

JavaServer Pages

Agenda

- Introdução
- Que são JavaServer Pages (JSP)
- Container JSP e Container Servlet
- Elementos sintácticos
- Directivas
- Acções
- Elementos de script
- Objectos Java
- Vida dos objectos
- Objectos implícitos
- O pacote `javax.servlet.jsp`
- Objecto implícito `REQUEST`; `RESPONSE`; `PAGECONTEXT`; `SESSION`; `APPLICATION`; `OUT`; `EXCEPTION`; `PAGE` e `CONFIG`
- Comentários
- Caracteres de escape

Introdução

- JavaServer Pages
 - Extensão da tecnologia **Servlet**
- Permite enviar conteúdos para a Web
- Permite a reutilização de componentes JAVA
 - Não é preciso para tal programar em Java
- Permite Cria Marcas personalizadas
 - Permite encapsular funcionalidades complexas
- Classes e interfaces específicas do JSP
 - Package **javax.servlet.jsp**
 - Package **javax.servlet.jsp.tagext**

JSP (Java Server Pages)

- Tecnologia Java para geração de conteúdo dinâmico
- O texto HTML é escrito junto com as tags JSP e código Java
- Não é uma ideia nova sendo usado em tecnologias concorrentes: ASP, PHP, Server-Side JavaScript e Cold Fusion

Servlets & JSP

- **JSP**

- Semelhante com HTML ou XHTML
 - A maior parte é Composta por Marcas HTML ou XHTML
 - Também conhecido por dados de template
- Deve ser usada quando o conteúdo é composto sobre tudo por “dados de template”
 - Pouco o conteúdo gerado dinamicamente

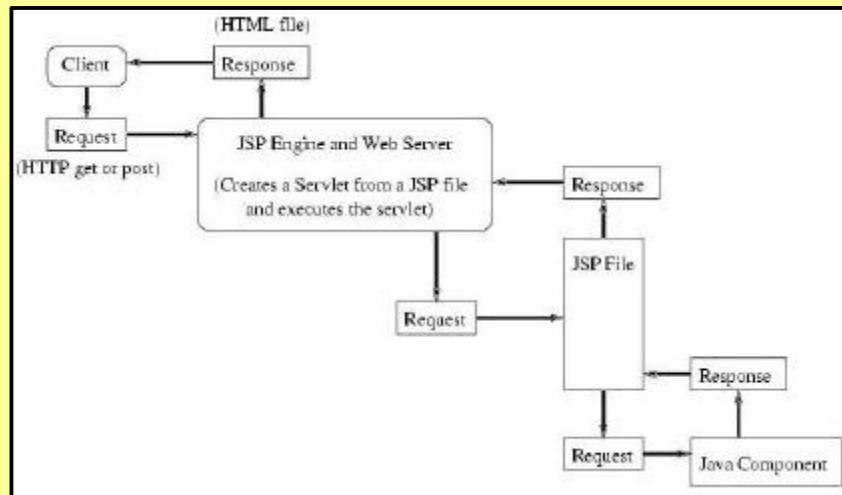
- **Servlets**

- Usado quando os dados de template são em reduzida quantidade
 - A maior parte do conteúdo é gerado dinamicamente

