Se você não tem seu próprio servidor SMTP, você pode usar o do Google como paliativo. Este tutorial vai te mostrar como fazer isso usando o JBoss AS 7.1.1

Primeiro, nós precisamos configurar um **mail session**. Pare o JBoss se ele estiver rodando, abra JBOSS/standalone/configuration/**standalone.xml.** Busque pelo trecho mostrado abaixo:

|  |
| --- |
| <mail-session jndi-name="java:jboss/mail/Default">  <smtp-server outbound-socket-binding-ref="mail-smtp"/>  </mail-session> |

Troque para conforme destacado abaixo

|  |
| --- |
| <mail-session jndi-name="java:jboss/mail/Default">  <smtp-server **ssl="true"** outbound-socket-binding-ref="mail-smtp">  **<login name="your\_google\_user\_id@gmail.com" password="your\_google\_password"/>**  </smtp-server>  </mail-session> |

Basicamente, você está habilitando o SSL e entrando com as informações da sua conta do Google.

Continue buscando por ‘smtp’ e você verá como o endereço do servidor SMTP é definido.

|  |
| --- |
| <outbound-socket-binding name="mail-smtp">  <remote-destination host="localhost" port="25"/>  </outbound-socket-binding> |

Troque com se segue:

|  |
| --- |
| <outbound-socket-binding name="mail-smtp">  <remote-destination host="**smtp.gmail.com**" port="**465**"/>  </outbound-socket-binding> |

Salve as alterações e reinicie o servidor.

Escreva um pequeno Servlet para testar o envio de email. Claro, use endereços de e-mails validos.

|  |
| --- |
| import java.io.IOException;  import java.io.PrintWriter;  import javax.annotation.Resource;  import javax.mail.Address;  import javax.mail.Message;  import javax.mail.Session;  import javax.mail.Transport;  import javax.mail.internet.InternetAddress;  import javax.mail.internet.MimeMessage;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  @WebServlet("/MailTest")  public class MailTest extends HttpServlet {  private static final long serialVersionUID = 1L;  @Resource(lookup = "java:jboss/mail/Default")  private Session mailSession;  public void service(HttpServletRequest req, HttpServletResponse res)  throws ServletException, IOException {  PrintWriter out = res.getWriter();  try {  MimeMessage m = new MimeMessage(mailSession);  Address from = new InternetAddress("bibhas@monkeyman.com");  Address[] to = new InternetAddress[] { new InternetAddress(  "bibhas@example.com") };  m.setFrom(from);  m.setRecipients(Message.RecipientType.TO, to);  m.setSubject("Google JavaMail Test");  m.setContent("Test from inside JBoss AS7 Server", "text/plain");  Transport.send(m);  out.println("Mail Sent Successfully.");  } catch (javax.mail.MessagingException e) {  e.printStackTrace(out);  }  }  } |

Note que o Google vai ignorar o endereço ‘FROM’ definido no código e sempre usar sua conta no campo ‘FROM’.

**Sending E-mail Using Google SMTP and JBoss AS 7**

If you don’t have your own SMTP server, you can use Google’s in a pinch. This tutorial will show you how to do that using JBoss AS 7.

First, we will need to configure a mail session. Stop JBoss if it is running. Open *JBOSS*/standalone/configuration/standalone.xml. Search for smtp. You will see a sample mail session defined like this:

<mail-session jndi-name="java:jboss/mail/Default">

<smtp-server outbound-socket-binding-ref="mail-smtp"/>

</mail-session>

Change it like this. I have highlighted the relevant changes.

<mail-session jndi-name="java:jboss/mail/Default">

<smtp-server **ssl="true"** outbound-socket-binding-ref="mail-smtp">

**<login name="your\_google\_user\_id@gmail.com" password="your\_google\_password"/>**

</smtp-server>

</mail-session>

Basically, you are enabling SSL and entering your Google account information.

Keep searching for smtp and you will see how the SMTP server address is defined.

<outbound-socket-binding name="mail-smtp">

<remote-destination host="localhost" port="25"/>

</outbound-socket-binding>

Change it as follows.

<outbound-socket-binding name="mail-smtp">

<remote-destination host="**smtp.gmail.com**" port="**465**"/>

</outbound-socket-binding>

Save changes. Start the server.

Write a small Servlet to test things out. Of course, use valid e-mail addresses.

import java.io.IOException;

import java.io.PrintWriter;

import javax.annotation.Resource;

import javax.mail.Address;

import javax.mail.Message;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/MailTest")

public class MailTest extends HttpServlet {

private static final long serialVersionUID = 1L;

@Resource(lookup = "java:jboss/mail/Default")

private Session mailSession;

public void service(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

PrintWriter out = res.getWriter();

try {

MimeMessage m = new MimeMessage(mailSession);

Address from = new InternetAddress("bibhas@monkeyman.com");

Address[] to = new InternetAddress[] { new InternetAddress(

"bibhas@example.com") };

m.setFrom(from);

m.setRecipients(Message.RecipientType.TO, to);

m.setSubject("Google JavaMail Test");

m.setContent("Test from inside JBoss AS7 Server", "text/plain");

Transport.send(m);

out.println("Mail Sent Successfully.");

} catch (javax.mail.MessagingException e) {

e.printStackTrace(out);

}

}

}

Note, Google will ignore the from address set in the code and always use your account as the from field.