

[illegible]

JavaMail 发送电子邮件内嵌图片

¥ 我要打赏

作者: thinkingg JavaMail 评论: 0 条 Windows技术QQ群: 227270512



下面的一个例子，是从你的机器发送HTML电子邮件，并内嵌图像。在这里，我们使用JangoSMTP服务器通过该电子邮件被发送到我们的目标电子邮件地址。

要发送电子邮件，其中包含一个内嵌的图像，遵循的步骤是：

- 获取一个Session
- 创建一个默认的MimeMessage对象，并设置发件人，收件人，主题（From, To, Subject）在消息中。
- 创建一个MimeMultipart的对象。
- 在我们的例子中，我们将在邮件的HTML部分和图像。因此，首先创建HTML内容，并在多部分对象将其设置为：

```
// first part (the html)
BodyPart messageBodyPart = new MimeBodyPart();
String htmlText = "<H1>Hello</H1><img src='cid:image'>";
messageBodyPart.setContent(htmlText, "text/html");
// add it
multipart.addBodyPart(messageBodyPart);
```

- 接下来创建一个DataHandler的如下添加图像：

```
// second part (the image)
messageBodyPart = new MimeBodyPart();
DataSource fds = new FileDataSource(
    "/home/manisha/javamail-mini-logo.png");

messageBodyPart.setDataHandler(new DataHandler(fds));
messageBodyPart.setHeader("Content-ID", "<image>");
```

- 接下来设置多重如下消息中：

```
message.setContent(multipart);
```

- 发送使用传输对象的消息。

创建Java类

创建一个Java类文件SendInlinImagesInEmail, 是其内容如下:

```
package com.yibai;

import java.util.Properties;
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.BodyPart;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

public class SendInlineImagesInEmail {
    public static void main(String[] args) {
        // Recipient's email ID needs to be mentioned.
        String to = "destinationemail@gmail.com";

        // Sender's email ID needs to be mentioned
        String from = "fromemail@gmail.com";
        final String username = "manishaspatil";//change accordingly
        final String password = "*****";//change accordingly

        // Assuming you are sending email through relay.jangosmtp.net
        String host = "relay.jangosmtp.net";

        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.port", "25");

        Session session = Session.getInstance(props,
            new javax.mail.Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication(username, password);
                }
            });

        try {

            // Create a default MimeMessage object.
            Message message = new MimeMessage(session);

            // Set From: header field of the header.
            message.setFrom(new InternetAddress(from));
```



```
// Set To: header field of the header.
message.setRecipients(Message.RecipientType.TO,
    InternetAddress.parse(to));

// Set Subject: header field
message.setSubject("Testing Subject");

// This mail has 2 part, the BODY and the embedded image
MimeMultipart multipart = new MimeMultipart("related");

// first part (the html)
BodyPart messageBodyPart = new MimeBodyPart();
String htmlText = "<H1>Hello</H1><img src='cid:image'>";
messageBodyPart.setContent(htmlText, "text/html");
// add it
multipart.addBodyPart(messageBodyPart);

// second part (the image)
messageBodyPart = new MimeBodyPart();
DataSource fds = new FileDataSource(
    "/home/manisha/javamail-mini-logo.png");

messageBodyPart.setDataHandler(new DataHandler(fds));
messageBodyPart.setHeader("Content-ID", "<image>");

// add image to the multipart
multipart.addBodyPart(messageBodyPart);

// put everything together
message.setContent(multipart);
// Send message
Transport.send(message);

System.out.println("Sent message successfully....");

} catch (MessagingException e) {
    throw new RuntimeException(e);
}
}
```

由于我们使用的是由主机提供商JangoSMTP提供的SMTP服务器，我们需要验证用户名和密码。javax.mail.PasswordAuthentication类用于验证的密码。

编译并运行

现在，我们则是准备好了，让我们编译上面的类。我已经保存了类SendInlineImagesInEmail.java 目录：/home/manisha/JavaMailAPIExercise. 我们需要 javax.mail.jar 和 activation.jar 文件在 classpath 中。执行下面的命令从命令提示符编译类（两个罐子被放置在/home/manisha/ 目录下）：

```
javac -cp /home/manisha/activation.jar:/home/manisha/javax.mail.jar: SendInlineImagesInEmail.java
```

现在，这个类被编译，执行下面的命令来运行：

```
java -cp /home/manisha/activation.jar:/home/manisha/javax.mail.jar: SendInlineImagesInEmail
```

验证输出

应该看到下面的消息命令控制台上：

```
Sent message successfully....
```

因为我通过JangoSMTP发送邮件到我的Gmail地址，下面的邮件会在我的Gmail帐户的收件箱中接收：



本站所有代码下载：请扫描本页面底部(右侧)二维码并关注微信公众号，回复："代码下载"获取。

上一篇：[JavaMail API 发送一个HTML电子邮件](#)

下一篇：[JavaMail 查询电子邮件](#)



登录



畅言一下

还没有评论，快来抢沙发吧！

易百教程正在使用畅言

Sign up

Already have account? [Sign in](#)

Email

Nickname

Username

Password

Confirm Password

captcha

QXf954

Git创建存储库 - Git教程™

1. Git创建存储库 - Git教程™

2. MongoDB更改用户密码和自定义数据 - MongoDB教程™

3. MongoDB启用身份验证 - MongoDB教程™

4. MongoDB添加用户 - MongoDB教程™

5. MongoDB管理用户和角色 - MongoDB教程™

6. [Ubuntu]error: C compiler cannot create executables - 易百教程™

7. Git快速入门 - Git教程™

1. Git创建存储库 - Git教程™

2. MongoDB更改用户密码和自定义数据 - MongoDB教程™

3. MongoDB启用身份验证 - MongoDB教程™

4. MongoDB添加用户 - MongoDB教程™

5. MongoDB管理用户和角色 - MongoDB教程™

6. [Ubuntu]error: C compiler cannot create executables - 易百教程™

7. Git快速入门 - Git教程™

热评话题

1. Git创建存储库 - Git教程™
2. MongoDB更改用户密码和自定义数据 - MongoDB教程™
3. MongoDB启用身份验证 - MongoDB教程™
4. MongoDB添加用户 - MongoDB教程™
5. MongoDB管理用户和角色 - MongoDB教程™
6. [Ubuntu]error: C compiler cannot create executables - 易百教程™
7. Git快速入门 - Git教程™

推荐/最新教程

- Spring JDBC
- Kotlin
- CouchDB
- JSoup
- NumPy
- JSF
- Cassandra
- Java

最新更新

- MongoDB
- MongoDB
- MongoDB
- MongoDB
- MongoDB
- MongoDB
- MongoDB
- MongoDB

站点信息

-
-
-
-
-

QQ学习群

- 227270512
- 418407075
- Java:172393525

关注微信公众号