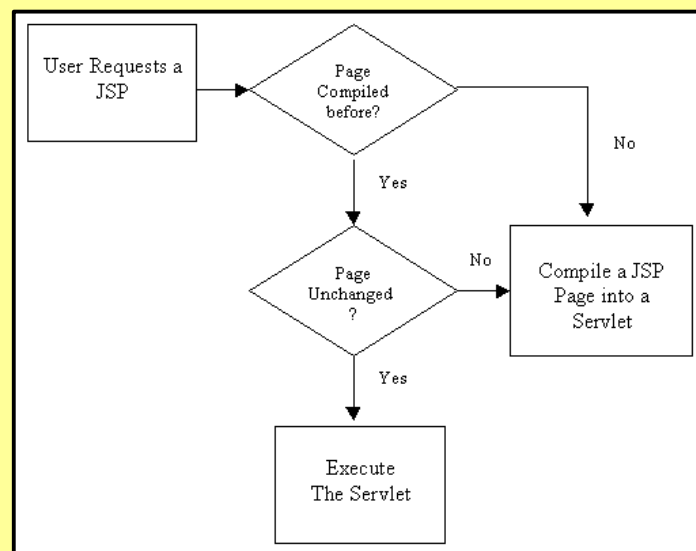
Ciclo de Vida JSP

- Páginas JSP também necessitam de Servidores específicos.
- Em geral, um servidor de **servlets** também é um servidor de **JSP**.
- Para disponibilizar um arquivo **JSP** basta gravá-lo em qualquer pasta visível do servidor com extensão **jsp**.
- Quando um utilizador faz uma requisição a uma página JSP:
 - Se for a primeira, a página **jsp** é convertida em um **servlet** e compilada.
 - Este **servlet** é invocado e retorna como resposta uma **string** HTML.
 - Esta **string** é retornada para o **cliente**.
- Todo o processamento ocorre no servidor, apenas o Html é retornado.

Funcionamento de uma JSP



Funcionamento de uma JSP (cont.)



Primeiro Exemplo

```
<html>
  <head>
    <%-- JSP Declarations --%>
    <%! String world; %>

    <!-- JSP Code -->
    <% world = "World"; %>
    <title>Hello <%= world %></title>
  </head>
  <body>
    Hello, <%= world %>.
  </body>
</html>
```

Sintaxe do JSP

Componentes do JSP

Comentários: `<!-- -->`; `<% -- ... --%>`

- Vários níveis de comentários: HTML; JSP e Servlets

Directivas: `<%@directiva Lista-atributos %>`

- Mensagens enviadas ao programa que compila/executa **JSPs** (container)
- Directivas: `page`; `include`; `taglib`

Acções `<jsp:comando> ... </jsp:comando>`

- Marcas JSP pré-definidas que encapsulam funcionalidades.
- Comandos: `jsp:useBean`; `jsp:setProperty`; `jsp:getProperty`; `jsp:param`; `jsp:include`; `jsp:forward`; `jsp:plugin`; `jsp:params`; `jsp:fallback`

Scripting: `<% ... %>`

- Permite a escrita de código Java nas JSP.
- `<% código %>`, `<%! Declaração %>`, and `<%= expressão %>`

Comentários

- Comentários ao nível do HTML
 - **Sintaxe:** `<!-- comentário -->`
 - O comentário é visível pelo utilizador da página quando este visualiza o conteúdo da mesma (comentário ao nível do HTML)
- Comentários ao nível do JSP
 - **Sintaxe:** `<%-- comentário --%>`
 - O comentário é visível somente na JSP e na servlet que traduz a JSP (comentário em nível de servidor)

Comentários ao nível do HTML

- Comentário é visível na pagina HTML enviada ao cliente

`<!-- comment <% expression%> -->`

Example:

`<!-- this is just Html comment -->`

`<!-- This page was loaded on <%= (new java.util.Date()).toLocaleString()%> -->`

View source:

`<!-- this is just Html comment -->`

`<!-- This page was loaded on January 1, 2000 -->`

Comentários ao nível do JSP

- Comentário não é visível na pagina HTML enviada ao cliente

`<% -- expression -- %>`

Example:

`<html>`

`<body>`

`<h2>A Test of Comments</h2>`

`<%--This comment will be invisible in page source --%>`

`</body>`

`</html>`

Exemplo JavaServer Page

- Mostra
 - Dados de Template (XHTML markup)
 - Criação de um Objecto Java (**java.util.Date**)
 - Conversão automática de uma expressão JSP para uma String
 - **Uso da Marca meta** para refrescar a página
- Primeira chamada de **clock.jsp**
 - Tempo de espera enquanto:
 - JSP container traduz the JSP numa servlet
 - JSP container compila a servlet
 - JSP container executa a servlet
 - Chamada seguintes são mais rápidas

```
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. 10.1: clock.jsp -->
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8
9   <head>
10     <meta http-equiv = "refresh" content = "60" />
11
12     <title>A Simple JSP Example</title>
13
14     <style type = "text/css">
15       .big { font-family: helvetica, arial, sans-serif;
16             font-weight: bold;
17             font-size: 2em; }
18     </style>
19   </head>
20
21   <body>
22     <p class = "big">Simple JSP Example</p>
23
24     <table style = "border: 6px outset;">
25       <tr>
26         <td style = "background-color: black;">
27           <p class = "big" style = "color: cyan;">
28
29             <!-- JSP expression to insert date/time -->
30             <%= new java.util.Date() %>
31
32           </p>
33         </td>
34       </tr>
35     </table>
```

