

[< VOLTAR](#)

Diretivas JSP e objetos implícitos

Capacitar o aluno para um melhor entendimento sobre alguns conceitos importantíssimos ao desenvolvimento de sites dinâmicos, por meio da utilização de diretivas e de objetos implícitos do JSP.

NESTE TÓPICO

- › Introdução
- › Diretivas
- › Objetos implícitos
- › Referências



Introdução

Conforme definido na aula 9, as diretivas são TAGs específicas que ajudam na configuração de páginas JSP (SANTOS, 2007, p.82). São elementos dinâmicos do JSP, assim como os scriptlets e JavaBeans, entre outros.

Já os objetos implícitos são objetos que podem ser acessados no JSP sem que haja necessidade de declará-los ou importá-los. Veremos suas principais utilizações nesta aula.

Diretivas

As diretivas estão divididas em três: - **page**, que configura as propriedades de uma página; - **include**, que insere o conteúdo de outro arquivo em uma página JSP e - **taglib**, que pode estender um conjunto de TAGs que o contêiner pode interpretar. (SANTOS, 2007, p.82).

Sintaxe geral para utilização de diretivas:

1. `<%@ diretiva nome_atrib_diretiva="valor" %>`

- **Diretiva @page**

Serve para configurar as propriedades da página.

- **import**

1. `<%@ page import="java.util.Vector" %>`
2. `<%@ page import="java.util.*" %>`
3. `<%@ page import="java.util.Vector, ava.util.ArrayList" %>`

Similar ao import de Java.

- **session**

1. `<%@ page session="true" %>` - Default
2. `<%@ page session="false" %>`

Controla quando um JSP participa ou não de sessões. O valor false não é similar a getSession (false).

- **buffer**

1. `<%@ page buffer="sizekb" %>`
2. `<%@ page buffer="none" %>`



Especifica o tamanho do buffer usado pelo PrintWriter do JSP.

- **autoflush**

1. `<%@ page autoflush="true" %>` - Default
2. `<%@ page autoflush="false" %>`

Controla se o buffer de saída será "esvaziado" automaticamente caso esteja cheio.

- **contentType**

1. `<%@ page contentType="text/html"%>`

Seta o content-type response header indicando o mime da resposta.

- **language**

1. `<%@ page language="Java"%>` - única suportada

Indica a linguagem que está sendo usada.

- **errorPage**

1. `<%@ page errorPage="URL Relativa"%>`

Indica que página deve ser processada caso uma exceção seja lançada. Tal página deve ter a opção `isErrorPage` setada para `true`.

- **isErrorPage**

1. `<%@ page isErrorPage="true"%>` -
2. `<%@ page isErrorPage="false"%>` - Default

Habilita uma página para tratar exceções por meio do objeto implícito `exception` (visto mais adiante).

- **info**

1. `<%@ page info="mensagem"%>`

Similar a `getServletInfo()`.



- **extend**

1. `<%@ page extends="package.class"%>`

Ajusta que classe o servlet gerado deve estender.

- **isThreadSafe**

1. `<%@ page isThreadSafe="true"%>` - Default
2. `<%@ page isThreadSafe="false"%>`

Ajusta quando o servlet gerado implementará `SingleThreadModel`

- **Diretiva @include**

Serve para inserir o conteúdo de outro arquivo em uma página JSP, fazendo com que seja compilado como se fosse um arquivo e permite a inclusão de textos em tempo de tradução. É possível usar a diretiva para incluir um cabeçalho padrão para todas as páginas, por exemplo, `<%@ include file="copyright.html" %>`

- **Diretiva @tagLib**

Serve para estender um conjunto de TAGs que o contêiner pode interpretar.

Objetos implícitos

Os objetos implícitos podem ser acessados no JSP sem que haja necessidade de declará-los ou importá-los.

Objetos:

- **Out:** usado para enviar respostas para o cliente.

```
1. <%  
2.     out.println("Exemplo");  
3. %>
```

- **request:** referencia a requisição corrente que causou a chamada ao serviço.

```
1. <%  
2.     String numero = request.getParameter("numero");  
3.     out.println("vc digitou o numero " + numero);  
4. %>
```

- **response:** usado para mandar uma resposta ao cliente. Permite enviar mensagens de erro, alterar códigos de status HTTP e headers em uma página JSP.



```
1. <%  
2.     response.setHeader("Pragma", "no-cache");  
3.     if (request.getProtocol().equals("HTTP/1.1")) {  
4.         response.setHeader("Cache-Control", "no-cache");  
5.     }  
6. %>
```

- **session:** referencia a sessão do usuário. Ausente quando session="false" for definido no início da página.

```
1. <%  
2.     Usuario user = (Usuario) session.getAttribute("user");  
3.     if (user == null){  
4.         response.sendRedirect("login.jsp");  
5.     }  
6. %>
```

- **exception:** usado para situações de erro. Possui escopo de página e só está acessível se antes for definido isErrorPage="true".

```
1. <%@ page isErrorPage="true" %>
2. <I><%= exception.getMessage() %></I>.
3. <PRE>
4. <%
5.     exception.printStackTrace(new java.io.PrintWriter(out));
6. %>
7. </PRE>
```

application: referencia o contexto da aplicação Web.

```
1. <%     out.println(application.getRealPath("/tmp")); %>
```

- **page:** Faz referência a própria página. Equivalente ao "this".
- **config:** Permite acesso a informações de configuração do JSP. É do tipo `javax.servlet.ServletConfig`.
- **pageContext:** refere-se ao ambiente da classe. Guarda referências aos objetos implícitos e dispõe de métodos de leitura e escrita para atributos em diferentes escopos.

Para conhecer um pouco mais sobre essas atividades, veja o slideshow abaixo. Este slideshow faz parte da sequência desta aula e, portanto, é essencial para a aprendizagem.



Exercício

- Elaborar um programa que faça o cálculo do IMC.

- Esse exemplo já foi colocado na aula 8, no entanto, lá o objetivo era simplesmente passar dados de um formulário ao servidor, para que algumas regras fossem resolvidas por meio do JSP.

- Aqui vamos criar uma classe (um JavaBean, discutido na aula 10) e fazer com que o JavaBean ofereça o meio necessário para transporte dos dados e suporte às regras para o cálculo do IMC e apresentação de seus resultados.

Interação entre Cliente e Servidor.



Referências

SANTOS, Rui Rossi dos. *Java na web*: programando sites dinâmicos. Rio de Janeiro: Axcel Books, 2007.



Avalie este tópico



ANTERIOR

Vetores e coleções de objetos

Biblioteca

(<https://www.uninove.br/conheca->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Implementação de métodos com declaração

Ajuda?

(<https://ava.uninove.br/curso/>)

em

© Todos os direitos reservados

