

JavaServer Pages

Expressões e Declarações
Scriptlets
Diretivas



O que é JSP?

- JSP é uma tecnologia que permite a mistura de elementos estáticos HTML com conteúdo dinamicamente produzido por servlets.
- A parte produzida dinamicamente vem inserida em marcadores especiais dentro do código HTML.
- Exemplo:

Seu número de inscrição é

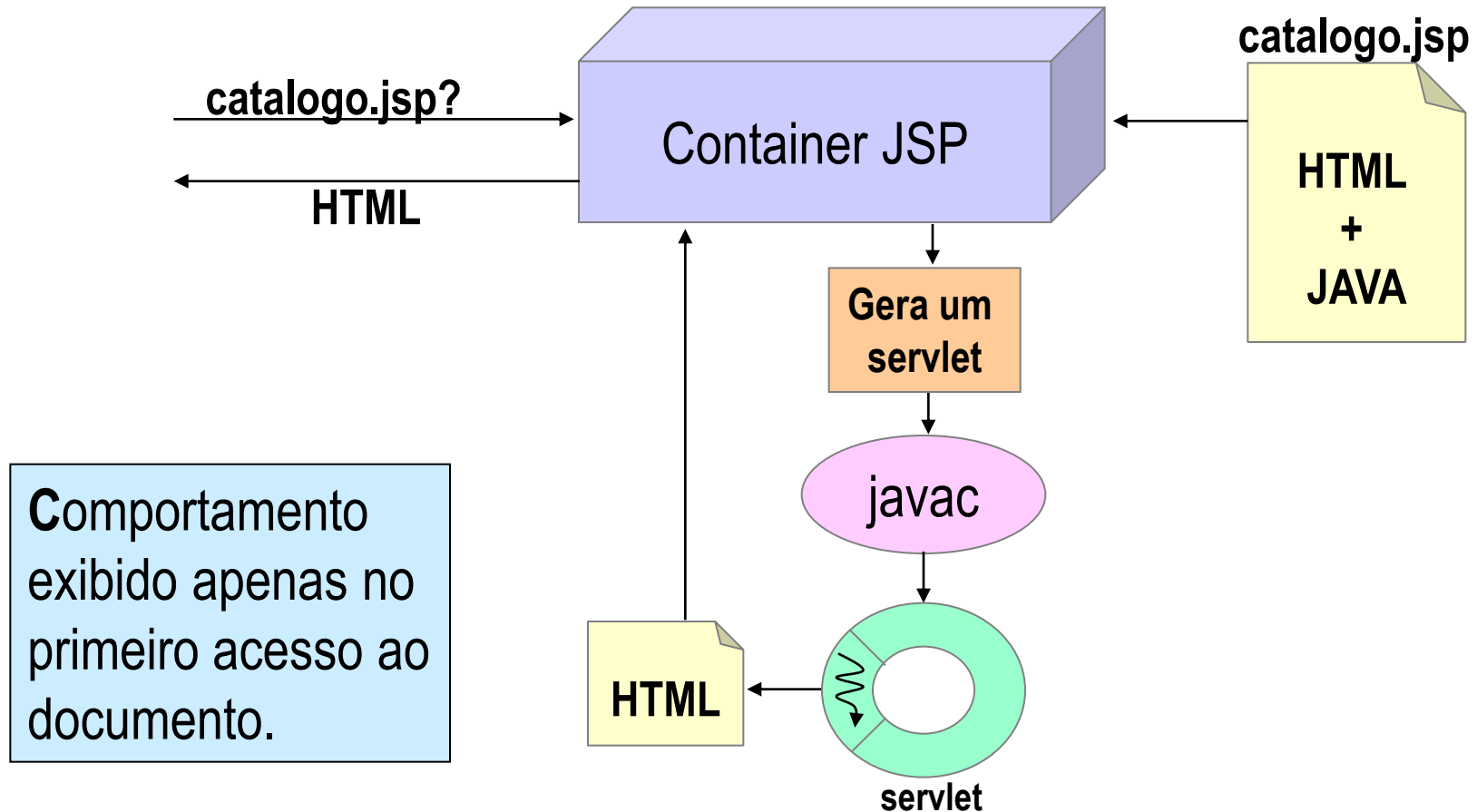
```
<b><%= request.getParameter("id") %></b>
```

