



Tutoriais

Enviando Email com JSP

por: César Medeiros

O envio de email, nos dias de hoje, para quem utiliza a internet, é algo comum e corriqueiro. É bastante comum encontrarmos sites/homepages a partir da qual podemos enviar um email ou enviar aquela página html para um amigo. Vale citar o fato que o envio de email via páginas html/jsp pode ser uma excelente ferramenta para webmasters que precisam enviar periodicamente mensagem padrão para grupos de usuários.

No decorrer deste tutorial trataremos dois exemplos: no primeiro, construiremos uma páginas jsp de onde será possível enviar emails; no segundo exemplo, faremos uma pequena aplicação que faz a leitura dos usuários de um banco de dados e envia uma mensagem personalizada para todos.

O envio do(s) email(s) será efetuado através da API JavaMail (<http://www.javasoft.com/products/javamail/index.html>), neste link tem todas as informações necessárias para a confecção de aplicações de que enviem e leiam emails, utilizando esta API você poderá fazer seu próprio WebMail, personalizando-o da forma como desejar.

Exemplo 1:

Neste exemplo construiremos três arquivos:

1. uma página html simples (preencheremail.html);
2. uma página JSP (enviaremail.jsp);
3. uma classe Java (Email.java).

O primeiro destes, a página html, terá um formulário com três objetos do tipo "TEXT", um do tipo "TEXTAREA" e um botão do tipo "SUBMIT" para enviar o formulário com os dados do email. Os dados que forem preenchidos neste formulário serão tratados pelo handler processador de formulário da página JSP (enviaremail.jsp) que foi definido como o action do formulário.

```
<html>
<body>

<form action="enviaremail.jsp" method="post">
<table border="0" align="center" bgcolor="tan">
<tr>
<td><b>Para..:</b></td>
<td><input type="Text" name="para"</td>
</tr>
<tr>
<td><b>De..:</b></td>
<td><input type="Text" name="de"</td>
</tr>
<tr>
<td><b>Assunto..:</b></td>
<td><input type="Text" name="assunto"</td>
</tr>
<tr>
<td colspan="2">
<textarea name="mensagem" rows=10 cols=45></textarea>
</td>
</tr>
</table>
<center> <input type="Submit" value="Enviar Email"></center>
</form>

</body>
</html>
```

Código do arquivo preencheremail.html

Abaixo segue o código da página JSP (enviaremail.jsp) com a lógica de programação responsável por criar e enviar o "email":

```

<html>
<body>

<%@ page import="jspbrasil.Email" %>
<jsp:useBean id="email" scope="session" class="jspbrasil.Email"/>
<%
try {
String mailServer = "mail.seumailserver.com.br"
String assunto = request.getParameter("assunto")
String para = { request.getParameter("para") };
String de = request.getParameter("de");
String mensagem =request.getParameter("mensagem");
email.sendSimpleMail(mailServer, assunto, para, de, mensagem);
%>
<p>Email enviado com Sucesso !!!</p>
<%
}
catch (AdressException e) { %>
<p>Endereço de Email inválido</p>
<%}%>
<%
catch (MessagingException e) { %>
<p>Impossível enviar o email.</p>
<%}%>

</body>
</html>

```

Código do arquivo enviaremail.jsp

No arquivo acima, optamos por levantar as exceções na própria página jsp de onde se envia o email, poderíamos ter definido uma página de erro para fazer este tratamento. São dois os possíveis erros decorrentes da tentativa de enviar o email:

O código abaixo, ilustra o tratamento da exceção gerada por endereço de email inválido.

```

<%
catch (AdressException e) { %>
<p>Endereço de Email inválido</p>
<%}%>

```

No trecho de código seguinte fazemos o tratamento da exceção gerada pelo não envio da mensagem, essa exceção pode ser causada por vários problemas, tais como: problema no servidor de email, problemas na rede entre outros.

```

<%
catch (MessagingException e) { %>
<p>Impossível enviar o email.</p>
<%}%>

```

Abaixo, mostraremos a listagem do método sendSimpleMail, que é o responsável por enviar o(s) email(s), neste método estão descritos os passos básicos para a construção de objetos e métodos para o envio de email; lembro, mais uma vez, que uma lida na API JavaMail (<http://www.javasoft.com/products/javamail/index.html>), é indispensável para quem pretende fazer algo mais elaborado relativo ao envio ou recebimento de emails.

```

package jspbrasil;

import javax.mail.*;
import javax.mail.internet.*;
import java.util.*;

public class Email {

    /*
     * . . .
     */

    public void sendSimpleMail (String mailServer, String subject,
                               String to,String from, String mensagem)
        throws AddressException, MessageException {

        Properties mailProps = new Properties();
        //definição do mailserver

        mailProps.put("mail.smtp.host", mailServer)

        Session mailSession = Session.getDefaultInstance(mailProps, null);

        //As duas linhas seguintes de código, colocam no
        //formato de endereços,
        //supostamente válidos, de email os dados
        //passados pelos parâmetros to e from.
        InternetAddress destinatario = new InternetAddress (to);
        InternetAddress remetente = new InternetAddress (from);

        //As duas linhas de código a seguir, são
        //responsáveis por setar os atributos e
        //propriedades necessárias do objeto message
        //para que o email seja enviado.
        //inicialização do objeto Message
        Message message = new MimeMessage (mailSession);

        //Definição de quem está enviando o email
        message.setFrom(remetente);

        //define o(s) destinatário(s) e qual o tipo do
        //destinatário.
        //os possíveis tipos de destinatário: TO, CC, BCC

        message.setRecipient( Message.RecipientType.TO, destinatario );
        //definição do assunto do email

        message.setSubject (subject);
        //definição do conteúdo da mensagem e do
        //tipo da mensagem

        message.setContent (mensagem.toString(), "text/plain");
        //a linha de código seguinte é a responsável
        //pelo envio do email

        Transport.send (message);
    }

}

```

Código do arquivo Email.java

Na segunda parte desse tutorial mostraremos um segundo exemplo, onde é possível enviar emails personalizados para uma lista de usuários a partir de uma consulta a um banco de dados.