name = "firstName" />
35         <input type = "submit" value = "Submit" />
36     </p>
37 </form>
38
39
40 <% // continua o scriptlet
41
42     } // fim do else
43
44 %> <%-- fim do scriptlet --%>
45 </body>
46 </html> <!--fim do documento XHTML -->

```



# Resumo

## forward2.jsp

(1 de 2)

Linha 21

Recebe a solicitação de  
forward1.jsp e, então,  
obtem o parâmetro  
firstName na solicitação

Obtem o parâmetro  
date da solicitação

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 27.12: forward2.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8  <head>
9      <title>Processing a forwarded request</title>
10     <style type = "text/css">
11         .big
12         {
13             font-family: tahoma, helvetica, arial, sans-serif;
14             font-weight: bold;
15             font-size: 2em;
16         }
17     </style>
18 </head>
19 <body>
20     <p class = "big">
21         Hello <%= request.getParameter( "firstName" ) %>, <br />
22         Your request was received <br /> and forwarded at
23     </p>
24     <table style = "border: 6px outset;">
25         <tr>
26             <td style = "background-color: black;">
27                 <p class = "big" style = "color: cyan;">
28                     <%= request.getParameter( "date" ) %>
29                 </p>
30             </td>
31         </tr>
32     </table>
33 </body>
34 </html>

```





### Saída do programa



## 27.6.3 Ação <jsp:useBean>

- **Ação <jsp:useBean>:**
  - **Permite que a JSP manipule objetos Java.**
    - **Cria um objeto Java ou localiza um objeto existente para uso na JSP.**





## Erro comum de programação 27.3

---

**Um ou ambos os atributos `<jsp:useBean>`, `class` e `type` devem ser especificados — caso contrário, ocorre um erro em tempo de conversão.**

Atributo	Descrição
<code>id</code>	O nome utilizado para manipular o objeto Java com as ações <code>&lt;jsp:setProperty&gt;</code> e <code>&lt;jsp:getProperty&gt;</code> . Uma variável desse nome também é declarada para utilização em elementos de script JSP. O nome especificado aqui faz distinção entre letras maiúsculas e minúsculas.
<code>scope</code>	O escopo em que o objeto Java é acessível — <code>page</code> , <code>request</code> , <code>session</code> ou <code>application</code> . O escopo-padrão é <code>page</code> .
<code>class</code>	O nome de classe completamente qualificado do objeto Java.
<code>beanName</code>	O nome de um JavaBean que pode ser utilizado com o método <code>instantiate</code> da classe <code>java.beans.Beans</code> para carregar um JavaBean na memória.
<code>type</code>	O tipo do JavaBean. Esse pode ser do mesmo tipo do atributo <code>class</code> , uma superclasse desse tipo ou uma interface implementada por esse tipo. O valor-padrão é o mesmo para o atributo <code>class</code> . Ocorre uma <code>ClassCastException</code> se o objeto Java não for do tipo especificado com o atributo <code>type</code> .

**Figura 27.13** | Atributos da ação `<jsp:useBean>`.

# Resumo

## Rotator.Java

(1 de 2)

Linhas 26-29

```
1 // Fig. 27.14: Rotator.java
2 // Um JavaBean que faz rotação de anúncios.
3 package com.deitel.jhttp6.jsp.beans;
4
5 public class Rotator
6 {
7     private String images[] = { "images/advjHTTP1.jpg",
8     "images/cppHTTP4.jpg", "images/iw3HTTP2.jpg",
9     "images/jwsFEP1.jpg", "images/vbnetHTTP2.jpg" };
10
11     private String links[] = {
12         "http://www.amazon.com/exec/obidos/ASIN/0130895601/" +
13         "deitelassociatin",
14         "http://www.amazon.com/exec/obidos/ASIN/0130384747/" +
15         "deitelassociatin",
16         "http://www.amazon.com/exec/obidos/ASIN/0130308978/" +
17         "deitelassociatin",
18         "http://www.amazon.com/exec/obidos/ASIN/0130461342/" +
19         "deitelassociatin",
20         "http://www.amazon.com/exec/obidos/ASIN/0130293636/" +
21         "deitelassociatin" };
22
23     private int selectedIndex = 0;
24
25     // retorna nome de arquivo de imagem ao anúncio atual
26     public String getImage()
27     {
28         return images[ selectedIndex ];
29     } // fim do método getImage
30
```

Retorna o nome do  
arquivo de imagem para  
a imagem da capa do  
livro



```
31 // retorna o URL ao site Web correspondente do anúncio
32 public String getLink()
33 {
34     return links[ selectedIndex ];
35 } // fim do método getLink
36
37 // atualiza selectedIndex assim as próximas chamadas para getImage
38 // getLink retornam um anúncio diferente
39 public void nextAd()
40 {
41     selectedIndex = ( selectedIndex + 1 ) % images.length;
42 } // fim do método nextAd
43 } // fim da classe Rotator
```

Retorna o hyperlink  
para o livro na  
Amazon.com

## Rotator.java

Atualiza Rotator de  
modo que as chamadas  
subseqüentes a `getImage`  
e `getLink` retornem  
informações para  
diferentes anúncios