JSP na aplicação web

■ Localização

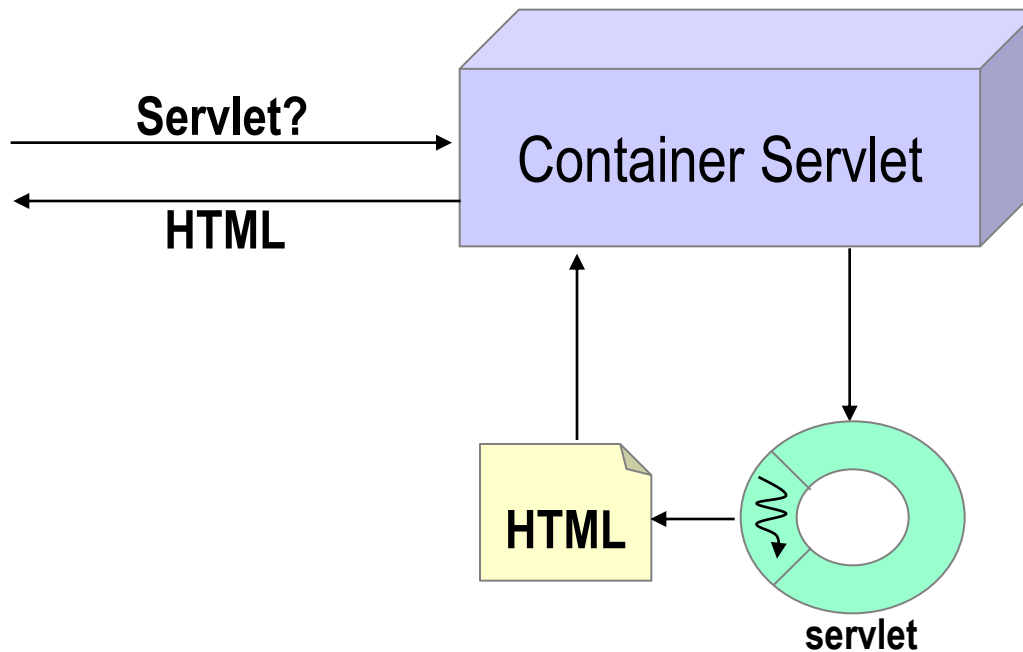


Funcionamento do JSP



Funcionamento do JSP

- Equivalente utilizando servlets puros:



Sintaxe JSP

■ Elementos JSP:

■ Elementos de *scripting*

- Permitem a inserção de código Java que tomará parte no servlet produzido pelo servidor JSP.

■ Diretivas

- Permitem o controle geral do servlet.

■ Ações

- Permitem especificar componentes JavaBeans a serem usados no servlet.

■ Linguagem de Expressões (EL)

- Simplificam o acesso a expressões e objetos implícitos

Sintaxe JSP

■ Elementos JSP:

■ Elementos de *scripting*

- `<%= expressões %>`
- `<%! declarações %>`
- `<% scriptlets %>`



Desaconselhados

■ Diretivas

- `<%@ page ... %>`
- `<%@ include ... %>`

■ Ações

- `<jsp:useBean ... />`

■ Linguagem de Expressões (EL)

- `${objeto.propriedade}`

Sintaxe JSP

■ Elementos JSP:

■ Elementos de *scripting*

- Permitem a inserção de código Java que tomará parte no servlet produzido pelo servidor JSP.

■ Diretivas

- Permitem o controle geral do servlet.

■ Ações

- Permitem especificar componentes a serem usados no servlet.

Elementos de *scripting*

■ Existem três tipos de elementos:

- Expressões: `<%= expressão %>`
 - São avaliadas e inseridas na saída produzida pelo servlet (como parâmetro dos métodos `println()`).
- *Scriptlets*: `<% código %>`
 - São inseridos no método `_jspService()` do servlet.
- Declarações: `<%! código %>`
 - São inseridas no corpo da classe (fora de qualquer método).
- ➡ Comentários JSP: `<%-- só permanece no .jsp --%>`

Elementos de *scripting* : Expressões

- As expressões são avaliadas, convertidas em String e inseridas na página.
- Alguns objetos pré-definidos estão disponíveis:
 - **request**: o objeto `HttpServletRequest` de `service()`
 - **response**: o objeto `HttpServletResponse` de `service()`
 - **session**: o objeto `HttpSession` associado a `request`.
 - **out**: o `PrintWriter` usado para gerar a resposta.

