Geração de Imagem com Java/JSP

Exemplo: Código de Barras

por: Ricardo Gil Guerreiro Scavasin
ricardogil@ig.com.br

e: Ulisses Telemaco Neto
ulisses@jspbrasil.com.br

Nesse artigo iremos discutir como criar imagens "On Fly", ou seja, aquelas criadas e enviadas para o cliente em tempo de execução, sem ocupar espaço de armazenamento do lado do servidor. Escolhemos como exemplo uma funcionalidade muito usada em várias aplicações Web: a geração dinâmica de código de barras. Lembramos que o intuito desse texto não é esgotar uma discussão sobre o processo de geração de códigos de barras. Apenas usamos esse exemplo para demonstrar a capacidade da tecnologia Java em gerar imagens em tempo de execução.

Teste o Exemplo

Estratégia Utilizada

A idéia básica usada no processo de geração de imagens "On Fly" descrito nesse texto é a seguinte: **Uma classe Java é responsável por gerar uma imagem e enviá-la para a saída padrão de uma página JSP.**

Códigos Fontes

Três códigos fontes foram usados para implementar o exemplo de geração de código de barras desse texto: dois códigos escritos em JSP e uma classe Java.

Abaixo segue o código que implementa a página JSP que exibe a imagem:

```
<%
String codBarras = request.getParameter("codBarras");
if(codBarras==null)
    codBarras = "1234567890";
%>

<html>
<head><title>Artigo JSPBrasil - Geração de Imagem "On Fly" em
JSP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
</head>
<body bgcolor="#FFFFFF">
<form>
Código de Barras:
<input type="text" name="codBarras">
<input type="submit" value="Gerar Código"><br>
* Coloque apenas números;<br>
* Todo código de barras é formado por uma quantidade par de
números;
</form>
<center>
<img src="exibe_imagem.jsp?p_barra=<%= codBarras %>"><br>
<%= codBarras %>
</center>
</body>
</html>
```

Esse é o mais simples dos códigos utilizados aqui e não contém grandes novidades. Uma tag HTML <img> tem como valor do atributo "src" uma página JSP. Isso indica que quando esse código for chamado a página JSP será executada.

O trecho é o código JSP que representa a imagem:

```
<%@ page import="artigos.Boleto" %>
<jsp:useBean id="img" scope="page" class="artigos.Boleto" />
<%
response.setContentType("image/jpeg");
response.setHeader ("Pragma", "no-cache");
response.setHeader ("Cache-Control", "no-cache");
response.setDateHeader ("Expires",0);
%>
<%! boolean retorno;%>
<%! String barras;%>
<%
barras = request.getParameter("p_barra");
retorno = img.criaImagem(barras,pageContext,25);
%>
```

```
int h)
                                    throws
ServletException,IOException {
int i, j, tam;
//Variavel que contem a representacao em Codigo de Barras de cada
um dos numeros.
//Nessa representacao:
// 0 significa Barra Fina (NARROW);
// 1 significa Barra Grossa (WIDE).
String[] barras = {"00110", //0
                   "10001", //1
                   "01001", //2
                   "11000", //3
                   "00101", //4
                   "10100", //5
                   "01100", //6
                   "00011", //7
                   "10010", //8
                   "01010"};//9
//O Codigo de barras e formado sempre por pares intercalados.
//Por exemplo 12:
//Pegando-se a representacao do 1 e do 2 na variavel acima
teriamos o seguinte:
//1001000011.
//Com isso, o primeiro numero representa as Barras Pretas (Fina ou
Grossa)
// e o Segundo numero representa as Barras Brancas, ou espacos,
(Fina ou Grossa).
int preto,branco;
int w = 9 + (9 * texto.length());
HttpServletResponse response;
response = (HttpServletResponse)pageContext.getResponse();
ServletOutputStream out = response.getOutputStream();
image = new java.awt.image.BufferedImage(w,h,
java.awt.image.BufferedImage.TYPE_INT_RGB);
tam = 0;
fillRect(0, 0, w, h, 0x00FFFFFF); //começo do Codigo de Barras = 0
= 00 / 1 = 00
fillRect(tam, 0, tam+1, h, 0x00000000);
tam++;
fillRect(tam, 0, tam+1, h, 0x00FFFFFF);
tam++;
fillRect(tam, 0, tam+1, h, 0);
tam++;
fillRect(tam, 0, tam+1, h, 0x00FFFFFF);
tam++;
//Conteudo do Codigo de Barras
for(i = 0; i <= texto.length()-1; i++) {
    preto = Integer.parseInt(String.valueOf(texto.charAt(i)));
    branco = Integer.parseInt(String.valueOf(texto.charAt(i+1)));
    i++;
    for(j = 0; j < 5; j++) {
        if(String.valueOf(barras[preto].charAt(j)).equals("0")) {
            fillRect(tam, 0, tam + 1, h, 0);
            tam++;
        }
```

```
        else { fillRect(tam, 0, tam + 3, h, 0);
            tam+=3;
        }
        if(String.valueOf(barras[branco].charAt(j)).equals("0")) {
            fillRect(tam, 0, tam + 1, h, 0x00FFFFFF); tam++;
        }
        else {
            fillRect(tam, 0, tam + 3, h, 0x00FFFFFF); tam += 3;
        }
    }
}

//fim da barra = 0 = 10 / 1 = 0
fillRect(tam, 0, tam+3, h, 0);
tam+=3;
fillRect(tam, 0, tam+1, h, 0x00FFFFFF);
tam++;
fillRect(tam, 0, tam+1, h, 0);
tam++;

JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);
encoder.encode(image);
//GifEncoder encoder = new GifEncoder(BarImage ,outb);
//encoder.encode();
// ESSAS DUAS LINHAS ACIMA CRIARIAM UM .gif
out.close();
return true; }

void fillRect(int x0, int y0, int x1, int y1, int color)
{
    for (int x=x0 ; x < x1 ; x++){
      for (int y=y0 ; y < y1 ; y++){
         image.setRGB(x, y, color);
      }
    }
}
}
```

O Código acima implementa a classe "Boleto". Essa classe possui um método chamado "criaImagem()" que recebe três valores como parâmetro: um String que representa a sequência de números que será traduzida para código de barras, um objeto do tipo PageContext que representa a página para onde será devolvida a imagem criada e finalmente a altura que a imagem terá (a largura da imagem, é definida pelo tamanho da cadeia números que será codificada).

O restante do código, numa primeira análise, parece um pouco confuso, porém não é o objetivo desse texto detalhar o processo de geração de código de barras. Ainda resta um trecho importante a ser analisado: depois de gerada, a imagem é, enfim, enviada para o objeto de saída ("out") da página ("pageContext")

```
JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);
encoder.encode(image);
```

Resumo

O objetivo desse texto foi mostrar como gerar imagens através de Java e exibi-las através de páginas HTML/JSP. As imagens geradas são consideradas "On Fly", ou seja, não ficam armazenadas no servidor. Ao invés disso, são geradas e enviadas aos cliente em tempo execução. No exemplo usado aqui foi implementado a geração de código de barras.