Linhas 39-42



# Resumo

```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.15: adrotator.jsp -->
6
7 <jsp:useBean id = "rotator" scope = "session"
8   class = "com.deitel.jhttp6.jsp.beans.Rotator" />
9
10 <html xmlns = "http://www.w3.org/1999/xhtml">
11   <head>
12     <title>AdRotator Example</title>
13     <style type = "text/css">
14       .big { font-family: helvetica, arial, sans-serif;
15             font-weight: bold;
16             font-size: 2em }
17     </style>
18     <!-- atualiza anúncio -->
19     <% rotator.nextAd(); %>
20   </head>
21   <body>
22     <p class = "big">AdRotator Example</p>
23     <p>
24       <a href = "<jsp:getProperty name = "rotator"
25         property = "link" />">
26
27         <img src = "<jsp:getProperty name = "rotator"
28           property = "image" />" alt = "advertisement" />
29       </a>
30     </p>
31   </body>
32 </html>

```

Utiliza a ação  
**jsp:useBean** para  
obter a referência ao  
objeto Rotator

adrotator.jsp

(1 de 2)

Linhas 7-8

Linha 19

Linhas 24-29

Invoca o método  
**nextAd** de  
Rotator

Define o hyperlink  
para o site  
Amazon.com

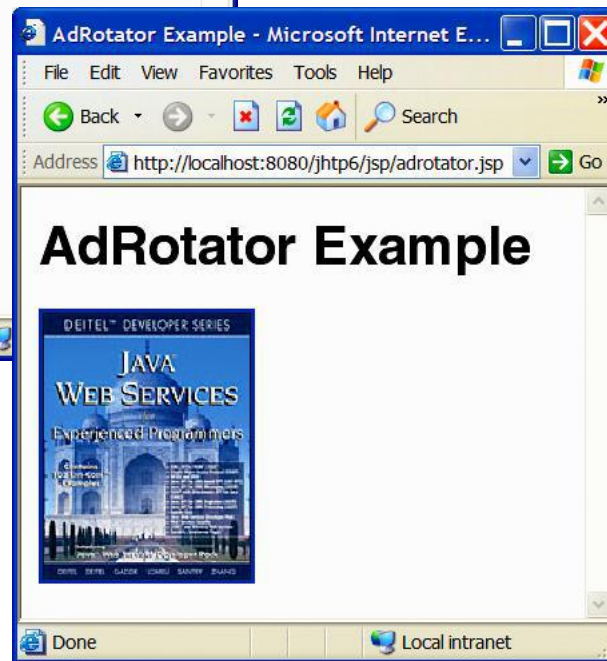


# Resumo

`adrotator.jsp`

(2 de 2)

Saída do programa



## 27.6.3 Ação <jsp:useBean> (Cont.)

- Ação <jsp:getProperty>:
  - Atributo name:
    - Especifica o objeto bean a manipular.
  - Atributo property:
    - Especifica a propriedade a obter.
  - Substitui a ação <jsp:getProperty> por expressões.
    - <%= rotator.getLink() %>
    - <%= rotator.getImage() %>



## Observação de engenharia de software 27.8

---

**A ação `<jsp:setProperty>` pode utilizar valores de solicitação-parâmetro para configurar propriedades JavaBean dos seguintes tipos: `Strings`, tipos primitivos (`boolean`, `byte`, `char`, `short`, `int`, `long`, `float` e `double`) e classes do tipo empacotador (`Boolean`, `Byte`, `Character`, `Short`, `Integer`, `Long`, `Float` e `Double`). Para um exemplo, consulte a Figura 27.22.**





Atributo	Descrição
name	O ID do JavaBean para o qual uma propriedade (ou propriedades) será configurada.
property	O nome da propriedade a configurar. Especificar "*" para esse atributo especifica que a JSP deve corresponder os parâmetros de solicitação às propriedades do bean. Para cada correspondência de parâmetro de solicitação (isto é, o nome do parâmetro de solicitação é idêntico ao nome de propriedade do bean), a propriedade correspondente no bean é configurada como o valor do parâmetro. Se o valor do parâmetro de solicitação for "", o valor de propriedade no bean permanecerá inalterado.
param	Se os nomes de parâmetro de solicitação não corresponderem aos nomes de propriedade de bean, esse atributo pode ser utilizado para especificar que parâmetro de solicitação deve ser utilizado para obter o valor de uma propriedade de bean específica. Esse atributo é opcional. Se esse atributo for omitido, os nomes de parâmetro de solicitação devem corresponder aos nomes de propriedade de bean.
value	O valor a atribuir a uma propriedade de bean. Em geral, o valor é o resultado de uma expressão de JSP. Esse atributo é particularmente útil para configurar propriedades de bean que não podem ser configuradas utilizando parâmetros de solicitação. Esse atributo é opcional. Se esse atributo for omitido, a propriedade do JavaBean deve ser de um tipo que possa ser configurado utilizando parâmetros de solicitação.

