# 邮件准备

## 1、qq

### 1、首先，我们需要打开QQ邮箱的SMTP服务，因为QQ邮箱对于一般的用户都是默认关闭SMTP服务的。



### 2、需要验证个人信息



### 3、出来一个密码，这个密码非常重要的



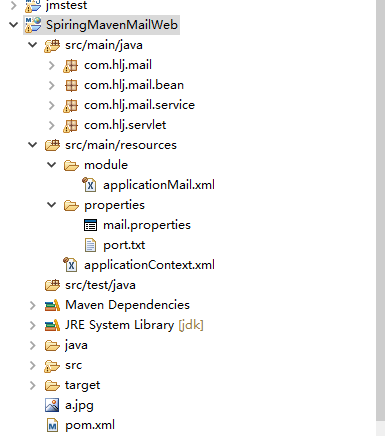
## 2、163邮箱，

### 1、同上这个授权码是自己设置的



# 问题解决：老子一直不能发送成功，原来是他麻痹的空格的问题，仔细检查propetries，发现赋值之后总是有空格，这他妈谁能发现。草，把所有尾巴后的空格删掉就Ok了





# 1、编辑propetries，设置我的发送邮箱的参数，这里使用了suhu、163、QQ邮箱，全部能够发送成功

## 1、参数说明

### mail.from 发件人地址（发送邮件时需要）。

### mail.host ：邮件交互的主机。要连接的SMTP服务器

### mail.password ：邮箱密码

### mail.smtp.auth：缺省是false，如果为true，尝试使用AUTH命令认证用户。

### mail.smtp.timeout ：Socket I/O超时值，单位毫秒，缺省值不超时。

### mail.username ：邮箱用户名（也就是发送者的邮箱）

### mail.transport.protocol：要装入session的协议（smtp、pop3、imap、nntp）。

### mail.smtp.port要连接的SMTP服务器的端口号，如果connect没有指明端口号就使用它，缺

## 2、编辑propetries

|  |
| --- |
| # all success  ##### sohu ###  #mail.from=zyj37651@sohu.com  #mail.host=smtp.sohu.com  #mail.password=\*\*\*\*\*\*  #mail.smtp.auth=true  #mail.smtp.timeout=25000  #mail.username=zyj37651@sohu.com      ##### 163 ###  #mail.from=mxzdhealer@163.com  #mail.host=smtp.163.com  #mail.password=\*\*\*\*\*\*  #mail.smtp.auth=true  #mail.smtp.timeout=25000  #mail.username=mxzdhealer@163.com    ##### qq ###  mail.from=1318830916@qq.com  mail.host=smtp.qq.com  #mail.password=pdrmthxordrujhbc  mail.password=huyhqvmgtynqicaa  mail.smtp.auth=true  mail.smtp.timeout=25000  mail.username=1318830916@qq.com |

# 2、编辑 applicationMail.xml

## 1、加载Properties文件，以及编辑一个简单的发送者

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans  xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns:p=*"http://www.springframework.org/schema/p"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>  <!-- 加载Properties文件 -->  <bean id=*"configurer"* class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>  <property name=*"locations"*>  <list>  <value>classpath:properties/mail.properties</value>  </list>  </property>  </bean>    <!-- 简单的发送的配置信息，包含各种邮件的内容 简单的发送。我这里只发送简单邮件中用到-->  <bean id=*"mailMessage"* class=*"org.springframework.mail.SimpleMailMessage"*>  <property name=*"from"*>  <value>${mail.from}</value>  </property>  <!-- 查看SimpleMailMessage源码还可以注入标题,内容等 -->  </bean> |

