□ JavaMail
JavaMail API □□
JavaMail API □□□□
JavaMail API □□□
JavaMail API □□□□□□
JavaMail API □□□□□□□□□
JavaMail API □□□□□□□□□□
JavaMail API □□□□HTML□□□□
JavaMail □□□□□□□□□□□
JavaMail □□□□□□
JavaMail □□□□□□
JavaMail□□/□□
JavaMail □□□□□□/□□
JavaMail □□□□□□
JavaMail □□□□□□
JavaMail Gmail SMTP□□□
JavaMail □□□□□□□
JavaMail □□□□
JavaMail □□□□
JavaMail SMTP□□□
JavaMail IMAP□□□
JavaMail POP3□□□

JavaMail 获取电子邮件

¥ 我要打赏 作者：诸葛非卿 JavaMail 评论：0 条 Windows技术QQ群：227270512

在前面的章节中，我们学会了如何检查电子邮件。现在让我们来看看如何获取每封电子邮件和阅读其内容。让我们编写一个Java类FetchingEmail 读取以下类型的电子邮件：

- 简单的电子邮件
- 电子邮件与附件
- 电子邮件与内嵌图像

其次在代码的基本步骤如下：

- 获取Session对象。
- 创建POP3存储对象，并连接到存储。
- 创建文件夹对象，并在您的邮箱中打开相应的文件夹。
- 检索消息。
- 分别关闭文件夹和存储对象。

创建Java类

创建一个Java创建一个Java类文件FetchingEmail，内容都是如下：

```
package com.yiibai;

import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Date;
import java.util.Properties;

import javax.mail.Address;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.NoSuchProviderException;
import javax.mail.Part;
import javax.mail.Session;
import javax.mail.Store;

public class FetchingEmail {

   public static void fetch(String pop3Host, String storeType, String user,
      String password) {
      try {
         // create properties field
         Properties properties = new Properties();
         properties.put("mail.store.protocol", "pop3");
         properties.put("mail.pop3.host", pop3Host);
         properties.put("mail.pop3.port", "995");
         properties.put("mail.pop3.starttls.enable", "true");
         Session emailSession = Session.getDefaultInstance(properties);
         // emailSession.setDebug(true);

         // create the POP3 store object and connect with the pop server
         Store store = emailSession.getStore("pop3s");

         store.connect(pop3Host, user, password);

         // create the folder object and open it
         Folder emailFolder = store.getFolder("INBOX");
         emailFolder.open(Folder.READ_ONLY);

         BufferedReader reader = new BufferedReader(new InputStreamReader(
	      System.in));

         // retrieve the messages from the folder in an array and print it
         Message[] messages = emailFolder.getMessages();
         System.out.println("messages.length---" + messages.length);

         for (int i = 0; i < messages.length; i++) {
            Message message = messages[i];
            System.out.println("---------------------------------");
            writePart(message);
            String line = reader.readLine();
            if ("YES".equals(line)) {
               message.writeTo(System.out);
            } else if ("QUIT".equals(line)) {
               break;
            }
         }

         // close the store and folder objects
         emailFolder.close(false);
         store.close();

      } catch (NoSuchProviderException e) {
         e.printStackTrace();
      } catch (MessagingException e) {
         e.printStackTrace();
      } catch (IOException e) {
         e.printStackTrace();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
   public static void main(String[] args) {

      String host = "pop.gmail.com";// change accordingly
      String mailStoreType = "pop3";
      String username =
         "abc@gmail.com";// change accordingly
      String password = "*****";// change accordingly

      //Call method fetch
      fetch(host, mailStoreType, username, password);

   }

   /*
   * This method checks for content-type
   * based on which, it processes and
   * fetches the content of the message
   */
   public static void writePart(Part p) throws Exception {
      if (p instanceof Message)
         //Call methos writeEnvelope
         writeEnvelope((Message) p);

      System.out.println("----------------------------");
      System.out.println("CONTENT-TYPE: " + p.getContentType());

      //check if the content is plain text
      if (p.isMimeType("text/plain")) {
         System.out.println("This is plain text");
         System.out.println("---------------------------");
         System.out.println((String) p.getContent());
      }
      //check if the content has attachment
      else if (p.isMimeType("multipart/*")) {
         System.out.println("This is a Multipart");
         System.out.println("---------------------------");
         Multipart mp = (Multipart) p.getContent();
         int count = mp.getCount();
         for (int i = 0; i < count; i++)
            writePart(mp.getBodyPart(i));
      }
      //check if the content is a nested message
      else if (p.isMimeType("message/rfc822")) {
         System.out.println("This is a Nested Message");
         System.out.println("---------------------------");
         writePart((Part) p.getContent());
      }
      //check if the content is an inline image
      else if (p.isMimeType("image/jpeg")) {
         System.out.println("--------> image/jpeg");
         Object o = p.getContent();

         InputStream x = (InputStream) o;
         // Construct the required byte array
         System.out.println("x.length = " + x.available());
         int i = 0;
         byte[] bArray = new byte[x.available()];

         while ((i = (int) ((InputStream) x).available()) > 0) {
            int result = (int) (((InputStream) x).read(bArray));
            if (result == -1)
            break;
         }
         FileOutputStream f2 = new FileOutputStream("/tmp/image.jpg");
         f2.write(bArray);
      }
      else if (p.getContentType().contains("image/")) {
         System.out.println("content type" + p.getContentType());
         File f = new File("image" + new Date().getTime() + ".jpg");
         DataOutputStream output = new DataOutputStream(
            new BufferedOutputStream(new FileOutputStream(f)));
            com.sun.mail.util.BASE64DecoderStream test =
                 (com.sun.mail.util.BASE64DecoderStream) p
                  .getContent();
         byte[] buffer = new byte[1024];
         int bytesRead;
         while ((bytesRead = test.read(buffer)) != -1) {
            output.write(buffer, 0, bytesRead);
         }
      }
      else {
         Object o = p.getContent();
         if (o instanceof String) {
            System.out.println("This is a string");
            System.out.println("---------------------------");
            System.out.println((String) o);
         }
         else if (o instanceof InputStream) {
            System.out.println("This is just an input stream");
            System.out.println("---------------------------");
            InputStream is = (InputStream) o;
            is = (InputStream) o;
            int c;
            while ((c = is.read()) != -1)
               System.out.write(c);
         }
         else {
            System.out.println("This is an unknown type");
            System.out.println("---------------------------");
            System.out.println(o.toString());
         }
      }

   }
   /*
   * This method would print FROM,TO and SUBJECT of the message
   */
   public static void writeEnvelope(Message m) throws Exception {
      System.out.println("This is the message envelope");
      System.out.println("---------------------------");
      Address[] a;

      // FROM
      if ((a = m.getFrom()) != null) {
         for (int j = 0; j < a.length; j++)
         System.out.println("FROM: " + a[j].toString());
      }

      // TO
      if ((a = m.getRecipients(Message.RecipientType.TO)) != null) {
         for (int j = 0; j < a.length; j++)
         System.out.println("TO: " + a[j].toString());
      }

      // SUBJECT
      if (m.getSubject() != null)
         System.out.println("SUBJECT: " + m.getSubject());

   }

}
```