**Figura 27.16** | Atributos da ação <jsp:setProperty>.

## Erro comum de programação 27.4

---

**Os erros de conversão ocorrem se você utilizar o atributo `value` da ação `<jsp:setProperty>` para configurar tipos de propriedade JavaBean que não podem ser configurados com parâmetros de solicitação.**



## 27.7 Diretivas

- **Diretivas JSP:**

- Mensagens ao contêiner de JSPs.
- Permitem ao programador:
  - Especificar configurações de página.
  - Incluir o conteúdo de outros recursos.
  - Especificar bibliotecas de tags personalizados.
- Delimitadas por `<%@` e `%>`.



Diretiva	Descrição
<code>page</code>	Define configurações de página para o contêiner de JSP processar.
<code>include</code>	Faz com que o contêiner de JSP realize uma inserção em tempo de tradução de outro conteúdo do recurso. Quando a JSP é traduzida em um servlet e compilada, o arquivo referenciado substitui a diretiva <code>include</code> e é traduzido como se originalmente fizesse parte do JSP.
<code>taglib</code>	Permite aos programadores utilizar novos tags de bibliotecas de tags que encapsulam funcionalidades mais complexas e simplificam a codificação de uma JSP.

**Figura 27.17 | Diretivas de JSP.**

## 27.7.1 Diretiva page

- **Diretiva JSP page:**
  - Especifica as configurações JSP globais no contêiner de JSPs.



## Erro comum de programação 27.5

---

**Fornecer múltiplas diretivas `page` com um ou mais atributos repetidos em comum é um erro em tempo de tradução de JSP, a menos que os valores para todos os atributos repetidos sejam idênticos. Além disso, fornecer uma diretiva `page` com um atributo ou valor que não é reconhecido é um erro em tempo de tradução de JSP.**



# Observação de engenharia de software 27.9

---

**De acordo com a especificação JSP (Seção 1.10.1), o atributo `extends` ‘não deve ser utilizado sem uma cuidadosa consideração, visto que restringe a capacidade do contêiner de JSPs de fornecer superclasses especializadas que podem melhorar a qualidade do serviço renderizado’. Lembre-se de que uma classe Java pode estender exatamente uma outra classe. Se sua JSP especifica uma superclasse explícita, o contêiner de JSPs não pode traduzir sua JSP em uma subclasse de uma das classes de servlet aprimoradas próprias do aplicativo de contêiner.**



Atributo	Descrição
<b>language</b>	A linguagem de criação de scripts utilizada na JSP. Atualmente, o único valor válido para esse atributo é <code>java</code> .
<b>extends</b>	Especifica a classe da qual a JSP traduzida pode herdar. Esse atributo deve ser um nome de classe completamente qualificado.
<b>import</b>	Especifica uma lista separada por vírgulas de nomes de tipo completamente qualificados e/ou pacotes que serão utilizados na JSP atual. Quando a linguagem de criação de scripts para <code>java</code> , a lista de importação padrão é <code>java.lang.*</code> , <code>javax.servlet.*</code> , <code>javax.servlet.jsp.*</code> e <code>javax.servlet.http.*</code> . Se múltiplas propriedades <code>import</code> forem especificadas, os nomes de pacote serão colocados em uma lista pelo contêiner.
<b>session</b>	Especifica se a página participa de uma sessão. Os valores para esse atributo são <code>true</code> (participa de uma sessão — o padrão) ou <code>false</code> (não participa de uma sessão). Quando a página fizer parte de uma sessão, o objeto implícito <code>session</code> está disponível para utilização na página. Caso contrário, <code>session</code> não está disponível e utilizar <code>session</code> no código de script resulta em um erro de tempo de tradução.
<b>buffer</b>	Especifica o tamanho do buffer de saída utilizado com o objeto implícito <code>out</code> . O valor desse atributo pode ser <code>none</code> para nenhum armazenamento em buffer ou um valor como <code>8kb</code> (o tamanho de buffer padrão). A especificação JSP indica que o buffer utilizado deve ter pelo menos o tamanho especificado.

**Figura 27.18 | Atributos da diretiva page. (Parte 1 de 3)**



Atributo	Descrição
<b>autoFlush</b>	Quando configurado como <b>true</b> (o padrão), esse atributo indica que o buffer de saída utilizado com objeto implícito <b>out</b> deve ser esvaziado automaticamente quando o buffer for preenchido. Se configurado como <b>false</b> , ocorre uma exceção se o buffer estourar. O valor desse atributo deve ser <b>true</b> se o atributo de buffer estiver configurado como <b>none</b> .
<b>isThreadSafe</b>	Especifica se a página é segura para thread. Se <b>true</b> (o padrão), a página é considerada segura para thread e pode processar múltiplas solicitações ao mesmo tempo. Se <b>false</b> , o servlet que representa a página implementa a interface <b>java.lang.SingleThreadModel</b> e, apenas uma solicitação pode ser processada por essa JSP por vez. O padrão JSP permite que existam múltiplas instâncias de uma JSP para JSPs que não são seguras para thread. Isso permite ao contêiner tratar solicitações mais eficientemente. Entretanto, isso não garante que os recursos compartilhados entre instâncias JSP sejam acessados de maneira segura para thread.
<b>info</b>	Especifica uma string de informações que descreve a página. Essa string é retornada pelo método <b>getServletInfo</b> do servlet que representa a JSP traduzida. Esse método pode ser invocado pelo objeto <b>page</b> implícito do JSP.
<b>errorPage</b>	Quaisquer exceções na página atual que não são capturadas são enviadas à página de erro para processamento. O objeto implícito <b>exception</b> da página de erro referencia a exceção original.