## 2、编辑邮件发送器JavaMailSenderImpl，并配置相关属性

|  |
| --- |
| <!-- 邮件发送器 163，qq，sohu -->  <!-- 申明JavaMailSenderImpl对象 -->  <bean id=*"mailSender"* class=*"org.springframework.mail.javamail.JavaMailSenderImpl"*>  <property name=*"host"* value=*"${mail.host}"* />  <property name=*"username"* value=*"${mail.username}"* />  <property name=*"password"* value=*"${mail.password}"* />  <property name=*"defaultEncoding"* value=*"UTF-8"* />  <property name=*"protocol"* value=*"smtp"*/>  <!-- <property name="port" value="25" /> --><!-- 默认提供端口支持，并且默认知道端口是多少，无需指定 -->  <property name=*"javaMailProperties"*>  <props>  <!-- 启动调试开关 ，这样控制台就会打印发送的日志-->  <prop key=*"mail.debug"*>true</prop>  <!-- 验证身份， 设置认证开关 -->  <prop key=*"mail.smtp.auth"*>${mail.smtp.auth}</prop>  <!-- 设置发送延时 -->  <prop key=*"mail.smtp.timeout"*>${mail.smtp.timeout}</prop>  <!-- <prop key="mail.smtp.port">25</prop> 上面的相关属性也可以设置在这里 -->    </props>  </property>  </bean> |

## 3、谷歌和其他的不太一样，也没用过，用的时候再说吧

|  |
| --- |
| <!-- 邮件发送器 gmail -->  <!-- <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">  <property name="host" value="${mail.host}" />  <property name="username" value="${mail.username}" />  <property name="password" value="${mail.password}" />  <property name="defaultEncoding" value="UTF-8"></property>  <property name="javaMailProperties">  <props>  <prop key="mail.smtp.auth">${mail.smtp.auth}</prop>  <prop key="mail.smtp.timeout">${mail.smtp.timeout}</prop>  <prop key="mail.smtp.port">465</prop>  <prop key="mail.smtp.socketFactory.port">465</prop>  <prop key="mail.smtp.socketFactory.fallback">false</prop>  <prop key="mail.smtp.socketFactory.class">javax.net.ssl.SSLSocketFactory</prop>  </props>  </property>  </bean> -->    </beans> |

# 3、编辑spring的配置文件，用来加载上面的applicationMail.xml（java中加载是spring的xml）

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:aop=*"http://www.springframework.org/schema/aop"*  xsi:schemaLocation=*"*  *http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans-3.1.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*  *http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd"*>  <context:annotation-config />  <context:component-scan base-package=*"com.hlj.servlet"* />  <context:component-scan base-package=*"com.hlj.mail"* />  <context:component-scan base-package=*"com.hlj.mail.other"* />    <import resource=*"classpath:module/applicationMail.xml"* />    </beans> |

# 4、设置邮件bean，发邮件的时候设置里面的属性值，就是发邮件的内容 get和set省略了

|  |
| --- |
| **public** **class** MailBean {  **private** String from; //发送者邮箱 ，不指定FromName，到了收件人列表中，会自动显示邮箱前半部分比如 mxzdhealer 后面的省略  **private** String fromName; //收件箱列表，所显示标题  **private** String[] toEmails; //接受者邮箱，设置数组，可以同时发送给多个人  **private** String subject; //邮件主题  **private** String text; //邮件内容  **private** Map<String,Object> attachment; //附件  **private** Map<String,Object> imageAttachment; //链接中的图片    **public** MailBean() {  }  **public** MailBean(String from, String fromName, String[] toEmails,  String subject, String text, Map<String, Object> attachment,  Map<String, Object> imageAttachment) {  **super**();  **this**.from = from;  **this**.fromName = fromName;  **this**.toEmails = toEmails;  **this**.subject = subject;  **this**.text = text;  **this**.attachment = attachment;  **this**.imageAttachment = imageAttachment;  } |
|  |

