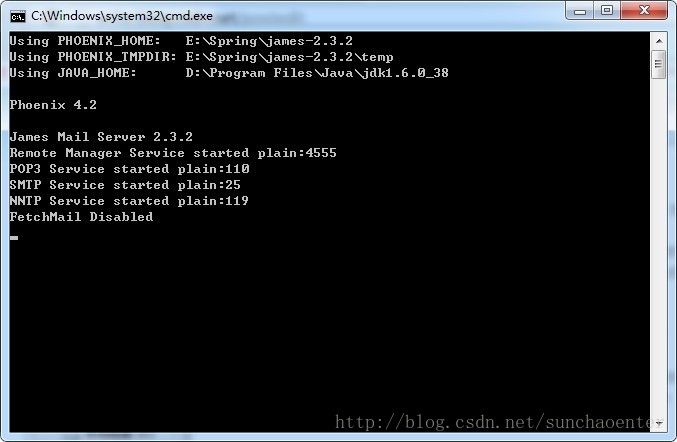
写这篇文章的背景是公司Android客户端需要实现一个功能，实现类似于密码找回或者用户注册完发送一个邮件给用户的功能，当然这些逻辑客户端只负责请求自己的服务端，自己的服务端再去请求邮件服务器。

邮件服务器使用的Apache的James，Java写的开源的，而且方便调用。远程调用使用的是Java Mail的API。

首先我们需要把James这个邮件服务器跑起来，官网下载地址：http://james.apache.org/，下载先来之后直接解压，直接运行bin\run.bat。出现如下图所示说明运行OK了。



如果运行不成功，可能是你的JDK或者环境变量没有配置好，这个我就不说了。

运行起来之后，会在apps\james\SAR-INF这个目录下生成一个config.xml文件，这就是邮件服务器的配置文件。打开它，修改一些配置：

1.修改<postmaster>localhost</postmaster>，改成你自己的服务器域名，比如：<postmaster>support@zhangyuan123.com</postmaster>

2.修改这些，也是改成自己的

<servernames autodetect="false" autodetectIP="false">

   <servername>zhangyuan123.com</servername>

</servernames>

3.修改dns server，这是为了可以发外网邮件，查看DNS地址使用ipconfig/all，然后在配置文件中dnsserver节点下加入server地址，比如：

**[html]** view plaincopy

1. **<dnsserver>**
2. **<servers>**
3. <!--Enter ip address of your DNS server, one IP address per server -->
4. <!-- element. -->
5. <!--
6. <server>127.0.0.1</server>
7. -->
8. **<server>**114.114.114.114**</server>**
9. **<server>**8.8.8.8**</server>**
10. **</servers>**
11. <!-- Change autodiscover to false if you would like to turn off autodiscovery -->
12. <!-- and set the DNS servers manually in the <servers> section -->
13. **<autodiscover>**false**</autodiscover>**
14. **<authoritative>**false**</authoritative>**
16. <!-- Maximum number of entries to maintain in the DNS cache -->
17. **<maxcachesize>**50000**</maxcachesize>**
18. **</dnsserver>**

4.找到C:\Windows\System32\drivers\etc\host文件，其中加入127.0.0.1   zhangyuan123.com

重新启动邮件服务器。

下面就添加一个用户到邮件服务器，James是使用telnet进行命令行操作的。win7默认没有打开telnet服务，可以搜一下如何打开。然后在命令行输入telnet zhangyuan123.com 4555，输入用户名root，密码root。添加一个用户使用命令：adduser admin admin，其他命令我就不说了。