**Figura 27.18 | Atributos da diretiva page. (Parte 2 de 3.)**

Attribute	Description
<code>isErrorPage</code>	Especifica se a página atual é uma página de erro que será invocada em resposta a um erro em outra página. Se o valor de atributo for <code>true</code> , o objeto implícito <code>exception</code> será criado e referenciará a exceção original que ocorrerem. Se <code>false</code> (o padrão), qualquer utilização do objeto <code>exception</code> na página resultará em um erro de tempo de tradução.
<code>contentType</code>	Especifica o tipo MIME dos dados na resposta ao cliente. O tipo padrão é <code>text/html</code> .
<code>pageEncoding</code>	Especifica a codificação de caractere da página JSP. O valor-padrão é <code>ISO-8859-1</code> .
<code>isELIgnored</code>	Especifica se o contêiner de JSP deve avaliar expressões que utilizam a <a href="#">Expression Language (EL)</a> — um novo recurso no JSP 2.0 que permite aos autores de JSP criar páginas JSP sem script. A EL é em geral utilizada com bibliotecas de tags JSP, que estão além do escopo deste livro. Uma expressão EL tem a forma <code>\${exp}</code> . Se o valor de atributo for <code>true</code> , as expressões EL são ignoradas, o que significa que o contêiner de JSP não avalia as expressões em tempo de tradução. Se <code>false</code> , as expressões EL são avaliadas pelo contêiner de JSP. Para obter informações adicionais sobre EL, visite <a href="http://java.sun.com/developer/EJTechTips/2004/tt0126.html">java.sun.com/developer/EJTechTips/2004/tt0126.html</a>

**Figura 27.18 | Atributos da diretiva page. (Parte 3 de 3.)**



## Erro comum de programação 27.6

---

**Utilizar o objeto implícito `session` da JSP em uma JSP que não tem o atributo `session` da sua diretiva `page` configurado como `true` é um erro em tempo de tradução.**



## 27.7.2 Diretiva `include`

- **Diretiva JSP `include`:**
  - Inclui conteúdo de um outro recurso em tempo de conversão de JSP.
    - Não tão flexível quanto a ação `<jsp:include>`.



# Resumo

includeDirective.jsp

(1 de 3)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.19: includeDirective.jsp -->
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>
9     <title>Using the include directive</title>
10    <style type = "text/css">
11      body
12      {
13        font-family: tahoma, helvetica, arial, sans-serif;
14      }
15      table, tr, td
16      {
17        font-size: .9em;
18        border: 3px groove;
19        padding: 5px;
20        background-color: #dddddd;
21      }
22    </style>
23  </head>
24  <body>
```



# Resumo

includeDirective.jsp