Uso de uma expressão JSP para inserir a Data e a Hora numa página Web

Elemento **meta** refresca a página todo os 60 segundos

Cria um objecto **Date** que é convertido para uma **String** implicitamente e é incluído num elemento do tipo paragrafo (**p**)


```

36     </body>
37
38     </html>

```

Uso de uma expressão JSP para inserir a Data e a Hora numa página Web

Saída do Programa

String que representa o objecto Date aparece aqui

Objectos Implícitos

- **Nível da Aplicação**
 - Objectos mantidos pelo container
 - Qualquer servlet ou JSP pode usar estes objectos
- **Nível da Página**
 - Objectos que só existem na página em que são definidos
 - Cada página possui uma instancia destes objectos
- **Nível de Pedido (Request scope)**
 - Objectos existem durante o envio de pedidos pelo cliente
- **Nível de Sessão (Session scope)**
 - Objectos existem durante a sessão do cliente
 - Estes objectos são válidos para às páginas que fazem parte de uma sessão

Objectos Implícitos (cont.)

| Object | |
|---------------------------|--|
| <i>Nível de Aplicação</i> | |
| application | <code>javax.servlet.ServletContext</code> |
| <i>Nível de página</i> | |
| config | <code>javax.servlet.ServletConfig</code> |
| exception | <code>java.lang.Throwable</code> |
| out | <code>javax.servlet.jsp.JspWriter.</code> |
| page | <code>java.lang.Object</code> referencia <code>this</code> para a instancia. |
| pageContext | <code>javax.servlet.jsp.PageContext.</code> |
| | |

Objectos Implícitos (cont.)

| Objectos Implícitos | Descrição |
|------------------------|--|
| response | <code>HttpServletResponse .</code> |
| <i>Nível de Pedido</i> | |
| request | <code>HttpServletRequest.</code> |
| <i>Nível de sessão</i> | |
| session | <code>javax.servlet.http.HttpSession.</code> |
| | |

Scripting

- O modo como os programadores podem inserir código numa página.
- Neste momento só é possível introduzir código em Java.
- o código contido no **scriptlet** é executado quando o pedido é executado pela página JSP
- o **scriptlet** pode possuir uma variedade de comandos, declarações de variáveis e métodos ou expressões válidas
- Todas as marcas HTML devem ser declarados fora do **scriptlet**.

Componentes de Scripting

- componentes de scripting
 - Scriptlets (blocos de código em Java limitados por `<% e %>`)
 - Comentários (`<%-- e --%>`) (não pode seu usados nos Scriptlets)
 - Expressões (`<%= e %>`) (contêm uma expressão em Java que é avaliada quando a página é executada). É feita uma conversão implícita da expressão para uma objecto do tipo String
 - Declarações (permite ao programador definir variáveis e métodos).
 - As variáveis são convertidas em variáveis de instancia da classe que traduz a jsp.
 - Os métodos são convertidos em métodos da classe que traduz a jsp.
 - Sequencia de Escape (permite incluir caracteres que possuem um significado especial para as JSP)

Componentes de Scripting (cont.)