Elementos de *scripting* : Expressões

<HTML>

<HEAD><TITLE>Expressões JSP</TITLE></HEAD>

<BODY>

<H1>Algumas expressões JSP</H1>

 Hora atual: <%= new java.util.Date() %>

 Hostname: <%= request.getRemoteHost() %>

 ID de Sessão : <%= session.getId() %>

 O parâmetro <CODE>autor</CODE>:

<%= request.getParameter("autor") %>

</BODY>

</HTML>

Elementos de *scripting* : *Scriptlets*

```
<HTML>
```

```
<HEAD><TITLE>Pagina configurável</TITLE></HEAD>
```

```
<%
```

```
String bgColor = request.getParameter("cor");
```

```
boolean temCor;
```

```
if (bgColor == null)
```

```
    temCor = true;
```

```
else {
```

```
    temCor = false;
```

```
    bgColor = "WHITE";
```

```
}
```

```
%>
```

Uma expressão dentro do html!

```
<BODY BGCOLOR="<%= bgColor %>">
```

Elementos de *scripting* : *Scriptlets*

```
<H2 ALIGN="CENTER">Testando a cor</H2>
```

```
<%
```

```
if (temCor) {  
    out.println("Você forneceu a cor " +bgColor+".");  
} else {  
    out.println("Você não informou nenhuma cor." +  
                "A cor branca será usada." +  
                "Forneça a cor com o parâmetro cor.");  
}
```

```
%>
```

```
</BODY>
```

```
</HTML>
```

Elementos de *scripting* : *Scriptlets*

- *Scriptlets* podem ser usados para inclusão condicional de código HTML ou JSP:

```
<% if (nota > 7.0f) { %>
```

```
Você foi <B>aprovado</B>!
```

```
<% } else { %>
```

```
Você foi <B> reprovado</B>!
```

```
<% } %>
```

- Quando traduzido para o servlet fica:

```
if (nota > 7.0f) {  
    out.println("Você foi <B>aprovado</B>!");  
} else {  
    out.println("Você foi <B>reprovado</B>!");  
}
```

Elementos de *scripting* : Declarações

- Com este elemento é possível definir variáveis e métodos para a classe servlet a ser gerada.
- Exemplo:

```
<HTML>
<HEAD><TITLE>Teste de declarações</TITLE></HEAD>
<BODY>
<H1>Declarações JSP</H1>
<%! int contador=0; %>
<H2>Requisições atendidas desde o último reboot:
<%= ++contador %></H2>
</BODY></HTML>
```

Servlet gerado

```
class _ServletGerado extends HttpJspBase {  
  
    int contador=0;   
  
    public void _jspService  
        (HttpServletRequest req,  
         HttpServletResponse resp)  
        throws IOException, ServletException {  
        out.print("<HEAD>");  
        out.print("<TITLE>Teste de  
                   declarações</TITLE></HEAD>");  
        out.print("<BODY>");  
        out.print("<H2>Requisições atendidas desde o ...");  
        out.print(++contador);  
        out.print("</H2></BODY></HTML>");  
    }  
}
```


Variáveis JSP

Variável	Objeto
request	O objeto request do método service().
response	O objeto response do método service().
out	O PrintWriter usado para gerar a resposta.
session	O objeto HttpSession vinculado ao request.
application	O objeto ServletContext.
config	O objeto ServletConfig (usado com javabeans).
pageContext	Guarda o objeto PageContext (útil p/ TagLibs)
page	Sinônimo para this. (não recomendada)

Sintaxe JSP

■ Elementos JSP:

■ Elementos de *scripting*

- Permitem a inserção de código Java que tomará parte no servlet produzido pelo servidor JSP.

■ Diretivas

- Permitem o controle de características do servlet gerado.

■ Ações

- Permitem especificar componentes a serem usados no servlet.