```

25 <table>
26   <tr>
27     <td style = "width: 160px; text-align: center">
28       <img src = "images/logotiny.png"
29         width = "140" height = "93"
30         alt = "Deitel & Associates, Inc. Logo" />
31     </td>
32     <td>
33       <!-- inclui banner.html neste JSP -->
34       <%@ include file = "banner.html" %>
35     </td>
36   </tr>
37   <tr>
38     <td style = "width: 160px">
39       <!-- inclui toc.html neste JSP -->
40       <%@ include file = "toc.html" %>
41     </td>
42     <td style = "vertical-align: top">
43       <!-- inclui clock2.jsp neste JSP -->
44       <%@ include file = "clock2.jsp" %>
45     </td>
46   </tr>
47 </table>
48 </body>
49 </html>

```

Utiliza a diretiva  
**include** para incluir  
banner.html

Utiliza a diretiva  
**include** para incluir  
toc.html

Utiliza a diretiva  
**include** para incluir  
clock2.jsp

Link 40

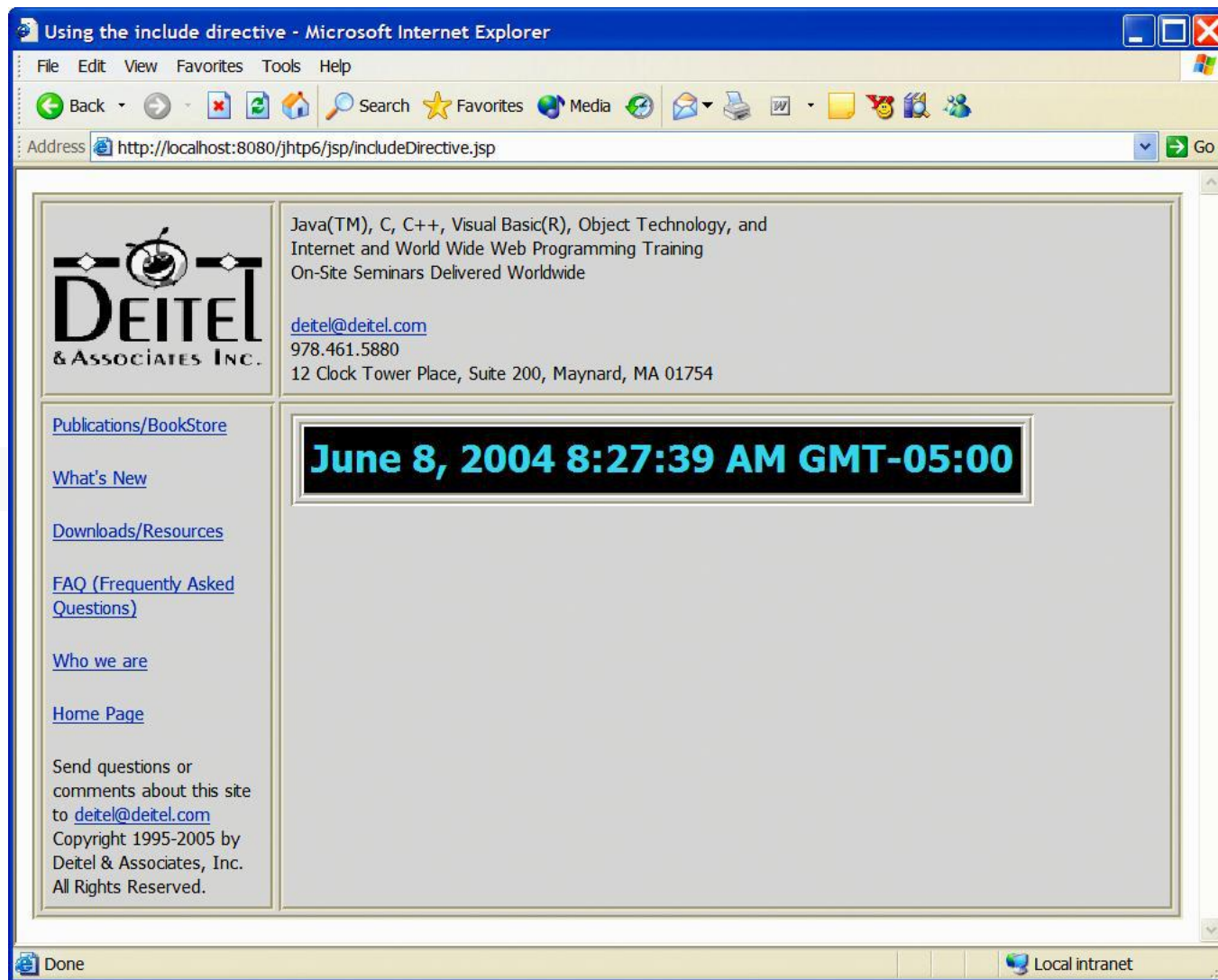


# Resumo

includeDirective.jsp

(3 de 3)

Saída do programa



## 27.8 Estudo de caso: Livro de visitas

- **Demonstra:**
  - Ação `<jsp:setProperty>`.
  - Diretiva JSP `page`.
  - Páginas de erro JSP.
  - Uso do JDBC.





# Resumo

GuestBean.java

GuestBean declara três propriedades de convidado: firstName, lastName e email

Linhas 7-9

```
1 // Fig. 27.20: GuestBean.java
2 // JavaBean para armazenar dados de um convidado no livro de convidados.
3 package com.deitel.jhttp6.jsp.beans;
4
5 public class GuestBean
6 {
7     private String firstName;
8     private String lastName; ←
9     private String email;
10
11     // configura o nome do convidado
12     public void setFirstName( String name )
13     {
14         firstName = name;
15     } // fim do método setFirstName
16
17     // obtém o nome do convidado
18     public String getFirstName()
19     {
20         return firstName;
21     } // fim do método getFirstName
22
```



# Resumo

## GuestBean.java

(2 de 2)

```
23 // configura o sobrenome do convidado
24 public void setLastName( String name )
25 {
26     lastName = name;
27 } // fim do método setLastName
28
29 // obtém o sobrenome do convidado
30 public String getLastName()
31 {
32     return lastName;
33 } // fim do método getLastName
34
35 // configura o endereço de correio eletrônico do convidado
36 public void setEmail( String address )
37 {
38     email = address;
39 } // fim do método setEmail
40
41 // obtém o endereço de correio eletrônico do convidado
42 public String getEmail()
43 {
44     return email;
45 } // fim do método getEmail
46 } // fim da classe GuestBean
```



# Resumo

GuestDataBean.java

(1 de 3)

Linha 19

Linha 22

Linha 23