| Literal | Sequencia de Escape |
|---------|---------------------|
| <% | <\% |
| %> | %\> |
| ' | \' |
| " | \" |
| \ | \\ |
| | |
| | |

```

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. 10.4: welcome.jsp -->
6  <!-- JSP that processes a "get" request containing data. -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9
10     <!-- head section of document -->
11     <head>
12         <title>Processing "get" requests with data</title>
13     </head>
14
15     <!-- body section of document -->
16     <body>
17         <% // begin scriptlet
18
19         String name = request.getParameter( "firstName" );
20                                     //Objecto HttpServletRequest
21
22         if ( name != null ) {
23             <%-- end scriptlet to insert fixed template data --%>
24
25             <h1>
26                 Hello <%= name %>, <br />
27                 Welcome to JavaServer Pages!
28             </h1>
29
30             <% // continue scriptlet
31
32             } // end if

```

Scriptlets usado
para inserir código
Java

Uso do objecto
request para ler um
parametro

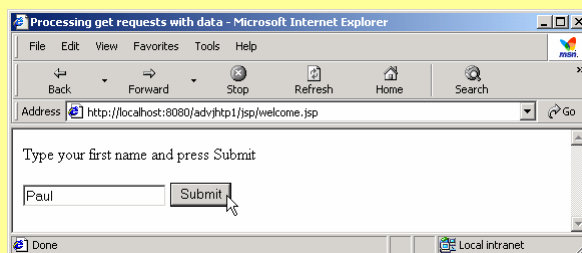
Declaração JSP

```

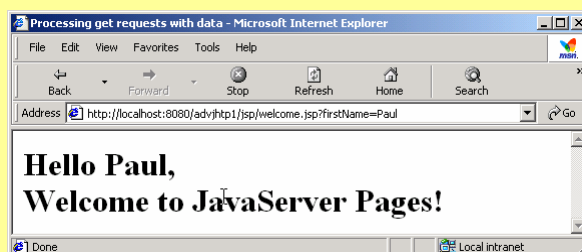
33     else {
34
35         %> <!-- end scriptlet to insert fixed template data --%>
36
37         <form action = "welcome.jsp" method = "get">
38             <p>Type your first name and press Submit</p>
39
40             <p><input type = "text" name = "firstName" />
41               <input type = "submit" value = "Submit" />
42             </p>
43         </form>
44
45         <% // continue scriptlet
46
47     } // end else
48
49     %> <!-- end scriptlet --%>
50 </body>
51
52 </html> <!-- end XHTML document -->

```

Scriptlets usado
para inserir
código Java



Saída das páginas



Acções Standard

- Proporcionam acesso a tarefas comuns realizadas num JSP
 - **Incluir** conteúdos de outros recursos
 - **Redirecionar** os pedidos para outros recursos
 - **Interagir** com **JavaBeans**
- JSP containers processam as acções quando são solicitadas
- Delimitados pelas marcas **<jsp:action>** e **</jsp:action>** em que **action** é a acção standard

Acções Standard

| Acção | Descrição |
|----------------------------|--|
| <jsp:include> | Inclui Dinamicamente outro recurso num JSP. Quando a JSP é executada o recurso referenciado é incluído e processado. Se o recurso é alterado entre pedidos as alterações são incluídas na próxima chamada da página |
| <jsp:forward> | Redireciona o processamento para outro JSP, servlet ou página estática. Esta acção termina com a execução da página JSP corrente. |
| <jsp:plugin> | Permite gerar uma página HTML que contenha as construções apropriadas (dependentes do browser) para que se faça o download do plug-in Java e a subsequente execução da applet especificada A tag <jsp:plugin> é substituída pela tag <object> ou pela tag <embed> , dependendo do browser do cliente |
| <jsp:param> | Usado em conjunto com as acções include , forward and plugin . Permite definir duplos do tipo (nome; valor) que podem ser lidos quando a acção é executada |
| | |

Acções Standard (cont.)

| Acção | Descrição |
|--------------------------------------|--|
| <i>JavaBean Manipulation</i> | |
| <code><jsp:useBean></code> | Associa uma instância de um JavaBean, definido com um determinada abrangência e com um id. |
| <code><jsp:setProperty></code> | Modifica o valor de uma propriedade da instância do JavaBean. Possui uma funcionalidade especial a qual permite modificar o atributo pelo uso de um parametro lido num pedido, desde que seja utilizado o mesmo identificador. |
| <code><jsp:getProperty></code> | Obtem e converte uma propriedade numa String de modo a ser possível incluir na resposta. |

