

JAVAMAIL API - SENDING EMAIL WITH INLINE IMAGESS

http://www.tutorialspoint.com/javamail_api/javamail_api_send_inlineimage_in_email.htm

Copyright © tutorialspoint.com

Here is an example to send an HTML email from your machine with inline image. Here we have used JangoSMTP server via which emails are sent to our destination email address. The setup is explained in the [Environment Setup](#) chapter.

To send a email with an inline image, the steps followed are:

- Get a Session
- Create a default MimeMessage object and set *From*, *To*, *Subject* in the message.
- Create a MimeMultipart object.
- In our example we will have an HTML part and an Image in the email. So first create the HTML content and set it in the multipart object as:

```
// 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);
```

- Next add the image by creating a Datahandler as follows:

```
// 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>");
```

- Next set the multipart in the message as follows:

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

- Send the message using the Transport object.

Create Java Class

Create a java class file **SendInlineImagesInEmail**, the contents of which are as follows:

```
package com.tutorialspoint;

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);  
}  
}  
}
```

As we are using the SMTP server provided by the host provider JangoSMTP, we need to authenticate the username and password. The *javax.mail.PasswordAuthentication* class is used to authenticate the password.

Compile and Run

Now that our class is ready, let us compile the above class. I've saved the class *SendInlineImagesInEmail.java* to directory : **/home/manisha/JavaMailAPIExercise**. We would need the jars *javax.mail.jar* and *activation.jar* in the classpath. Execute the command below to compile the class *both the jars are placed in /home/manisha/directory* from command prompt:

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

Now that the class is compiled, execute the below command to run:

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

Verify Output

You should see the following message on the command console:

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

As I'm sending an email to my gmail address through JangoSMTP, the following mail would be received in my gmail account inbox:



Loading [MathJax]/jax/output/HTML-CSS/jax.js