```

1 // Fig. 27.21: GuestDataBean.java
2 // Classe GuestDataBean faz uma conexão de banco de dados e suporta
3 // inserir dados no banco de dados e recuperá-los.
4 package com.deitel.jhttp6.jsp.beans;
5
6 import java.sql.SQLException;
7 import javax.sql.rowset.CachedRowSet;
8 import java.util.ArrayList;
9 import com.sun.rowset.CachedRowSetImpl; // implementação CachedRowSet
10
11 public class GuestDataBean
12 {
13     private CachedRowSet rowSet;
14
15     // constrói objeto GuestDataBean
16     public GuestDataBean() throws Exception
17     {
18         // carrega o driver MySQL
19         Class.forName( "com.mysql.jdbc.Driver" );
20
21         // especifica propriedades de CachedRowSet
22         rowSet = new CachedRowSetImpl();
23         rowSet.setUrl( "jdbc:mysql://localhost/guest" );
24         rowSet.setUsername( "jhttp6" );
25         rowSet.setPassword( "jhttp6" );
26

```

Carrega o driver MySQL

24

Cria um objeto **CachedRowSet**

utilizando

Configura a propriedade  
de banco de dados do  
**CachedRowSet**

Configura a propriedade de  
senha de banco de dados do  
**CachedRowSet**



```

27 // obtém lista de títulos
28 rowSet.setCommand(
29     "SELECT firstName, lastN
30 rowSet.execute(); ←
31 } // fim do construtor GuestDa

```

Executa a consulta  
especificada pela  
propriedade do comando

Configura a propriedade do  
comando de banco de dados  
do CachedRowSet

GuestDataBean.java

(2 de 3)

```

33 // retorna um ArrayList de GuestBeans
34 public ArrayList< GuestBean > getGuestList() throws SQLException
35 {
36     ArrayList< GuestBean > guestList = new ArrayList< GuestBean >();
37
38     rowSet.beforeFirst(); // move cursor antes da primeira linha
39
40     // obtém dados da linha
41     while ( rowSet.next() ) ←
42     {
43         GuestBean guest = new GuestBean();
44
45         guest.setFirstName( rowSet.getString( 1 ) );
46         guest.setLastName( rowSet.getString( 2 ) );
47         guest.setEmail( rowSet.getString( 3 ) );
48
49         guestList.add( guest );
50     } // fim do while
51
52     return guestList;
53 } // fim do método getGuestList
54

```

Move o cursor do  
CachedRowSet para antes  
da primeira linha

Cria objetos GuestBean para cada  
linha no CachedRowSet

Linha 38

Linhas 41-50



# Resumo

GuestDataBean.java

```

55 // insere um convidado no banco de dados guestbook
56 public void addGuest( GuestBean guest ) throws SQLException
57 {
58     rowSet.moveToInsertRow(); // move cursor para a linha a ser inserida
59
60     // atualiza as três colunas da linha a ser inserida
61     rowSet.updateString( 1, guest.getFirstN );
62     rowSet.updateString( 2, guest.getLastN );
63     rowSet.updateString( 3, guest.getEmail() );
64     rowSet.insertRow(); // insere linha a ser inserida no banco de dados
65     rowSet.moveToCurrentRow(); // move cursor para a linha atual
66     rowSet.acceptChanges(); // propaga alterações para o banco de dados
67 } // fim do método addGuest
68 } // fim da classe GuestDataBean
  
```

Invoca o método  
moveToInsertRow do  
CachedRowSet para se lembrar  
da linha atual e move o cursor  
para a linha

Invoca o método  
updateString de

Invoca o método insertRow

Invoca o método

Invoca o método acceptChanges de  
CachedRowSet para propagar as alterações no  
rowset para o banco de dados subjacente

Linha 64

Linha 65

Linha 66



# Resumo

bookLogin.jsp

```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.22: guestBookLogin.jsp -->
6
7 <!-- configurações de página -->
8 <%@ page errorPage = "guestBookErrorPage.jsp" %>
9
10 <!-- beans usados neste JSP -->
11 <jsp:useBean id = "guest" scope = "page"
12   class = "com.deitel.jhttp6.jsp.beans.GuestBean" />
13 <jsp:useBean id = "guestData" scope = "request"
14   class = "com.deitel.jhttp6.jsp.beans.GuestDataBean" />
15
16 <html xmlns = "http://www.w3.org/1999/xhtml">
17 <head>
18   <title>Guest Book Login</title>
19   <style type = "text/css">
20     body
21     {
22       font-family: tahoma, helvetica, arial, sans-serif;
23     }
24     table, tr, td
25     {
26       font-size: .9em;
27       border: 3px groove;
28       padding: 5px;
29       background-color: #dddddd;
30     }
31   </style>
32 </head>
33 <body>

```

Diretiva **page** define as informações globalmente disponíveis na JSP

Utiliza as ações **jsp:useBean** para obter referências a objetos **GuestBean** e **GuestDataBean**

Linhas 11-14



```

34 <jsp:setProperty name = "guest" property = "*" />
35 <% // inicia scriptlet
36 if ( guest.getFirstName() == null ||
37     guest.getLastName() == null ||
38     guest.getEmail() == null )
39 {
40 <%-- fim do scriptlet para inserir dados de template
41 <form method = "post" action = "guestBookLogin.js
42     <p>Enter your first name, last name and email
43         address to register in our guest book.</p>
44     <table>
45         <tr>
46             <td>First name</td>
47             <td>
48                 <input type = "text" name = "firstName" />
49             </td>
50         </tr>
51         <tr>
52             <td>Last name</td>
53             <td>
54                 <input type = "text" name = "lastName" />
55             </td>
56         </tr>
57         <tr>
58             <td>Email</td>
59             <td>
60                 <input type = "text" name = "email" />
61             </td>
62         </tr>

```

Configura as propriedades de GuestBean com valores de

Verifica se o usuário preencheu todas as entradas, incluindo o primeiro nome, sobrenome e correio eletrônico

citacão, pois put têm os es que as GuestBean

3)

Linha 34

Linhas 36-89



# Resumo

guestBookLogin.jsp

(3 de 3)