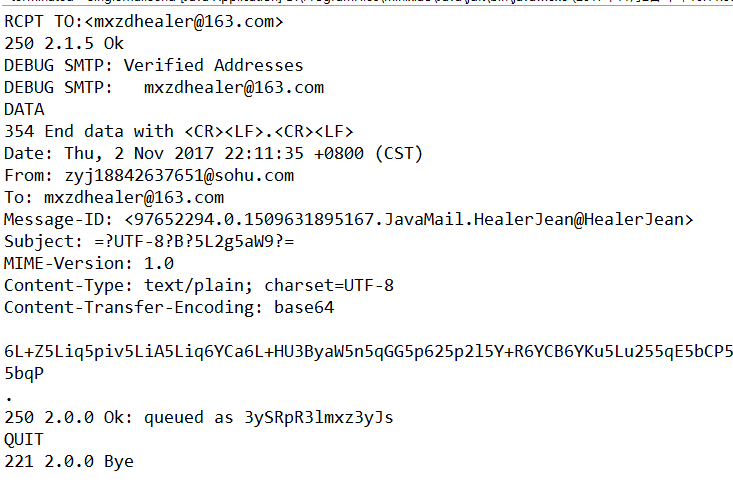
# 5、编辑发邮件的service

|  |
| --- |
| @Service  **public** **class** MailSenderService {  //利用系统的配置文件生成，可以直接使用  @Autowired  **private** JavaMailSenderImpl javaMailSenderImpl;    **public** **void** sendMail(MailBean mailBean) **throws** UnsupportedEncodingException, MessagingException{  MimeMessage msg = createMimeMessage(mailBean);  javaMailSenderImpl.send(msg);  }  /\*\*  \* 创建MimeMessage  \*/  **public** MimeMessage createMimeMessage(MailBean mailBean) **throws** MessagingException, UnsupportedEncodingException{  MimeMessage mimeMessage = javaMailSenderImpl.createMimeMessage();  MimeMessageHelper messageHelper = **new** MimeMessageHelper(mimeMessage, **true**, "UTF-8");    /\*\*  \* 谁发送的 setFrom 必须指定  \*/  messageHelper.**setFrom**(mailBean.getFrom(), mailBean.getFromName());  // 不指定FromName，到了收件人列表中，会自动显示邮箱前半部分比如 mxzdhealer @后面的省略  //messageHelper.setFromName(mailBean. getFromName());  messageHelper.setTo(mailBean.getToEmails());  //可以只给一个人发  //messageHelper.setTo("mxzdhealer@163.com");  messageHelper.setSubject(mailBean.getSubject()); //主题  messageHelper.setText(mailBean.getText(), **true**); //内容 html: true  /\*\*  \* 发送附件，添加到附件  \*/  Map<String,Object> attachment = mailBean.getAttachment();  **if**(attachment!=**null**){  **for**(String key:attachment.keySet()){  messageHelper.addAttachment(key, (FileSystemResource)attachment.get(key));  }  }  /\*\*  \* 链接中的图片  \*/  Map<String,Object> imageAttachment = mailBean.getImageAttachment();  **if**(imageAttachment!=**null**){  **for**(String key:imageAttachment.keySet()){  messageHelper.addInline(key, (FileSystemResource)imageAttachment.get(key));  }  }    **return** mimeMessage;  }    } |

# 6、不使用2、3中的bean和service，利用applicationMail.xml 中的，发送简单的邮件

|  |
| --- |
| **public** **class** SingleMailSend {  **static** ApplicationContext *context* = **new** ClassPathXmlApplicationContext(  "applicationContext.xml");      **static** MailSender *sender* = (MailSender) *context*.getBean("mailSender");  **static** SimpleMailMessage *mailMessage* = (SimpleMailMessage) *context*.getBean("mailMessage");  **public** **static** **void** main(String args[]) **throws** MessagingException {  *mailMessage*.setSubject("你好");  *mailMessage*.setText("这个是一个通过Spring框架来发送邮件的小程序");  *mailMessage*.setTo("mxzdhealer@163.com");  *sender*.send(*mailMessage*);  }  } |