Acção `<jsp:include>`

Acção `<jsp:include>` action

- Permite incluir conteúdo dinâmico numa **JSP**
- Mais flexível do que a directiva **include**

```

1  <!-- Fig. 10.7: banner.html          -->
2  <!-- banner to include in another document -->
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&nbsp;<br />
8          On-Site Seminars Delivered Worldwide
9      </p>
10
11     <p>
12         <a href = "mailto:geral@est.ips.pt">
13             deitel@deitel.com</a><br />
14
15             978.579.9911<br />
16             490B Boston Post Road, Suite 200,
17             Sudbury, MA 01776
18         </p>
19 </div>

```

Banner (banner.html) que deve ser incluído em todos os documentos XHTML criados dinamicamente

```

1  <!-- Fig. 10.8: toc.html              -->
2  <!-- contents to include in another document -->
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 1995-2002 by Deitel & Associates, Inc.
33     All Rights Reserved.
34 </p>

```

Índice (toc.html) que vai ser incluído no documento final


```

1  <!-- Fig. 10.9: clock2.jsp -->
2  <!-- date and time to include in another document -->
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 to determine client local and -->
11                 <!-- format date accordingly -->
12                 <%
13                     // get client locale
14                     java.util.Locale locale = request.getLocale();
15
16                     // get DateFormat for client's Locale
17                     java.text.DateFormat dateFormat =
18                         java.text.DateFormat.getDateInstance(
19                             java.text.DateFormat.LONG,
20                             java.text.DateFormat.LONG, locale );
21
22                 <%-- end script -->
23
24                 <!-- output date -->
25                 <%= dateFormat.format( new java.util.Date() ) %>
26             </p>
27         </td>
28     </tr>
29 </table>

```

JSP clock2.jsp mostra um relógio na parte principal do documento XHTML final

Uso de **Locale** para formatar a Data de acordo com a configuração do cliente

```

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. 10.7: 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                 font-family: tahoma, helvetica, arial, sans-serif;
15             }
16
17             table, tr, td {
18                 font-size: .9em;
19                 border: 3px groove;
20                 padding: 5px;
21                 background-color: #dddddd;
22             }
23         </style>
24     </head>
25
26     <body>
27         <table>
28             <tr>
29                 <td style = "width: 160px; text-align: center">
30                     <img src = "images/logotiny.png"
31                         width = "140" height = "93"
32                         alt = "Deitel & Associates, Inc. Logo" />
33                 </td>
34             </tr>

```

JSP include.jsp Liga os documentos pelo uso da ação <jsp: include> (Parte 1).

```

35         <td>
36
37             <!-- include banner.html in this JSP -->
38             <jsp:include page = "banner.html"
39                 flush = "true" />
40
41         </td>
42     </tr>
43
44     <tr>
45         <td style = "width: 160px">
46
47             <!-- include toc.html in this JSP -->
48             <jsp:include page = "toc.html" flush = "true" />
49
50         </td>
51
52         <td style = "vertical-align: top">
53
54             <!-- include clock2.jsp in this JSP -->
55             <jsp:include page = "clock2.jsp"
56                 flush = "true" />
57
58         </td>
59     </tr>
60 </table>
61 </body>
62 </html>

```

Inclui banner.html

Liga os documentos pelo uso da acção

inclui toc.html

inclui clock2.jsp

Documento final

Acção<jsp:forward>

• Acção<jsp:forward>

- É utilizada para transferir o controle de uma página JSP para um outro recurso no mesmo contexto
- todo conteúdo gerado pela página actual é descartado e o processamento começa novamente no recurso alternativo

• Acção <jsp:param>

- Opcionalmente, é possível especificar parâmetros na execução de um <jsp:forward>
- estes parâmetros serão recebidos pelo novo recurso

```
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. 10.11: forward1.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8
9  <head>
10   <title>Forward request to another JSP</title>
11 </head>
12
13 <body>
14   <% // begin scriptlet
15
16       String name = request.getParameter( "firstName" );
17
18       if ( name != null ) {
19
20   %> <!-- end scriptlet to insert fixed template data --%>
21
22       <jsp:forward page = "forward2.jsp">
23         <jsp:param name = "date"
24           value = "<%= new java.util.Date() %>" />
25       </jsp:forward>
26
27   <% // continue scriptlet
28
29       } // end if
30   else {
31
32   %> <!-- end scriptlet to insert fixed template data --%>
33
```

JSP forward1.jsp
recebe o parâmetro
firstName inclui um
parâmetro date e
redireciona o
processamento para
to forward2.jsp
(Parte 1).

