Diretivas do JSP

As diretivas do JSP são tags que afetam a estrutura dos servlets na construção da página JSP. Um exemplo:

```
<%@ diretiva atributo="valor" %>
```

No JSP existem três tipos de diretivas: `page`, `include` e `taglib`. A diretiva `page` lhe permite controlar a estrutura do servlet, tornado-o capaz de importar classes, personalizar a superclass do servlet, configurar o `contentType` entre outras coisas. Esta diretiva pode ser colocada em qualquer ponto da página. A diretiva `include` lhe permite incluir um arquivo na página JSP em tempo de interpretação. Esta diretiva deve ser colocada no ponto de inserção do código. A diretiva `taglib` foi implementada no JSP 1.1. É usada para definir custom tags.

A diretiva `page` é a mais utilizada, portanto será o tópico principal desta parte do Capítulo 2 deste tutorial. A diretiva `page` lhe permite configurar um ou mais dos seguintes atributos: `import, contentType, isThreadSafe, session, buffer, autoflush, extends, info, errorPage, isErrorPage and language`.

Atributo Import

Este atributo da diretiva `page` lhe permite especificar os pacotes (packages) a serem utilizados pelo servlet criado da compilação da página JSP. Se você não especificar nenhuma classe a importar, como padrão, serão chamadas as seguintes classes: `java.lang.*, javax.servlet.*, javax.servlet.jsp.*, javax.servlet.http.*`.

Você pode utilizar o atributo import de três formas:

1. Assumindo que você está declarando uma classe `xyz` que não está dentro de um pacote, mas sim no diretório raiz dos servlets (no caso do JWS2.0 é o diretório "`c:\jws2.0\servlets`" – onde `jws2.0` é o diretório onde você instalou o JavaWebServer 2.0):
   ```
   <%@ page import="xyz" %>
   ```
2. Assumindo que você está declarando a classe `xyz` do pacote `myclasses` (que no caso do JWS 2.0 ficaria no diretório "`c:\jws2.0\servlets\myclasses`"):
   ```
   <%@ page import="myclasses.xyz" %>
   ```
3. Assumindo que você precisa acessar várias classes do pacote `myclasses`:
   ```
   <%@ page import="myclasses.*" %>
   ```

Os atributos de importação podem aparecer em qualquer parte do documento, mas como idependente de onde estão posicionados, afetam a página como um todo, normalmente são colocados no início da página.

Atributo contentType

O atributo `contentType` configura o tipo de conteúdo que será enviado junto com o cabeçalho. Ou seja, este atributo configura o tipo MIME da página que está sendo enviada para o cliente. Por exemplo:

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

Tem o mesmo efeito de:

```
<% response.setContentType("text/plain"); %>
```

O tipo de conteúdo padrão para as páginas JSP é `text/html`. O tipo de conteúdo `text/plain` é texto puro.

Atributo isThreadSafe

O atributo `isThreadSafe` controla se o servlet gerado pela página JSP implementa ou não a interface `SingleThreadModel`. Este atributo pode ter apenas dois estados:

```
<%@ page isThreadSafe="true" %> (Padrão)
<%@ page isThreadSafe="false" %>
```

Os servlets normalmente são acessados simultaneamente por diversos requests dos usuários. Resultando em uma série de instâncias de acessos aos métodos (service) da mesma instância do servlet. Essa condição admite que o servlet é *Thread Safe*, ou seja, o servlet garante que a sincronização de dados em seus campos não retornarão valores inconsistentes.

Em alguns casos, você não precisa se preocupar se dois usuários ocasionalmente pegarem o mesmo valor de uma variável, como no caso de contadores de acessos. Mas no caso de ID's incrementados automaticamente, a história muda de figura, pois se dois usuários pegarem o mesmo ID o sistema pode travar. Então é necessário que você controle a sincronização das instâncias. Por exemplo, o código abaixo não é *Thread Safe* pois dois usuários podem pegar o mesmo número de ID antes que o `idNum` seja incrementado.

```
<%! Private int idNum = 0; %>
<%
      String userID = "userID" + idNum;
      out.println("Seu ID é: " + userID + ".");
      idNum = idNum + 1;
%>
```

Este código acima deveria usar um bloco de sincronização com a seguinte sintaxe:

```
synchronized(someObject) {...}
```

Este bloco de sincronização garante que um bloco não será executado antes que outro termine, formando uma fila se necessário.

Então nós temos o código de ID's revisado:

```
<%! Private int idNum = 0; %>
<%
synchronized(this) {
      String userID = "userID" + idNum;
      out.println("Seu ID é: " + userID + ".");
      idNum = idNum + 1;
}
%>
```

Se você configurar o `isThreadSafe` para o estado `false`, ou seja desligado, o servlet implementará a interface `SingleThreadModel`, garantindo acessos sequenciais.

Atributo session

O atributo `session` controla se a página utiliza ou não sessões de usuários. O uso deste atributo prevê uma das duas formas:

`<%@ page session="true" %>` (Padrão)
`<%@ page session="false" %>`

O valor `true` indica que a variável pré-definida `session` foi configurada para a sessão de usuário existente, se existir uma; caso contrário, uma nova sessão é criada e a variável `session` é configurada como `false`.

Atributo buffer

Este atributo configura o tamanho que o `buffer` precisa atingir para começar a enviar a página em si. O valor padrão para este atributo é 8kb, mas outros valores podem ser configurados da seguinte forma:

```
<%@ page buffer="32kb" %>
```

Esta tag acima configura o `buffer` para começar a enviar dados para o cliente quando atingir 32kb ou a página estiver completa.

Atributo extends

O atributo extends indica a superclass do servlet gerado pela página JSP:

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

Atributo info

O atributo info guarda um valor que pode ser obtido através do método `getServletInfo`. Use o atributo info da seguinte forma:

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

Atributo errorPage

O atributo errorPage especifica uma página de erro a ser processada se alguma exceção ocorrer:

```
<%@ page errorPage="url" %>
```

Atributo isErrorPage

O atributo `isErrorPage` indica se a página atual é uma página de erro chamada pelo atributo `errorPage` em outra página JSP. Desse modo é possível se tratar as exceções:

Pagina1.jsp

```
...
<%@ page errorPage="Pagina1erro.jsp" %>
...
...
```

Pagina1erro.jsp

```
<%@ isErrorPage="true" %>
...
<p><%=exception%>
<p><%exception.printStackTrace(new PrintWriter(out));%>
...
```

Atributo language

Este atributo foi concebido para implementações de novas linguagens Scripts. No momento não se preocupe com este atributo pois o padrão é:

```
<%@ page language="java" %>
```

Sintaxe XML-based das Diretivas

Você também pode utilizar tags XML para declarar as diretivas, por exemplo:

```
<jsp:directive.page import="java.util.*" />
```