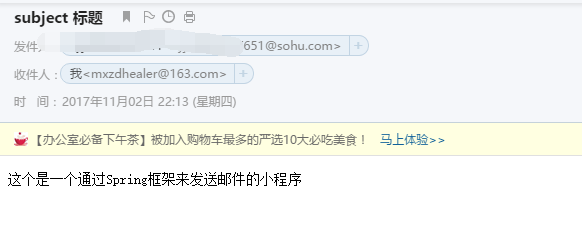
## 1、测试运行，观察控制台，发送成功



## 2、邮件接收成功，观察内容，因为没有指定FormName，所以收件人邮箱列表中就显示了发件人的邮箱



## 3、点开，观察内容



# 7、发送带有链接的图片，发送给单个人，不设置setFromName

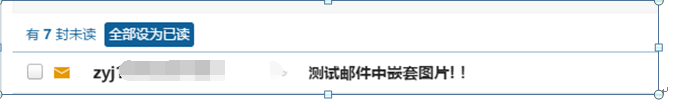
## 1、链接中图片有id，"<img src=cid:image/> 所以下面只要添加即可

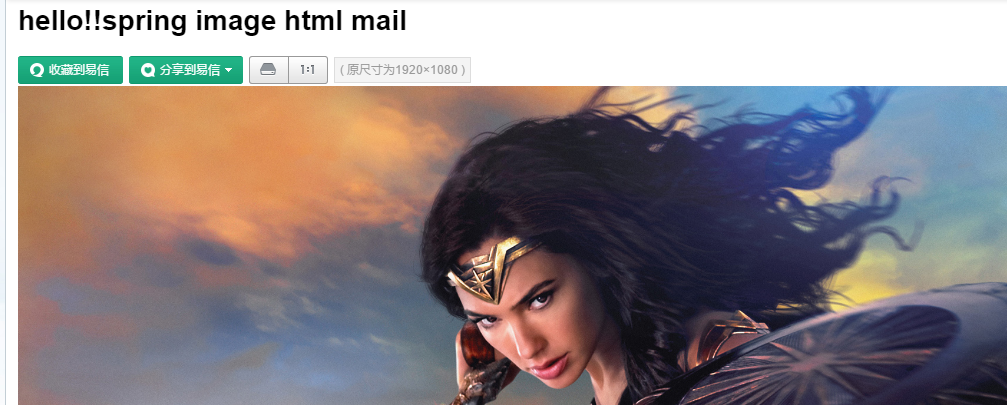
|  |
| --- |
| **public** **class** ImageMail {  **public** **static** **void** main(String[] args) **throws** MessagingException, UnsupportedEncodingException {  ApplicationContext ctx = **new** ClassPathXmlApplicationContext("applicationContext.xml");    MailSenderService mailSenderService = (MailSenderService) ctx.getBean("mailSenderService");  MailBean mailBean = **new** MailBean();    mailBean.setFrom("zyj17651@sohu.com");  mailBean.setFromName(**null**); //不设置name    mailBean.setSubject("测试邮件中嵌套图片!！");  mailBean.setText(  "<html><head></head><body><h1>hello!!spring image html mail</h1>"  +"<a href=http://www.baidu.com>baidu</a>" + "<img src=cid:image/></body></html>");  //可以发送给多人，这里只发送给一个人  mailBean.setToEmails(**new** String[]{"mxzdhealer@163.com"});  /\*\*  \* 添加上述链接中的图片  \*/  FileSystemResource img = **new** FileSystemResource(**new** File("a.jpg"));  Map<String,Object> imageaAttachment = **new** HashMap<String, Object>();  imageaAttachment.put("image", img);//添加图片链接  mailBean.setImageAttachment(imageaAttachment);    mailSenderService.sendMail(mailBean);  System.*out*.println("邮件发送成功...");  } |

