

JSP 发送邮件

虽然使用JSP实现邮件发送功能很简单，但是需要有JavaMail API，并且需要安装JavaBean Activation Framework。

- 您可以从 [Java 网站](#) 下载最新版本的 [JavaMail](#)，打开网页右侧有个 **Downloads** 链接，点击它下载。
- 您可以从 [Java 网站](#) 下载最新版本的 [JAF \(版本 1.1.1\)](#)。

你也可以使用本站提供的下载链接：

- [JavaMail mail.jar 1.4.5](#)
- [JAF \(版本 1.1.1\) activation.jar](#)

下载并解压这些文件，在根目录下，您将会看到一系列jar包。将mail.jar包和activation.jar包加入CLASSPATH变量中。

发送一封简单的邮件

这个例子展示了如何从您的机器发送一封简单的邮件。它假定localhost已经连接至网络并且有能力发送一封邮件。与此同时，请再一次确认mail.jar包和activation.jar包已经添加进CLASSPATH变量中。

```
<%@ page import="java.io.*,java.util.*,javax.mail.*"%>
<%@ page import="javax.mail.internet.*,javax.activation.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>
<%
    String result;
    // 收件人的电子邮件
    String to = "abcd@gmail.com";

    // 发件人的电子邮件
    String from = "mcmohd@gmail.com";

    // 假设你是从本地主机发送电子邮件
    String host = "localhost";

    // 获取系统属性对象
    Properties properties = System.getProperties();

    // 设置邮件服务器
    properties.setProperty("mail.smtp.host", host);

    // 获取默认的Session对象。
    Session mailSession = Session.getDefaultInstance(properties);

    try{
        // 创建一个默认的MimeMessage对象。
        MimeMessage message = new MimeMessage(mailSession);
```

```
// 设置 From: 头部的header字段
message.setFrom(new InternetAddress(from));
// 设置 To: 头部的header字段
message.addRecipient(Message.RecipientType.TO,
                    new InternetAddress(to));
// 设置 Subject: header字段
message.setSubject("This is the Subject Line!");
// 现在设置的实际消息
message.setText("This is actual message");
// 发送消息
Transport.send(message);
result = "Sent message successfully....";
}catch (MessagingException mex) {
    mex.printStackTrace();
    result = "Error: unable to send message....";
}
%>
<html>
<head>
<title>Send Email using JSP</title>
</head>
<body>
<center>
<h1>Send Email using JSP</h1>
</center>
<p align="center">
<%
    out.println("Result: " + result + "\n");
%>
</p>
</body>
</html>
```

现在访问<http://localhost:8080/SendEmail.jsp>，它将会发送一封邮件给abcd@gmail.com 并显示如下结果：

```
Send Email using JSP
Result: Sent message successfully....
```

如果想要把邮件发送给多人，下面列出的方法可以用来指明多个邮箱地址：

```
void addRecipients(Message.RecipientType type,
                  Address[] addresses)
throws MessagingException
```

参数的描述如下：

- type：这个值将会被设置成TO，CC,或BCC。CC代表副本，BCC代表黑色副本，例子程序中使用的是TO。

- addresses：这是一个邮箱地址的数组，当指定邮箱地址时需要使用InternetAddress()方法。

发送一封HTML邮件

这个例子发送一封简单的HTML邮件。它假定您的localhost已经连接至网络并且有能力发送邮件。与此同时，请再一次确认mail.jar包和activation.jar包已经添加进CLASSPATH变量中。

这个例子和前一个例子非常相似，不过在这个例子中我们使用了setContent()方法，将"text/html"做为第二个参数传给它，用来表明消息中包含了HTML内容。

```
<%@ page import="java.io.*,java.util.*,javax.mail.*"%>
<%@ page import="javax.mail.internet.*,javax.activation.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>
<%
    String result;
    // 收件人的电子邮件
    String to = "abcd@gmail.com";

    // 发件人的电子邮件
    String from = "mcmohd@gmail.com";

    // 假设你是从本地主机发送电子邮件
    String host = "localhost";

    // 获取系统属性对象
    Properties properties = System.getProperties();

    // 设置邮件服务器
    properties.setProperty("mail.smtp.host", host);

    // 获取默认的Session对象。
    Session mailSession = Session.getDefaultInstance(properties);

    try{
        // 创建一个默认的MimeMessage对象。
        MimeMessage message = new MimeMessage(mailSession);
        // 设置 From: 头部的header字段
        message.setFrom(new InternetAddress(from));
        // 设置 To: 头部的header字段
        message.addRecipient(Message.RecipientType.TO,
                               new InternetAddress(to));
        // 设置 Subject: header字段
        message.setSubject("This is the Subject Line!");

        // 设置 HTML消息
        message.setContent("<h1>This is actual message</h1>",
                           "text/html" );

        // 发送消息
```

```
        Transport.send(message);
        result = "Sent message successfully....";
    }catch (MessagingException mex) {
        mex.printStackTrace();
        result = "Error: unable to send message....";
    }
}%>
<html>
<head>
<title>Send HTML Email using JSP</title>
</head>
<body>
<center>
<h1>Send Email using JSP</h1>
</center>
<p align="center">
<%
    out.println("Result: " + result + "\n");
%>
</p>
</body>
</html>
```

