



## Tutoriais

### Tratando Cookies

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

Cookie é um mecanismo padrão fornecido pelo protocolo HTTP e que permite gravarmos pequenas quantidades de dados persistentes no navegador de um usuário. Tais dados podem ser recuperados posteriormente pelo navegador. Esse mecanismo é usado quando queremos recuperar informações de algum usuário. Com os cookies, pode-se reconhecer quem entra num site, de onde vem, com que periodicidade costuma voltar. Para se ter uma idéia de como eles fazem parte da sua vida, dê uma olhada na sua máquina. Se você usa o Internet Explore 5.1, vá a c:\windows\cookies. No Communicator 4.7, os cookies ficam em c:\arquivos de programas\netscape\users. Os cookies em si não atrapalham ninguém, se propriamente usados.

Como padrão, os cookies expiram tão logo o usuário encerra a navegação naquele site, porém podemos configurá-los para persistir por vários dias. Além dos dados que ele armazena, um cookie recebe um nome; um servidor pode então definir múltiplos cookies e fazer a identificação entre eles através dos seus nomes.

Os cookies são associados ao URL da página que os manipula.

### Gerenciando Cookies

Os cookies são definidos por um servidor da web. Quando um usuário solicita um URL cujo servidor e diretório correspondam àqueles de um ou mais de seus cookies armazenados, os cookies correspondentes são enviados de volta para o servidor. As páginas JSP acessam os seus cookies associados através do método `getCookies()` do objeto implícito `request`. De forma similar, as páginas JSP podem criar ou alterar cookies através do método `addCookie()` do objeto implícito `response`. Esses métodos são resumidos na tabela abaixo:

Objeto Implícito	Método	Descrição
<code>request</code>	<code>getCookies()</code>	retorna uma matriz de cookies acessíveis da página
<code>response</code>	<code>addCookie()</code>	envia um cookie para o navegador para armazenagem/modificação

### A classe Cookie

Manipulamos um cookie através de instâncias da classe `javax.servlet.http.Cookie` (calma, você não precisa se preocupar com isso. É que o container JSP insere o comando `"import javax.servlet.http.*;"` automaticamente no servlet associado que é gerado na compilação da página JSP).

Essa classe fornece apenas um tipo de construtor que recebe duas variáveis do tipo `String`, que representam o nome e o valor do cookie. O cookie tem a seguinte sintaxe:

```
Cookie cookie = new Cookie("nome da fera" , "valor da fera");
```

Abaixo apresentamos os métodos fornecidos pela classe `Cookie`:

Método	Descrição
getName()	retorna o nome do cookie
getValue()	retorna o valor armazenado no cookie
getDomain()	retorna o servidor ou domínio do qual o cookie pode ser acessado
getPath()	retorna o caminho deURL do qual o cookie pode ser acessado
getSecure()	indica se o cookie acompanha solicitações HTTP ou HTTPS.
setValue()	atribui um novo valor para o cookie
setDomain()	define o servidor ou domínio do qual o cookie pode ser acessado
setPath(nome do path)	define o caminho de URL do qual o cookie pode ser acessado
setMaxAge(inteiro)	define o tempo restante (em segundos) antes que o cookie expire
setSecure(nome)	retorna o valor de um único cabeçalho de solicitação como um número inteiro

Depois de construir uma nova instância, ou modificar uma instância recuperada através do método `getCookies()`, é necessário usar o método `addCookie( )` do objeto `response`, com a finalidade salvar no navegador do usuário as alterações feitas no cookie. Para apagar um cookie utilizamos a seguinte técnica: chamamos o método `"setMaxAge(0)"` com valor zero e depois mandamos gravar chamando o método `"addCookie( )"`. Isso faz com que o cookie seja gravado e imediatamente (após zero segundos) expira.

### Definindo um cookie

O primeiro passo, então, ao usar um cookie dentro de uma página, é defini-lo. Isto é feito criando uma instância da classe `Cookie` e chamando os métodos `"sets"` para definir os valores de seus atributos.

Vamos ver o seguinte exemplo:

```
<%
String email = request.getParameter("email");
String cookieName = "cookieJSPBrasil";
Cookie cookieJSP = new Cookie(cookieName, email);

cookieJSP.setMaxAge(7 * 24 * 60 * 60); //define o tempo de vida como 7 dias
(604800 segundos)
cookieJSP.setVersion(0); //versão 0 da especificação de cookie
cookieJSP.setSecure(false); //indica que o cookie deve ser transferido pelo
protocolo HTTP padrão
cookieJSP.setComment("Email do visitante"); //insere um comentário para o cookie
response.addCookie(cookieJSP); //grava o cookie na máquina do usuário
%>

<html>
<head>
<title>Grava Cookie</title>
</head>

<body>

<h1>Grava Cookie</h1>
Esta página grava um cookie na sua máquina.<br>
<a href='readcookie.jsp'>Lê conteúdo do cookie</a>
</body>
</html>
```

Código do arquivo `addcookie.jsp`

Vamos criar uma página html com um formulário que irá fornecer um email que será gravado pela página "addcookie.jsp":

```
<html>
<body>

<form action="addcookie.jsp">
<input type="text" name="email">
<input type="submit" value="ok">

</body>
</html>
```

arquivo mailtcookie.html

No exemplo acima o cookie é identificado pelo nome cookieJSP e recebe o valor passado pelo usuário através de um formulário.

### Recuperando um Cookie

A página jsp vista anteriormente tem a finalidade de receber um valor (email) passado através de um formulário de uma página html. Este valor é armazenado de forma persistente em um cookie, e pode ser acessado pelas outras páginas JSP que compartilham o domínio e o caminho originalmente atribuídos ao cookie.