## 2、测试运行，观察控制台



## 3、观察邮件





# 8、发送带有附件的邮件，发送两个附件，发送给多个人，设置setFromName

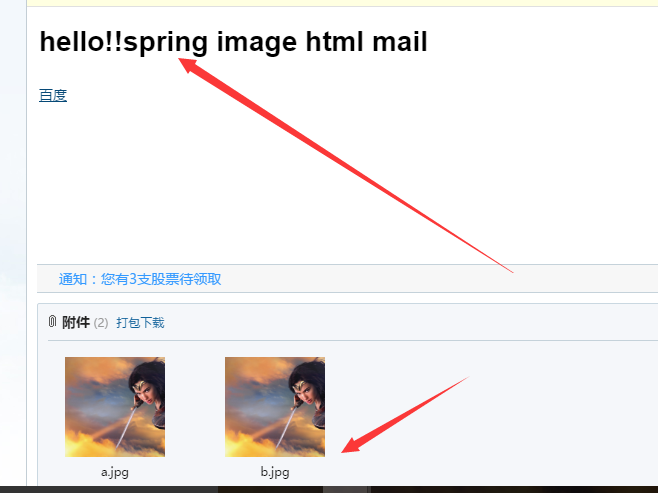
|  |
| --- |
| **public** **class** AttachmentMail {    **public** **static** **void** main(String[] args) **throws** MessagingException, UnsupportedEncodingException {  ApplicationContext ctx = **new** ClassPathXmlApplicationContext(  "applicationContext.xml");    MailSenderService mailSenderService = (MailSenderService) ctx.getBean("mailSenderService");  MailBean mailBean = **new** MailBean();    mailBean.setFrom("zyj188@sohu.com");  mailBean.setFromName("你好 HealerJean");  mailBean.setSubject("测试附件的发送!！");  mailBean.setToEmails(**new** String[]{"729088240@qq.com","mxzdhealer@163.com"});  mailBean.setText("<a href='www.baidu.com'><font color='red'>fdsfdsf</font></a>");  mailBean.setText(  "<html><head></head><body><h1>hello!!spring image html mail</h1>"  +"<a href=http://www.baidu.com>百度</a>" + "</body></html>");    //整个项目的路径下,添加附件  FileSystemResource img = **new** FileSystemResource(**new** File("a.jpg"));  Map<String,Object> attachment = **new** HashMap<String, Object>();  attachment.put("a.jpg", img);//添加到附件  attachment.put("b.jpg", img);//添加到附件  mailBean.setAttachment(attachment);    mailSenderService.sendMail(mailBean);    System.*out*.println("邮件发送成功...");  } |

## 1、测试运行，观察控制台



## 2、观察邮件





# 9、各种邮箱的端口

|  |
| --- |
| sina.com:  POP3服务器地址:pop3.sina.com.cn（端口：110）  SMTP服务器地址:smtp.sina.com.cn（端口：25）  sina.cn：  POP3服务器地址:pop3.sina.com（端口：110） ------- > pop.sina.com  SMTP服务器地址:smtp.sina.com（端口：25）  sinaVIP：  POP3服务器:pop3.vip.sina.com （端口：110）  SMTP服务器:smtp.vip.sina.com （端口：25）    sohu.com:  POP3服务器地址:pop3.sohu.com（端口：110）  SMTP服务器地址:smtp.sohu.com（端口：25）    126邮箱：  POP3服务器地址:pop.126.com（端口：110）  SMTP服务器地址:smtp.126.com（端口：25）    139邮箱：  POP3服务器地址：POP.139.com（端口：110）  SMTP服务器地址：SMTP.139.com(端口：25)  163.com:  POP3服务器地址:pop.163.com（端口：110）  SMTP服务器地址:smtp.163.com（端口：25）    QQ邮箱  POP3服务器地址：pop.qq.com（端口：110）  SMTP服务器地址：smtp.qq.com（端口：25）    QQ企业邮箱  POP3服务器地址：pop.exmail.qq.com （SSL启用 端口：995）  SMTP服务器地址：smtp.exmail.qq.com（SSL启用 端口：587/465）    yahoo.com:  POP3服务器地址:pop.mail.yahoo.com  SMTP服务器地址:smtp.mail.yahoo.com    yahoo.com.cn:  POP3服务器地址:pop.mail.yahoo.com.cn（端口：995）  SMTP服务器地址:smtp.mail.yahoo.com.cn（端口：587    HotMail  POP3服务器地址：pop3.live.com（端口：995）  SMTP服务器地址：smtp.live.com（端口：587）    gmail(google.com)  POP3服务器地址:pop.gmail.com（SSL启用端口：995）  SMTP服务器地址:smtp.gmail.com（SSL启用 端口：587）    263.net:  POP3服务器地址:pop3.263.net（端口：110）  SMTP服务器地址:smtp.263.net（端口：25）    263.net.cn:  POP3服务器地址:pop.263.net.cn（端口：110）  SMTP服务器地址:smtp.263.net.cn（端口：25）    x263.net:  POP3服务器地址:pop.x263.net（端口：110）  SMTP服务器地址:smtp.x263.net（端口：25）    21cn.com:  POP3服务器地址:pop.21cn.com（端口：110）  SMTP服务器地址:smtp.21cn.com（端口：25）    Foxmail：  POP3服务器地址:POP.foxmail.com（端口：110）  SMTP服务器地址:SMTP.foxmail.com（端口：25）    china.com:  POP3服务器地址:pop.china.com（端口：110）  SMTP服务器地址:smtp.china.com（端口：25）    tom.com:  POP3服务器地址:pop.tom.com（端口：110）  SMTP服务器地址:smtp.tom.com（端口：25）    etang.com:  POP3服务器地址:pop.etang.com  SMTP服务器地址:smtp.etang.com |

