



## Tutoriais

### Tratando Formulários

por: Ulisses Telemaco Neto  
[ulisses@jspbrasil.com.br](mailto:ulisses@jspbrasil.com.br)

Os formulários são ferramentas úteis e muito usada em diversas aplicações: cadastro registros em um banco de dados, validação de um login/senha, envio de email, envio de dados de um pesquisa, etc. Hoje é difícil desenvolver uma aplicação para Web que não exija o uso de formulários. Pois bem, na lição de hoje vamos aprender manipular formulários em aplicações JSP.

Apresentamos abaixo um código para mostrar o formato de um formulário HTML e de seus objetos.

```
<html>
<body>

<!-- cabeçalho do formulário -->
<form name="nomedoformulario" action="paginajsp.jsp" method="get">

<!-- caixa de texto -->
<input type="text" name="variavel1" size=40 maxlength=40>

<!-- caixa de texto para senha -->
<input type="password" name="variavel2" size=40 maxlength=40>

<!--objeto do tipo radio -->
<input type="radio" name="variavel2" value="valordavariavel">Texto da Varivavel
2

<!--objeto do tipo checkbox -->
<input type="checkbox" name="variavel3" value="xxxxx"> Texto da Varivavel 3

<!--objeto do tipo select -->
<select name="variavel4">
<option value="valor1">Valor 1
<option value="valor2">Valor 2
<option value="valor3">Valor 3
</select>

<!-- area de texto -->
<textarea name="variavel5" cols=40 rows=2>
Texto da Variavel 5
</textarea>

<!-- objeto hidden, para enviar dados que o usuário não vê no formulário -->
<input type="hidden" name="asd" value="asd">

<!-- botão -->
<input type="button" value="textodobotao">

<!-- botao de limpar informações do formulário -->
<input type="submit" value="limpar">

<!-- botao de enviar formulário -->
<input type="submit" value="ok">

<!-- imagem colocada para funcionar com botao de envio de formulário -->
<input type="image" src="pathdaimagem/image.gif">

<!-- objeto para anexar arquivo -->

<input type="file" name="asdas" accept="asd">

</form>

</body>
</html>
```

É importante fazermos algumas observações a cerca do código acima:

- no cabeçalho do formulário, indicamos através de action="pathdoarquivo/paginajsp.jsp" o arquivo JSP que receberá os seus dados.
- cada objeto do formulário recebe um nome. Deve-se tomar bastante cuidado ao nomear tais objetos, isto porque, como sabemos, as variáveis Java são sensíveis maiúscula/minúscula. Portanto, os objetos:

```
<input name="variavel1" type="text" value="">
<input name="Variavel1" type="text" value="">
```

São objetos diferentes porque receberam nomes diferentes (variavel1 e Variavel1).

Mostraremos mais um exemplo (bastante simples) de como enviar dados a partir de um formulário a uma página JSP.

```
<%-- Arquivo teste.jsp --%>
<html>
<body>

<center><h1> <%= request.getParameter("teste") %> </h1></center>

<form action="teste.jsp" method=get>
<input type="text" name="teste" size=40 maxlength=40><br>
<input type="submit" value="enviar">
</form>

</body>
</html>
```

A página jsp acima, chamada "teste.jsp", contém um formulário que envia para ela mesma. O valor digitado em uma caixa de texto será mostrado como título da página. Observe como fizemos isso:

- a página para qual nós enviaremos os dados do formulário é designada no cabeçalho do formulário:

```
<form action="teste.jsp" method=get>
```

- o nome do objeto caixa de texto caixa de texto ("teste") é usado na expressão `request.getParameter("teste")`. Note que se usássemos `request.getParameter("Teste")` (com T maiúsculo), a página não iria retornar o valor digitado na caixa de texto.

O próximo exemplo é formado por dois arquivos. O primeiro pode contém apenas códigos HTML e o segundo contém códigos HTML e JSP.

Arquivo "envia\_mês.htm":

```
<html>
<body>

<h3>Qual o mês do seu aniversário?</h3>
<form action="recebe_mes.jsp" method=get>
<select name="mesNasceu">
<option value="1">Janeiro
<option value="2">Fevereiro
<option value="3">Março
<option value="4">Abril
<option value="5">Maio
<option value="6">Junho
<option value="7">Julho
<option value="8">Setembro
<option value="9">Agosto
<option value="10">Outubro
<option value="11">Novembro
<option value="12">Dezembro
</select>

<input type="submit" value="enviar">
</form>

</body>
</html>
```

Arquivo "recebe\_mês.jsp":

```

<%@ page import=java.util.Date %>
<%@ page import=java.lang.String %>
<%
String msg = "";
String mesString = request.getParameter("mesNasceu");
int mes = Integer.parseInt(mesString);
Date dateNow = new Date();
int monthNow = dateNow.getMonth() + 1;

mes -= monthNow;

if (mes == 1)
msg = "Falta apenas "+ mes +" mês para o seu aniversário.";

if (mes == -1)
msg = "Seu aniversário foi no mês passado";

if (mes > 1)
msg = "Faltam "+ mes +" meses para o seu aniversário.";

if (mes == 0)
msg = "Oba... estamos no mês do seu aniversário.";

else if (mes < 1)
{
mes *= -1;
msg = "Seu aniversário foi a "+ mes +" meses atrás.";
}

%>

<html>
<body>
<center>
<h3><%= msg %></h3>
<br><br><br>
<a href="Javascript:history.back(-1)">voltar</a>
</center>

</body>
</html>

```

O exemplo acima é um pouco menos simples que o primeiro. O arquivo "envia\_mes-jsp" contém um formulário com um objeto select que envia o mês que a pessoa nasceu. Após escolher o mês e clicar no botão "ok", o formulário chama a página "recebe\_mes.jsp" e envia seus dados para ela. Esta segunda página é um pouco menos simples que a primeira. Vamos analisar o que ela faz:

- nas primeiras linhas utilizamos as tags "page import" para indicar quais classes iremos utilizar em nossa página:

```

<%@ page import=java.util.Date %>
<%@ page import=java.lang.String %>

```

- cinco objetos são criados e inicializados.

- Usamos o método "request.getParameter('nomedoparametro')" com a finalidade de pegar o valor passado para a página através de algum formulário ou passando diretamente em sua URL. O segundo objeto foi inicializado utilizando esse método para pegar o valor passado pelo formulário:

```
String mesString = request.getParameter("mesNasceu");
```

- O valor passado através de um formulário ou através da URL da página é sempre do tipo String. Ao inicializarmos o terceiro objeto, o método "Integer.parseInt(variavelString)" transformou o valor contido na variável mesString em Inteiro.

```
int mes = Integer.parseInt(mesString);
```

- O penúltimo objeto criado é do tipo Date (daí a importância de termos importado a classe java.util.Date na primeira linha de nossa página). Ele é inicializado com a hora local do servidor.

```
Date dateNow = new Date();
```

- Na inicialização do último objeto utilizamos o método "dateNow.getMonth()" que retorna um inteiro indicando o valor da variável. Somamos 1 ao valor retornado a partir desse método porque ele retorna 0 para janeiro, 1 para fevereiro e assim por diante.

```
int monthNow = dateNow.getMonth() + 1;
```

- Cinco teste são efetuados dentro de um script (<% e %>). Eles são usados para definir o valor que a variável "msg" terá, ou seja, a partir dos testes, decidiremos qual mensagem será exibida na tela.

- Após efetuar os testes, o texto HTML é inserido na página.

- Uma expressão (<%= %>) é usada para exibir o valor da variável "msg":

```
<%= msg %>
```