Linhas 22-25

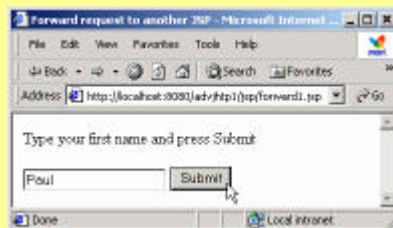
Processamento
passa para
forward2.jsp

```

34     <form action = "forward1.jsp" method = "get">
35         <p>Type your first name and press Submit</p>
36
37         <p><input type = "text" name = "firstName" />
38             <input type = "submit" value = "Submit" />
39         </p>
40     </form>
41
42     <% // continue scriptlet
43
44     } // end else
45
46     %> <%-- end scriptlet --%>
47 </body>
48
49 </html> <!-- end XHTML document -->

```

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

```

JSP **forward2.jsp** recebe o comando de **forward1.jsp** e faz uso dos parametros para compor a resposta a enviar ao cliente (Parte 1).

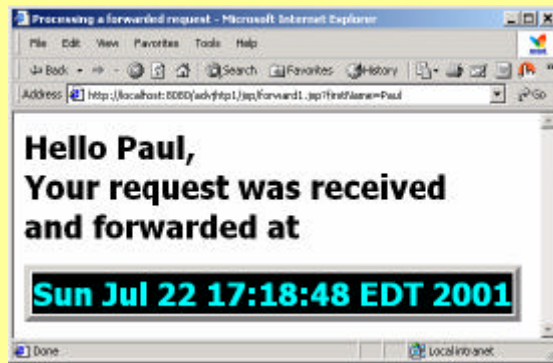
Linhas 23-24

Recebe request de **forward1.jsp**, e obtém parâmetro **firstName**

Lê o parâmetro **date** a partir do objecto request

```
36 </body>
37
38 </html>
```

(Parte 2).



Acção <jsp:plugin>

- Adiciona uma applet ou um JavaBean a uma página
- Permite fazer o download e a instalação do Plug-in Java
- A tag <jsp:plugin> é substituída pela Marca <object> ou pela Marca <embed>, dependendo do browser do cliente

Acção <jsp:plugin> (cont.)

- A sintaxe básica é:

```
<jsp:plugin type="tipo do plugin"
           code="arquivo da classe"
           codebase="caminho da classe"
           <jsp:params>
             parâmetros
           </jsp:params>
</jsp:plugin>
```

- Exemplo:

```
<jsp:plug-in type="applet" code="Applet1.class"
             codebase="/caminho/seila"
             width="400"
             height="300"
</jsp:plugin>
```

```
1  // Fig. 10.14: ShapesApplet.java
2  // Applet that demonstrates a Java2D GeneralPath.
3  package com.deitel.advjhtml.jsp.applet;
4
5  // Java core packages
6  import java.applet.*;
7  import java.awt.event.*;
8  import java.awt.*;
9  import java.awt.geom.*;
10
11 // Java extension packages
12 import javax.swing.*;
13
14 public class ShapesApplet extends JApplet {
15
16     // initialize the applet
17     public void init()
18     {
19         // obtain color parameters from XHTML file
20         try {
21             int red = Integer.parseInt( getParameter( "red" ) );
22             int green = Integer.parseInt( getParameter( "green" ) );
23             int blue = Integer.parseInt( getParameter( "blue" ) );
24
25             Color backgroundColor = new Color( red, green, blue );
26
27             setBackground( backgroundColor );
28         }
29
30         // if there is an exception while processing the color
31         // parameters, catch it and ignore it
32         catch ( Exception exception ) {
33             // do nothing
34         }
35     }
36 }
```

Uso do
<jsp:plugin>
(Parte 1).

Linha 14

Linha 21-27

Define a cor do fundo
da **JApplet** com base
nos parâmetros