添加完了之后我们就开始使用客户端模仿调用服务器发送一封邮件。

**[java]** view plaincopy

1. **package** org.sunchao;
3. **import** java.io.IOException;
4. **import** java.util.Properties;
6. **import** javax.mail.Authenticator;
7. **import** javax.mail.Folder;
8. **import** javax.mail.Message;
9. **import** javax.mail.MessagingException;
10. **import** javax.mail.PasswordAuthentication;
11. **import** javax.mail.Session;
12. **import** javax.mail.Store;
13. **import** javax.mail.internet.InternetAddress;
14. **import** javax.mail.internet.MimeMessage;
15. **import** javax.mail.internet.MimeUtility;
17. **public** **class** Mail {
18. // String host = "smtp.163.com"; // 163的smtp服务器
19. // 本机smtp服务器
20. **private** **static** **final** String HOST = "zhangyuan123.com";
21. // 邮件发送人的邮件地址
22. **private** **static** **final** String FROM = "admin@" + HOST;
23. // 邮件接收人的邮件地址
24. **private** **static** **final** String TO = "test@qq.com";
25. // 发件人的邮件帐户
26. **private** **static** **final** String USERNAME = "admin";
27. // 发件人的邮件密码
28. **private** **static** **final** String PASSWORD = "admin";
30. **public** **static** **void** main(String[] args) {
31. Mail.sendMail();
32. // Mail.getMail();
33. }
35. /\*\*
36. \* 发送邮件
37. \*/
38. **public** **static** **void** sendMail() {
39. // 创建Properties 对象
40. Properties props = System.getProperties();
41. // 添加smtp服务器属性
42. props.put("mail.smtp.host", HOST);
43. props.put("mail.smtp.auth", "true");
45. // 创建邮件会话
46. Session session = Session.getDefaultInstance(props,
47. **new** Authenticator() {
48. @Override
49. **public** PasswordAuthentication getPasswordAuthentication() {
50. **return** **new** PasswordAuthentication(USERNAME, PASSWORD);
51. }
52. });
53. **try** {
54. // 定义邮件信息
55. MimeMessage message = **new** MimeMessage(session);
56. message.setFrom(**new** InternetAddress(FROM));
57. message.addRecipient(Message.RecipientType.TO, **new** InternetAddress(
58. TO));
59. message.setSubject("I hava my own mail server");
60. message
61. .setText("From now, you have your own mail server, congratulation!");
62. // 发送消息
63. session.getTransport("smtp").send(message);
64. // Transport.send(message); //也可以这样创建Transport对象发送
65. System.out.println("SendMail Process Over!");
66. } **catch** (MessagingException e) {
67. e.printStackTrace();
68. }
69. }
71. /\*\*
72. \* 接收邮件
73. \*/
74. **public** **static** **void** getMail() {
75. String host = "localhost";
76. **final** String username = "admin";
77. **final** String password = "admin";
78. // 创建Properties 对象
79. Properties props = **new** Properties();
81. // 创建邮件会话
82. Session session = Session.getDefaultInstance(props,
83. **new** Authenticator() {
84. @Override
85. **public** PasswordAuthentication getPasswordAuthentication() {
86. **return** **new** PasswordAuthentication(username, password);
87. }
88. });
89. **try** {
90. // 获取邮箱的pop3存储
91. Store store = session.getStore("pop3");
92. store.connect(host, username, password);
93. // 获取inbox文件
94. Folder folder = store.getFolder("INBOX");
95. folder.open(Folder.READ\_ONLY); // 打开，打开后才能读取邮件信息
96. // 获取邮件消息
97. Message message[] = folder.getMessages();
98. **for** (**int** i = 0, n = message.length; i < n; i++) {
99. System.out.println(i + ": " + message[i].getFrom()[0] + "\t"
100. + message[i].getSubject());
101. **try** {
102. message[i].writeTo(System.out);
103. } **catch** (IOException e) {
104. e.printStackTrace();
105. }
106. }
107. // 关闭资源
108. folder.close(**false**);
109. store.close();
110. } **catch** (MessagingException e) {
111. e.printStackTrace();
112. }
113. System.out.println("GetMail Process Over!");
114. }
116. // 邮件主题中文字符转换
117. **public** **static** String transferChinese(String strText) {
118. **try** {
119. strText = MimeUtility.encodeText(**new** String(strText.getBytes(),
120. "GB2312"), "GB2312", "B");
121. } **catch** (Exception ex) {
122. ex.printStackTrace();
123. }
124. **return** strText;
125. }
126. }

把上面代码中qq邮箱地址换成自己的就行。最后run一把就OK了。但是QQ邮箱自身有过滤机制，它会认为来自zhangyuan123.com的邮件是垃圾邮件，你需要设置一下白名单：  
  
添加域名白名单：



再发每一次，QQ邮箱就可以收到了。

对于腾讯的过滤，如果不让他过滤，我估计得交钱给腾讯了，其他的163的，新浪的大家也可以试一下。