Diretivas

- São usadas para definir a estrutura do servlet gerado a partir da página JSP.
- São 3 as diretivas JSP:
 - **page**: importação de classes, content-type, etc.
 - **include**: inclusão de arquivos no servlet.
 - **taglib**: uso de *tags* JSP.
- Sintaxe geral de uma diretiva:

```
<%@ diretiva atributo1="valor1"  
atributo2="valor2" atributoN="valorN" %>
```

Diretiva page

- Exemplo: `<%@ page contentType="text/plain"%>`
- Possui os seguintes atributos:
 - `import`
 - `java.lang`, `javax.servlet`, `javax.servlet.jsp` e `javax.servlet.http` são importados automaticamente.
 - `contentType`
 - Define o valor do campo Content-type do protocolo HTTP.
 - `isThreadSafe` (true ou false)
 - Se false, o servlet gerado implementará `SingleThreadModel`
 - `session` (true ou false)
 - Define se o objeto session será ou não instanciado.

Diretiva page

■ Possui os seguintes atributos:

■ buffer

- Define o tamanho do buffer associado ao objeto out.

■ autoFlush (true ou false)

- Despacha ou não, automaticamente, o conteúdo do buffer quando o mesmo estiver cheio.

■ extends

- Indica a superclasse do servlet a ser gerado.

■ info

- String a ser recuperado com o método `getServletInfo()`.

Diretiva page

■ Possui os seguintes atributos:

- **errorPage** (URL da página de erro)
 - Especifica uma página JSP responsável por tratar as exceções lançadas pela página atual. Usada em conjunto com isErrorPage.
- **isErrorPage** (true ou false)
 - Indica se esta é uma página de tratamento de exceções de uma outra página JSP. Variável exception disponível. Usada em conjunto com errorPage
- **language**
 - Inútil: indica a linguagem utilizada para programação. Por enquanto não tem utilidade.

Diretiva page

■ Exemplos:

```
<%@ page import="java.util.*" %>
```

```
<%@ page import="java.sql.*, java.io.*" %>
```

```
<%@ page contentType="text/plain" %>
```

```
<%@ page isThreadSafe="false" %>
```

(Default é true)

```
<%@ page extends="LojaServlet" %>
```

```
<%@ page session="false" %>
```

(Default é true)

Diretiva page

- Onde utilizar a diretiva page:
 - Em qualquer lugar da página JSP, o mais comum é colocá-la no início do documento, após o <BODY>
- Apenas o atributo **import** pode aparecer em mais de uma diretiva **page** na mesma página.

Diretiva include

- Utilizada para incluir arquivos em tempo de **geração do servlet**.
- O arquivo incluído pode ter elementos JSP e HTML.
- Sintaxe da diretiva:

```
<%-- Arquivo modificado em 12/2/2001 --%>
```

```
<%@ include file="URL do arquivo" %>
```
- Aplicações:
 - Barras de navegação reutilizáveis
 - Cabeçalhos e rodapés comuns a várias páginas do site.
 - Contadores, etc.

Diretiva include

■ Exemplo:

```
<HTML>
<HEAD><TITLE>Diretiva include</TITLE></HEAD>
<BODY>
<H1>Curso de Servlets&JSP</H1>
<P> Era uma vez uma linguagem chamada Java...
<P> Blá, blá, blá
<P>
<%@ include file="Rodape.jsp" %>
</BODY>
</HTML>
```

Diretiva include

■ Arquivo Rodape.jsp:

```
<%@ page import="java.util.Date" %>
```

```
<P><HR>
```

```
Curso de Servlets&JSP. Fred &copy; 2001.<BR>
```

```
Contatos: <A HREF="mailto:fred@gi.cefetpb.br">  
fred@gi.cefetpb.br</A>.
```

```
Data: <%= new Date() %>
```

Ação `jsp:include`

- Utilizada para incluir arquivos em tempo de **atendimento a requisições**
- O arquivo incluído não pode conter diretivas JSP, mas pode ter sido gerado por um JSP ou por um servlet
- Sintaxe:

```
<jsp:include page="URL" flush="true" />
```

- O atributo `flush` deve ser sempre `true`!