```

63         <tr>
64             <td colspan = "2">
65                 <input type = "submit" value = "Submit" />
66             </td>
67         </tr>
68     </table>
69 </form>
70 <% // continua scriptlet
71     } // fim do if
72     else
73     {
74         guestData.addGuest( guest );
75     } <!-- fim do scriptlet para inserir ação jsp
76         <!-- encaminha para exibir conteúdo de livro de convidado --%>
77         <jsp:forward page = "guestBookView.jsp" />
78     <% // continua scriptlet
79     } // fim do else
80 <%> <!-- fim do scriptlet --%>
81 </body>
82 </html>

```

Adiciona o visitante GuestBean ao  
GuestDataBean guestData

Linha 77

Encaminha a solicitação ao  
guestBookView.jsp





## 27.8 Estudo de caso: Livro de visitas (*Continuação*)

- **Linha 34:**

- Especifica ‘\*’ para o atributo property.
- Corresponde os parâmetros de solicitação com as propriedades.
- Pode configurar as propriedades individualmente.
  - `<jsp:setProperty name = “guest” property = “firstName”  
param = “firstName” />`
  - `<jsp:setProperty name = “guest” property = “lastName”  
param = “lastName” />`
  - `<jsp:setProperty name = “guest” property = “email”  
param = “email” />`
  - `<% guest.setFirstName(  
request.getParameter( “firstName” ) ); %>`



# Resumo

```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.23: guestBookView.jsp -->
6
7 <!-- configurações de página --%>
8 <%@ page errorPage = "guestBookErrorPage.jsp" %>
9 <%@ page import = "java.util.*" %>
10 <%@ page import = "com.deitel.jhttp6.jsp.beans.*" %>
11
12 <!-- GuestDataBean para obter lista de convidados --%>
13 <jsp:useBean id = "guestData" scope = "request"
14   class = "com.deitel.jhttp6.jsp.beans.GuestDataBean" />
15
16 <html xmlns = "http://www.w3.org/1999/xhtml">
17   <head>
18     <title>Guest List</title>
19     <style type = "text/css">
20       body
21       {
22         font-family: tahoma, helvetica, arial, sans-serif;
23       }
24       table, tr, td, th
25       {
26         text-align: center;
27         font-size: .9em;
28         border: 3px groove;
29         padding: 5px;
30         background-color: #dddddd;
31       }
32     </style>
33   </head>

```

Utiliza a diretiva **page import** para especificar classes e pacotes Java utilizados no contexto da JSP

Utiliza a ação **jsp:useBean** para obter uma referência a **GuestDataBean**

Linhas 13-14



# Resumo

guestBookView.jsp

(2 de 2)

Linhas 45-64

Utiliza um scriptlet e expressões para exibir o sobrenome, primeiro nome e endereço de correio eletrônico para todos os convidados

```

34 <body>
35   <p style = "font-size: 2em;">Guest List</p>
36   <table>
37     <thead>
38       <tr>
39         <th style = "width: 100px;">Last name</th>
40         <th style = "width: 100px;">First name</th>
41         <th style = "width: 200px;">Email</th>
42       </tr>
43     </thead>
44     <tbody>
45       <% // inicia o scriptlet
46         List guestList = guestData.getGuestList();
47         Iterator guestListIterator = guestList.iterator();
48         GuestBean guest;
49
50         while ( guestListIterator.hasNext() )
51         {
52           guest = ( GuestBean ) guestListIterator.next();
53           <%-- fim do scriptlet; insere dados da template fixa --%>
54           <tr>
55             <td><%= guest.getLastName() %></td>
56             <td><%= guest.getFirstName() %></td>
57             <td>
58               <a href = "mailto:<%= guest.getEmail() %>">
59                 <%= guest.getEmail() %></a>
60             </td>
61           </tr>
62           <% // continua o scriptlet
63             } // fim do while
64           <%-- fim do scriptlet --%>
65         </tbody>
66       </table>
67     </body>
68 </html>

```

# Resumo

Utiliza a diretiva `page` `isErrorPage` para especificar que `guestBookError-Page` é uma página de erro

(1 de 3)

Linha 8

```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 27.24: guestBookErrorPage.jsp -->
6
7 <!-- configurações de página --%>
8 <%@ page isErrorPage = "true" %>
9 <%@ page import = "java.util.*" %>
10 <%@ page import = "java.sql.*" %>
11
12 <html xmlns = "http://www.w3.org/1999/xhtml">
13   <head>
14     <title>Error!</title>
15     <style type = "text/css">
16       .bigRed
17       {
18         font-size: 2em;
19         color: red;
20         font-weight: bold;
21       }
22     </style>
23   </head>

```



# Resumo

questBookErrorPage

Utiliza um objeto **exception** implícito para determinar o erro a ser exibido

As linhas 28 e 36