现在你可以尝试使用以上JSP文件来发送HTML消息的电子邮件。

在邮件中包含附件

这个例子告诉我们如何发送一封包含附件的邮件。

```
<%@ page import="java.io.*,java.util.*,javax.mail.*"%>
<%@ page import="javax.mail.internet.*,javax.activation.*"%>
<%@ page import="javax.servlet.http.*,javax.servlet.*" %>
<%
    String result;
    // 收件人的电子邮件
    String to = "abcd@gmail.com";

    // 发件人的电子邮件
    String from = "mcmohd@gmail.com";

    // 假设你是从本地主机发送电子邮件
    String host = "localhost";

    // 获取系统属性对象
    Properties properties = System.getProperties();

    // 设置邮件服务器
    properties.setProperty("mail.smtp.host", host);
```

```
// 获取默认的Session对象。
Session mailSession = Session.getDefaultInstance(properties);

try{
    // 创建一个默认的MimeMessage对象。
    MimeMessage message = new MimeMessage(mailSession);

    // 设置 From: 头部的header字段
    message.setFrom(new InternetAddress(from));

    // 设置 To: 头部的header字段
    message.addRecipient(Message.RecipientType.TO,
        new InternetAddress(to));

    // 设置 Subject: header字段
    message.setSubject("This is the Subject Line!");

    // 创建消息部分
    BodyPart messageBodyPart = new MimeBodyPart();

    // 填充消息
    messageBodyPart.setText("This is message body");

    // 创建多媒体消息
    Multipart multipart = new MimeMultipart();

    // 设置文本消息部分
    multipart.addBodyPart(messageBodyPart);

    // 附件部分
    messageBodyPart = new MimeBodyPart();
    String filename = "file.txt";
    DataSource source = new FileDataSource(filename);
    messageBodyPart.setDataHandler(new DataHandler(source));
    messageBodyPart.setFileName(filename);
    multipart.addBodyPart(messageBodyPart);

    // 发送完整消息
    message.setContent(multipart );

    // 发送消息
    Transport.send(message);
    String title = "Send Email";
    result = "Sent message successfully....";
}catch (MessagingException mex) {
    mex.printStackTrace();
    result = "Error: unable to send message....";
}
```

```
%>
<html>
<head>
<title>Send Attachment Email using JSP</title>
</head>
<body>
<center>
<h1>Send Attachment Email using JSP</h1>
</center>
<p align="center">
<%
    out.println("Result: " + result + "\n");
%>
</p>
</body>
</html>
```

用户认证部分

如果邮件服务器需要用户名和密码来进行用户认证的话，可以像下面这样来设置：

```
props.setProperty("mail.user", "myuser");
props.setProperty("mail.password", "mypwd");
```

使用表单发送邮件

使用HTML表单接收一封邮件，并通过request对象获取所有邮件信息：

```
String to = request.getParameter("to");
String from = request.getParameter("from");
String subject = request.getParameter("subject");
String messageText = request.getParameter("body");
```

获取以上信息后，您就可以使用前面提到的例子来发送邮件了。

[← JSP 自动刷新](#)[JSP 连接数据库 →](#)**1 篇笔记****写笔记**

//163邮箱发送带有附件的邮件到qq邮箱

```
<%@ page import="java.util.*" %>
<%@ page import="javax.mail.*" %>
```

```
<%@ page import="javax.mail.internet.*" %>
<%@ page import="javax.activation.*" %>
<%

    String result;
    // 收件人的电子邮件
    String to = "abc@qq.com";

    // 发件人的电子邮件名和密码
    String from = "abcd@163.com";
    String psd = "87800288b";
    //设置用户名
    String user = "abcd";

    Properties properties = new Properties();

    try {
        // 设置用户的认证方式
        properties.setProperty( "mail.smtp.auth", "true" );
        //设置传输协议
        properties.setProperty( "mail.transport.protocol", "smtp" );
        //SMTP邮件服务器
        properties.setProperty( "mail.smtp.host", "smtp.163.com" );
        //SMTP邮件服务器默认端口
        properties.setProperty( "mail.smtp.port", "25" );

        // 获取默认的Session对象。
        Session mailSession = Session.getDefaultInstance( properties );

        // 创建一个默认的MimeMessage对象。
        Message message = new MimeMessage( mailSession );
        // 根据session对象获取邮件传输对象Transport
        Transport transport = mailSession.getTransport();

        // 设置 From: 头部的header字段
        message.setFrom( new InternetAddress( from ) );
        // 设置 To: 头部的header字段
        message.addRecipient( Message.RecipientType.TO, new InternetAddress( to ) );
        // 设置 Subject: header字段
        message.setSubject( "This is the Subject Line!" );
        // 现在设置的实际消息

        BodyPart messageBodyPart = new MimeBodyPart();

        messageBodyPart.setText("This is message body");

        Multipart multipart = new MimeMultipart();

        multipart.addBodyPart(messageBodyPart);
```

```
messageBodyPart = new MimeBodyPart();
//使用绝对路径
String filename = "C:\\apache-tomcat-7.0.90\\webapps\\ROOT\\file.txt";
//添加文件到message
DataSource source = new FileDataSource(filename);

messageBodyPart.setDataHandler(new DataHandler(source));
messageBodyPart.setFileName(filename);

multipart.addBodyPart(messageBodyPart);
message.setContent(multipart);
message.setSentDate(new Date());

// 设置发件人的账户名和密码
transport.connect(user,psd);
// 发送邮件，并发送到所有收件人地址，message.getAllRecipients() 获取到的是在创建邮件对象时添加的所有收件人，抄送人，密送人
transport.sendMessage( message, message.getAllRecipients() );

result = "Sent message successfully...";
} catch (Exception e) {
    e.printStackTrace();
    result = "Error: unable to send message...";
}

%>
<html>
<head>
<title>Send Email using JSP</title>
</head>
<body>
<center>
<h1>Send Email using JSP</h1>
</center>
<p align="center">
<%
    out.println("Result: " + result + "\n");
%>
</p>
</body>
</html>
```

chankin 7个月前 (08-27)