Diretrizes do JSP

Introdução

No Capítulo 1, vimos que existem dois tipos de tags: *Scripting-oriented* e *XML-based*. Neste capítulo estudaremos melhor os dois tipos de tags e veremos como a linguagem *Scripting* lhe permite acessar e modificar propriedades dos seus servlets (ou beans).

Diretrizes do JSP

A linguagem *Script* padrão em uma página JSP é naturalmente Java. Os servlets também são escritos em Java puro. Entramos novamente no assunto sobre a diferença entre JSP e servlets. A diferença é que o JSP, apesar de ser compilado em Java puro (exatamente como um servlet), aceita linguagens *Scripts* e tags HTML em seu conteúdo. Os servlets não aceitam linguagens *Scripting* nem tags HTML. O único jeito de se manipular tags *Scripting* via servlets é utilizando os objetos *response* e *request*. Mas essa é uma outra história. O fato é que os JSP's são compilados *on-th-fly*, ou seja – antes de mais nada – são interpretados e então compilados em servlets.

Por ora sabemos então que JavaServerPages (JSP) são páginas que se comportam exatamente como servlets. E que servlets são classes do Java utilizadas para criar componentes para o servidor. A diferença é que você pode criar um componente (objeto) que cuide do acesso aos dados e criar então uma página JSP (interface) que apresente um formulário e acione o componente para cadastrar informações no banco de dados.

Concluindo, o interpretador do Java compila tanto o servlet como a página JSP em servlets. Dessa forma, o JSP transforma-se em uma poderosa ferramenta! Uma vez que a própria página JSP é um servlet, então podemos criar formulários para acessar e modificar até os métodos (públicos) de objetos ordinários, como Beans, ou a página em si! Além de manipular exceções causadas por erros.

Enfim, para um JSP se comportar como um servlet, algumas tags são diretivas, ou seja, servem para uma única finalidade: orientar o interpretador a compilar o JSP em servlet. Temos então as diretrizes do JSP que definem as regras para organizar a linguagem Scripting do JSP nos padrões de compilação dos servlets:

- *Expressões* `<%= expressão %>`
  Esta tag permite imprimir diretamente no HTML ou passar expressões para outras funções ou tags.
- *Scriptlets* `<% código %>`
  Esta tag permite que você utilize linguagens Scripts para interagir com a página. O que estiver dentro destas tags será executado pelo _JspService.
- *Declarações* `<%! declaração %>`
  Tag que permite declarações de variáveis de instância, fora de quaisquer funções ou métodos existentes (portanto fora do _JspService).
- *Diretivas* `<%@ diretiva atributo1="valor1" atributoN="valorN" %>`
  Tags que processam informações especiais que devem ser explicitamente declaradas. Estas tags podem mudar configurações de diversos objetos, carregar classes, importar arquivos etc.

Expressões

Tags de expressões <%= valor %> são utilizadas para escrever conteúdo dinâmico direto no código, antes mesmo de ser compilado. Ou seja, neste tipo de construção, não se usa ponto-e-vírgula (;). Isso se dá pois quem vai colocar o ponto-e-vírgula é o interpretador. Então a seguinte construção:

```
<%= "Renato!" %>
```

Na verdade é interpretada e passada para o compilador da seguinte forma:

```
out.println("Renato!");
```

E no HTML o resultado final seria:

```
Renato!
```

Resumindo, esta diretiva permite escrever os resultados das expressões contidas entre as tags <%= e %>.

Scriptlets

Estas tags delimitam um código genérico que pode conter um Script qualquer, inclusive declarações (privadas). Scriptlets não podem entretanto imprimir expressões diretamente na página JSP. A sintaxe usual da tag Scriptlet é:

```
<% código %>
```

Scriptlets também podem ser escritas através de tags *XML-based*:

```
<jsp:scriptlet> scriptlet </jsp:scriptlet>
```

As tags Scriptlets também podem quebrar o código Java para se mesclarem com o HTML, por exemplo:

```
<% if (variavel == null) { %>
<p>Erro!
<% } else { %>
<p>OK!<br>
Variável: <%=variavel%>
<% } %>
```

Note que os comandos normais estão dentro das tags Scriptlets, mas para imprimir o valor da variável no HTML é usada uma tag de expressões.

Declarações

As tags de declarações são usadas para definir variáveis e métodos específicos para uma página JSP. A sintaxe para se declarar uma variável é:

```
<%! variavel = valor; %>
```

Que também pode ser escrita em tags *XML-based*:

```
<jsp:declaration> variavel = valor; </jsp:declaration>
```

Ao se criar uma declaração, tenha em mente que outros *requests* estarão criando instâncias da sua página JSP. Ou seja, as declarações são criadas em todas as instâncias abertas. Ao modificar o valor de uma variável de instância, você estará modificando o valor de todas as outras instâncias. Para criar variáveis independentes para cada instância, utilize a tag Scriptlet:

```
<% variavel = valor; %>
```

Existem também as variáveis estáticas, usadas para declarar variáveis de classes, que são compartilhadas entre todas as instâncias das classes.

```
<%! static public int counter=0; %>
```

Variáveis Pré-definidas

As variáveis pré-definidas são objetos implícitos, que existem enquanto a página JSP existir. Estas variáveis não podem ser usadas em declarações, apenas em Scriptlets e expressões. Algumas das principais variáveis pré-definidas:

- *request*
  Esta é a variável que a classe HttpServletResponse cria contendo os parâmetros passados pelo http. Você pode ter acesso também ao tipo do request (GET ou POST), além do cabeçalho da página (cookies etc.).
- *response*
  Esta variável é a responsável pela resposta ao browser do cliente, criada pela classe HttpServletResponse.
- *out*
  Variável gerada pela classe PrintWriter utilizada para enviar dados para o cliente, utilizada em tags Scriptlets, em substituição das tags de expressões.
- *session*
  Esta variável é criada pela classe HttpSession invocada pelo request. Este tipo de variável pode ficar guardada enquanto existir uma sessão de usuário.
- *application*
  Esta variável é criada em ServletContext, obtido via getServletConfig().getContext(). Diferentemente das variáveis session, as variáveis de aplicação são compartilhadas por todos os clientes.