```
24 <body>
25   <p class = "bigRed">
26     <% // scriptlet para determinar tipo de exceção
27         // e gera saída do começo da mensagem de erro
28         if ( exception instanceof SQLException )
29     {
30     <%
31
32         A SQLException
33
34     <%
35     } // fim do if
36     else if ( exception instanceof ClassNotFoundException )
37     {
38     <%
39
40         A ClassNotFoundException
41
42     <%
43     } // fim do else if
44     else
45     {
46     <%
47
```



An exception

```
48
49
50 <%
51   } // fim do else
52 %>
53 <%-- fim do scriptlet para inserir dados de template fixa --%>
54
55   <%-- continua saída de mensagem de erro --%>
56   occurred while interacting with the guestbook database.
57 </p>
58 <p class = "bigRed">
59   The error message was:<br />
60   <%= exception.getMessage() %>
61 </p>
62 <p class = "bigRed">Please try again later</p>
63 </body>
64 </html>
```

Exibe a mensagem do erro real  
a partir da exceção

## Resumo

guestBookErrorPage  
.jsp

(3 de 3)

Linha 60



Guest Book Login - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <http://localhost:8080/jhttp6/jsp/guestBookLogin.jsp> Go

Enter your first name, last name and email address to register in our guest book.

First name	<input type="text" value="Paul"/>
Last name	<input type="text" value="Deitel"/>
Email	<input type="text" value="deitel@deitel.com"/>
<input type="button" value="Submit"/>	

Done Local intranet

Guest List - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <http://localhost:8080/jhttp6/jsp/guestBookLogin.jsp> Go

## Guest List

Last name	First name	Email
Deitel	Paul	<a href="mailto:deitel@deitel.com">deitel@deitel.com</a>

Done Local intranet

**Figura 27.25 | Janelas de saída do exemplo do livro de convidados JSP. (Parte 1 de 3.)**

Guest Book Login - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <http://localhost:8080/jhttp6/jsp/guestBookLogin.jsp> Go

Enter your first name, last name and email address to register in our guest book.

First name	Joe
Last name	Doe
Email	doe@bug2bug.com
<input type="button" value="Submit"/>	

Done Local intranet

Guest List - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <http://localhost:8080/jhttp6/jsp/guestBookLogin.jsp> Go

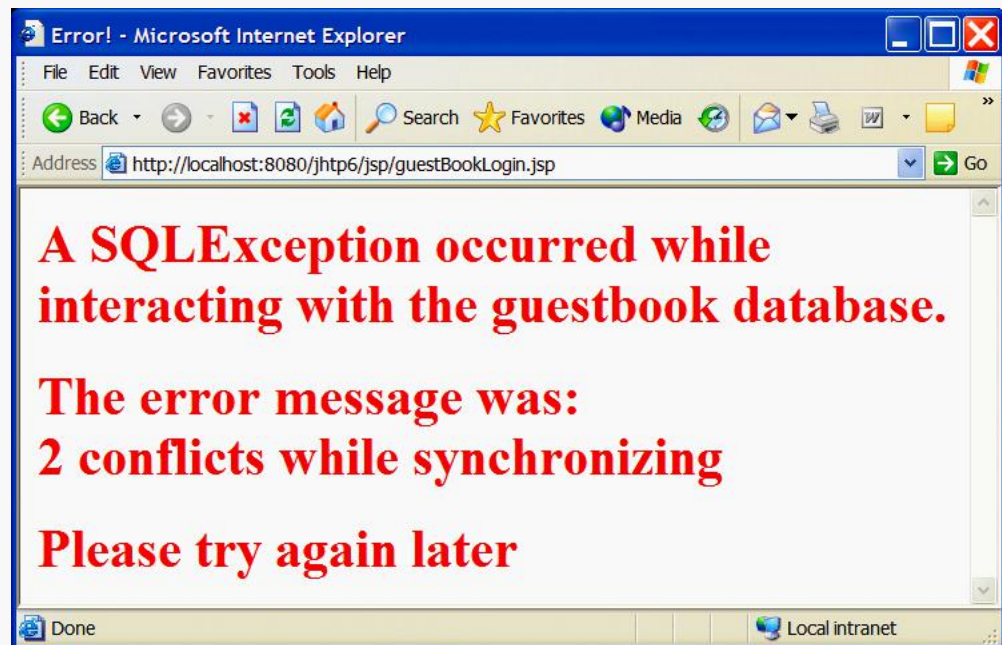
## Guest List

Last name	First name	Email
Doe	Joe	<a href="mailto:doe@bug2bug.com">doe@bug2bug.com</a>
Deitel	Paul	<a href="mailto:deitel@deitel.com">deitel@deitel.com</a>

Done Local intranet

**Figura 27.25 | Janelas de saída do exemplo de livro de convidado JSP. (Parte 2 de 3.)**





**Figura 27.25** | Janelas de saída do exemplo de livro de convidados JSP. (Parte 3 de 3.)