Os cookies são recuperados através do método `getCookies()` do objeto implícito `request`. A página abaixo mostra um exemplo de recuperação do valor de um cookie.

```
<%
String cookieName = "cookieJSPBrasil";

Cookie listaPossiveisCookies[] = request.getCookies();
Cookie cookieJSP = null;

if (listaPossiveisCookies != null) {
//quando não existe cookies associados o método getCookies() retorna um valor
null
int numCookies = listaPossiveisCookies.length;

for (int i = 0 ; i < numCookies ; ++i) {
if (listaPossiveisCookies[i].getName().equals(cookieName)) { //procura pelo
cookie
cookieJSP = listaPossiveisCookies[i];
break;
}
}
}
%>

<html>
<body>

<h1>Lê Cookie</h1>
<% if (cookieJSP != null) { %>
A pagina "addcookie" gravou o seguinte email: <%= cookieJSP.getValue() %>
<% }
else { %>
O cookie não gravou ou o prazo do cookie expirou.

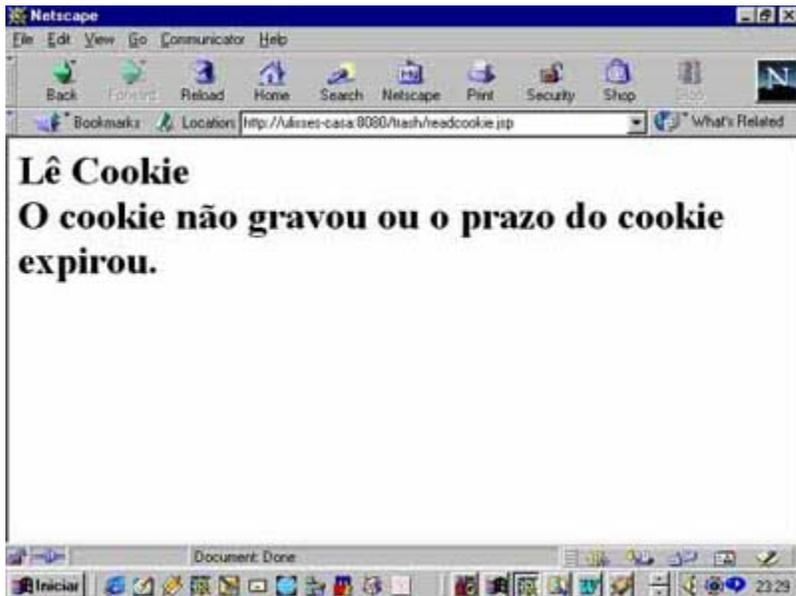
<% } %>

</body>
</html>
```

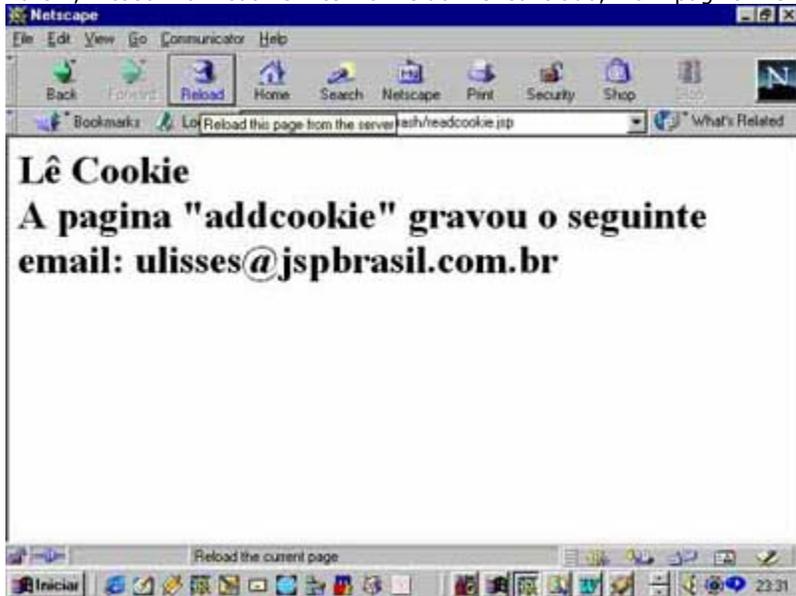
código do arquivo "readcookie.jsp"

O primeiro scriptlet nesta página recupera os cookies associados aquela página e depois tenta encontrar um cookie identificado pelo nome "cookieJSPBrasil".

Caso o cookie não esteja presente a página JSP exibirá a seguinte tela:



Porém, caso o cookie tenha sido encontrado, a página exibirá a seguinte tela:



### Considerações finais sobre cookies

Apesar da praticidade de se utilizar os cookies oferecidos pelo protocolo HTTP, devemos fazer algumas considerações quanto a sua utilização:

- o tamanho dos dados armazenados (nome e valor) não devem ultrapassar 4K.
- o navegador pode armazenar múltiplos cookies, contanto obedece um certo limite. Um navegador armazena até 20 cookies por configuração de domínio. O navegador armazena também 300 cookies em geral. Se qualquer um destes dois limites forem alcançados, espera-se que o navegador exclua cookies menos utilizados.
- o domínio atribuído a um cookie deve ter pelo menos dois pontos. Quando nenhum domínio é especificado na criação de um cookie através do método `setDomain(nome do domínio)`, então ele só poderá ser lido apenas pelo host que originalmente o definiu.