> 您可以通过取消注释语句上设置调试emailSession.setDebug(true);

编译并运行

现在，我们班是准备好了，让我们编译上面的类。我已经保存了类FetchingEmail.java目录： /home/manisha/JavaMailAPIExercise. 我们需要 javax.mail.jar andactivation.jar 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: FetchingEmail.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: FetchingEmail
```

验证输出

你应该看到下面的消息命令控制台上:

```
messages.length---3
---------------------------------
This is the message envelope
---------------------------
FROM: XYZ <xyz@gmail.com>
TO: ABC <abc@gmail.com>
SUBJECT: Simple Message
----------------------------
CONTENT-TYPE: multipart/alternative; boundary=047d7b343d6ad3e4ea04e8ec6579
This is a Multipart
---------------------------

----------------------------
CONTENT-TYPE: text/plain; charset=ISO-8859-1
This is plain text
---------------------------
Hi am a simple message string....

--
Regards
xyz

This is the message envelope
---------------------------
FROM: XYZ <xyz@gmail.com>
TO: ABC <abc@gmail.com>
SUBJECT: Attachement
----------------------------
CONTENT-TYPE: multipart/mixed; boundary=047d7b343d6a99180904e8ec6751
This is a Multipart
---------------------------

----------------------------
CONTENT-TYPE: text/plain; charset=ISO-8859-1
This is plain text
---------------------------
Hi I've an attachment.Please check

--
Regards
XYZ

----------------------------
CONTENT-TYPE: application/octet-stream; name=sample_attachement
This is just an input stream
---------------------------
Submit your Tutorials, White Papers and Articles into our Tutorials Directory. This is a tutori
als database where we are keeping all the tutorials shared by the internet community for the be
nefit of others.


This is the message envelope
---------------------------
FROM: XYZ <xyz@gmail.com>
TO: ABC <abc@gmail.com>
SUBJECT: Inline Image
----------------------------
CONTENT-TYPE: multipart/related; boundary=f46d04182582be803504e8ece94b
This is a Multipart
---------------------------

----------------------------
CONTENT-TYPE: text/plain; charset=ISO-8859-1
This is plain text
---------------------------
Hi I've an inline image


[image: Inline image 3]

--
Regards
XYZ

----------------------------
CONTENT-TYPE: image/png; name="javamail-mini-logo.png"
content typeimage/png; name="javamail-mini-logo.png"
```

在这里，你可以看到有三封邮件在邮箱中。首先一个简单的邮件消息 "Hi am a simple message string...."。第二个邮件有附件。附件的内容也印如上所示。第三个邮件有一个内嵌图像。

0

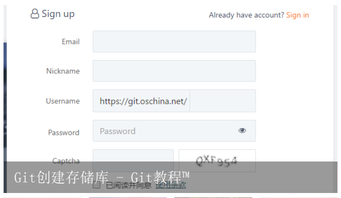
登录

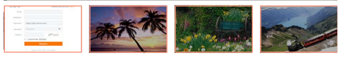
来说两句吧...

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 https://git.oschina.net/
Password Password
Captcha



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™