# 10、服务器使用阿里云ECS服务器，禁用了最后查询问题是因为阿里云屏蔽了非SSL链接的25发送邮件端口。  这里直接使用阿里云邮箱进行发送，并使用ssl验证

#阿里云服务器禁用了其他运营上的25端口，这里添加的为 阿里云的邮箱，并且设置为SSL验证  
mail.from=healerjean@aliyun.com  
mail.host=smtp.aliyun.com  
mail.password=lallalla  
mail.smtp.auth=true  
mail.smtp.timeout=25000  
mail.username= healerjean@aliyun.com  
mail.smtp.port=465  
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory

*<?*xml version="1.0" encoding="UTF-8"*?>*<beans  
 xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:p="http://www.springframework.org/schema/p"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">  
  
  
   
 <!-- 简单的发送者 -->  
 <bean id="mailMessage" class="org.springframework.mail.SimpleMailMessage">   
 <property name="from">   
 <value>${mail.from}</value>   
 </property>   
 <!-- 查看SimpleMailMessage源码还可以注入标题,内容等 -->   
 </bean>   
   
 <!-- 邮件发送器 163，qq，sohu -->  
 <!-- 申明JavaMailSenderImpl对象 -->   
 <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">   
 <property name="host" value="${mail.host}" />   
 <property name="username" value="${mail.username}" />   
 <property name="password" value="${mail.password}" />   
 <property name="defaultEncoding" value="UTF-8" />   
 <property name="protocol" value="smtp"/>   
 <property name="port" value="${mail.smtp.port}" /> <!-- 默认提供端口支持，默认是25 -->  
 <property name="javaMailProperties">  
 <props>  
 <!-- 启动调试开关 -->  
 <prop key="mail.debug">debug</prop>  
 <!-- 验证身份， 设置认证开关 ，默认false，这里设置为true-->   
 <prop key="mail.smtp.auth">${mail.smtp.auth}</prop>  
 <!-- 设置发送延时 -->  
 <prop key="mail.smtp.timeout">${mail.smtp.timeout}</prop>   
 <prop key="mail.smtp.socketFactory.class">${mail.smtp.socketFactory.class}</prop>  
 </props>  
 </property>  
 </bean>  
  
  
   
</beans>

具体看多点的代码，封装的很好哦

# 11、SpringBoot

// JavaMailSenderImpl javaMailSender = new JavaMailSenderImpl();  
// javaMailSender.setHost("smtp.163.com");  
// javaMailSender.setUsername("healerjean@163.com");  
// javaMailSender.setPassword("");  
// Properties properties = new Properties();  
// properties.put("mail.smtp.auth", true);  
// properties.put("mail.smtp.timeout", 25000);  
// javaMailSender.setJavaMailProperties(properties);