```

36
37 public void paint( Graphics g )
38 {
39     // create arrays of x and y coordinates
40     int xPoints[] =
41         { 55, 67, 109, 73, 83, 55, 27, 37, 1, 43 };
42     int yPoints[] =
43         { 0, 36, 36, 54, 96, 72, 96, 54, 36, 36 };
44
45     // obtain reference to a Graphics2D object
46     Graphics2D g2d = ( Graphics2D ) g;
47
48     // create a star from a series of points
49     GeneralPath star = new GeneralPath();
50
51     // set the initial coordinate of the GeneralPath
52     star.moveTo( xPoints[ 0 ], yPoints[ 0 ] );
53
54     // create the star--this does not draw the star
55     for ( int k = 1; k < xPoints.length; k++ )
56         star.lineTo( xPoints[ k ], yPoints[ k ] );
57
58     // close the shape
59     star.closePath();
60
61     // translate the origin to (200, 200)
62     g2d.translate( 200, 200 );
63
64     // rotate around origin and draw stars in random colors
65     for ( int j = 1; j <= 20; j++ ) {
66         g2d.rotate( Math.PI / 10.0 );
67

```

(Parte 2).

Linhas 40-66

```

68         g2d.setColor(
69             new Color( ( int ) ( Math.random() * 256 ),
70                       ( int ) ( Math.random() * 256 ),
71                       ( int ) ( Math.random() * 256 ) ) );
72
73         g2d.fill( star ); // draw a filled star
74     }
75 }
76

```

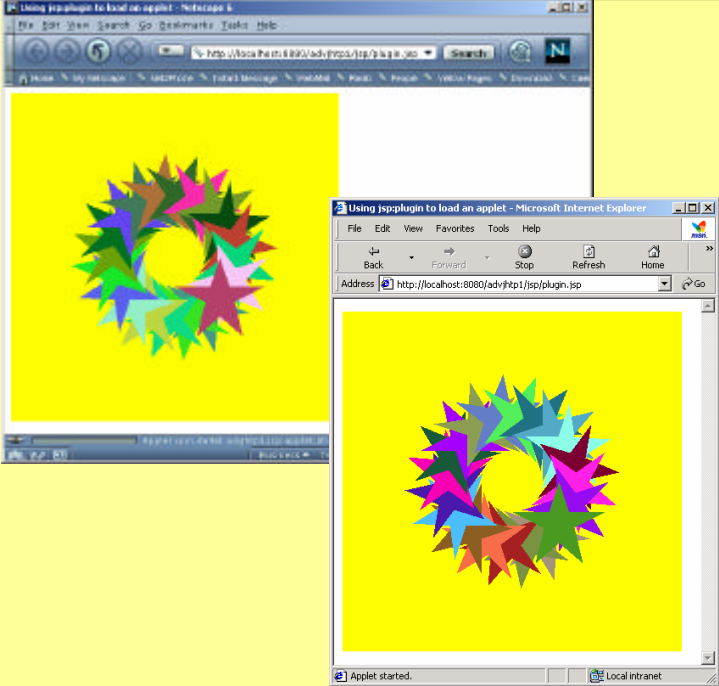
(Parte 3).

```
1 <!-- Fig. 10.15: plugin.jsp -->
2
3 <html>
4
5 <head>
6 <title>Using jsp:plugin to load an applet</title>
7 </head>
8
9 <body>
10 <jsp:plugin type = "applet"
11 code = "com.deitel.advjhtml.jsp.applet.ShapesApplet"
12 codebase = "/advjhtml/jsp"
13 width = "400"
14 height = "400">
15
16 <jsp:params>
17 <jsp:param name = "red" value = "255" />
18 <jsp:param name = "green" value = "255" />
19 <jsp:param name = "blue" value = "0" />
20 </jsp:params>
21
22 </jsp:plugin>
23 </body>
24 </html>
```

Uso de `<jsp:plugin>` para visualizar a JApplet na JSP

Linhas 16-20

Uso de `jsp:param` para definir a cor de fundo da JApplet



(Parte 2).

The image shows two overlapping screenshots of a web browser window. The top screenshot shows a yellow background with a colorful starburst applet in the center. The bottom screenshot shows the same applet, but with a status bar at the bottom that reads 'Applet started.' and 'Local intranet'.

Directivas

- Directivas JSP
 - Mensagens que são enviadas para o container do JSP
 - Permite ao programador definir:
 - Definir atributos da página
 - Incluir conteúdos de outros recursos
 - Definir bibliotecas de Marcas.
 - Limitadas por `<%@` and `%>`

Directivas (cont.)

| Directiva | Descrição |
|----------------|---|
| page | Define atributos associados à página tidos em conta pelo container do JSP |
| include | Inclui o conteúdo de um recurso na página actual. |
| taglib | Permite aos programadores incluir as suas próprias marcas. |
| | |

Exemplo

Uso da directiva Include

```
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. 10.28: includeDirective.jsp -->
6
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8
9    <head>
10     <title>Using the include directive</title>
11
12     <style type = "text/css">
13       body {
14         font-family: tahoma, helvetica, arial, sans-serif;
15       }
16
17       table, tr, td {
18         font-size: .9em;
19         border: 3px groove;
20         padding: 5px;
21         background-color: #dddddd;
22       }
23     </style>
24   </head>
25
26   <body>
27     <table>
28       <tr>
29         <td style = "width: 160px; text-align: center">
30           <img src = "images/logotiny.png"
31             width = "140" height = "93"
32             alt = "Deitel & Associates, Inc. Logo" />
33         </td>
34
```

JSP
includeDirective
.jsp mostra a cópia
de conteúdo em
tempo de tradução
da JSP para a
Servlet
(Parte 1).

Linha 5

```

35         <td>
36             <!-- include banner.html in this JSP -->
37             <%@ include file = "banner.html" %>
38
39         </td>
40     </tr>
41
42     <tr>
43         <td style = "width: 160px">
44             <!-- include toc.html in this JSP -->
45             <%@ include file = "toc.html" %>
46
47         </td>
48
49         <td style = "vertical-align: top">
50             <!-- include clock2.jsp in this JSP -->
51             <%@ include file = "clock2.jsp" %>
52
53         </td>
54     </tr>
55 </table>
56 </body>
57 </html>

```

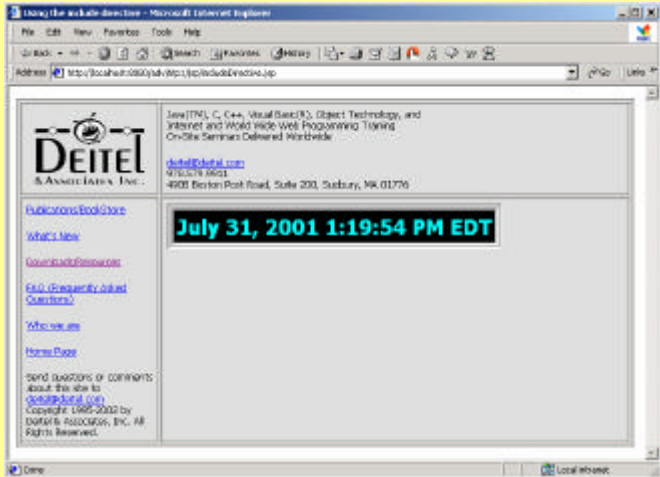
Uso da diretiva **include** para incluir **banner.html**

includeDirective.jsp mostra a cópia de conteúdo em tempo de tradução da JSP para a Servlet

Uso da diretiva **include** para incluir **toc.html**

Linha 47

Uso da diretiva **include** para incluir **clock2.jsp**



JSP **includeDirective.jsp** mostra a cópia de conteúdo em tempo de tradução da JSP para a Servlet (Parte 3).