

## spring boot 邮件服务

作用

spring 自带Email

传统实现

邮件协议

SMTP 协议

POP3 协议

范例

# spring boot 邮件服务

## 作用

互联网发展到现在，想必大家都知道发送邮件应该是网站的必备功能之一:用户注册发送邮箱验证、忘记密码、监控提醒以及发送营销信息等。

## spring 自带Email

Spring Email抽象的核心是MailSender接口，MailSender的实现能够把Email发送给邮件服务器，由邮件服务器实现邮件发送的功能。



## 传统实现

早期发送邮件是通过Java自带的JavaMail类来发送邮件的，需要自己封装消息体；

后来Spring推出了JavaMailSender类大大简化了发送邮件的过程，JavaMailSender继承自MailSender，提供了更强大的邮件发送功能，可支持不同类型的邮件发送。再到现在的Spring Boot又对其进行封装从而出现了spring-boot-starter-mail，进一步优化和完善邮件发送功能。

## 邮件协议

发送邮件的本质是将一个人的信息传输给另外一个人，那么如何传输就需要商量好标准，这些标准就是协议。最初只有两个协议：

## SMTP 协议

SMTP 的全称是“Simple Mail Transfer Protocol”，即简单邮件传输协议。它是一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式。SMTP 协议属于 TCP/IP 协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。SMTP 服务器就是遵循 SMTP 协议的发送邮件服务器。

SMTP 认证，简单地说就是要求必须在提供了账户名和密码之后才可以登录 SMTP 服务器，这就使得那些垃圾邮件的散播者无可乘之机。增加 SMTP 认证的目的是为了使用户避免受到垃圾邮件的侵扰。

## POP3 协议

POP3是Post Office Protocol 3的简称，即邮局协议的第3个版本,它规定怎样将个人计算机连接到Internet的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准,POP3允许用户从服务器上把邮件存储到本地主机（即自己的计算机）上,同时删除保存在邮件服务器上的邮件，而POP3服务器则是遵循POP3协议的接收邮件服务器，用来接收电子邮件的。但目前的 POP3 邮件服务器大都可以“只下载邮件，服务器端并不删除”，也就是改进的 POP3 协议。

**SMTP 和 POP3 是最初的两个协议，随着邮件的不断发展后来又增加了两个协议：**

## IMAP 协议

IMAP全称是Internet Mail Access Protocol，即交互式邮件存取协议，它是跟 POP3类似邮件访问标准协议之一。不同的是，开启了IMAP后，您在电子邮件客户端收取的邮件仍然保留在服务器上，同时在客户端上的操作都会反馈到服务器上，如：删除邮件，标记已读等，服务器上的邮件也会做相应的动作。所以无论从浏览器登录邮箱或者客户端软件登录邮箱，看到的邮件以及状态都是一致的。

## Mime 协议

MIME的英文全称是"Multipurpose Internet Mail Extensions" 多功能Internet 邮件扩充服务，它是一种多用途网际邮件扩充协议，在1992年最早应用于电子邮件系统，但后来也应用到浏览器。MIME意为多目Internet邮件扩展，它设计的最初目的是为了在发送电子邮件时附加多媒体数据，让邮件客户程序能根据其类型进行处理。然而当它被 HTTP协议支持之后，它的意义就更为显著了。它使得HTTP传输的不仅是普通的文本，而变得丰富多彩。

## QQ邮箱步骤

QQ邮箱 -> 设置 -> 账户 -> POP3/SMTP服务:开启服务后会获得QQ的授权码。

## 范例

**pom.xml文件**

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
```

## 配置文件

```
spring:
  mail:
    host: smtp.qq.com
    username: 2734572804@qq.com
    password: hwnotfswuvzfdhab
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true
            require: true
```

## 创建邮件的实体类

```
@Data
public class Mail implements Serializable {
    private static final long serialVersionUID =
-2116367492649751914L;
    private String recipient; //邮件接收人
    private String subject; //邮件主题
    private String content; //邮件内容
}
```

## 文本发送邮件

```
public class MailUtil {

    @Value("${spring.mail.username}")
    private String MAIL_SENDER; //邮件发送者

    @Autowired
    private JavaMailSender javaMailSender;

    /**
     * 发送文本邮件
     * @param mail
     */
    public void sendSimpleMail(Mail mail){
        SimpleMailMessage mailMessage = new SimpleMailMessage();
        mailMessage.setFrom(MAIL_SENDER);
```

```

        mailMessage.setTo(mail.getRecipient());
        mailMessage.setSubject(mail.getSubject());
        mailMessage.setText(mail.getContent());
        javaMailSender.send(mailMessage);
    }
}

```

其中:

from: 即为邮件发送者, 一般设置在配置文件中

to: 邮件接收者, 此参数可以为数组, 同时发送多人

subject: 邮件主题

content: 邮件的主体

copyTo: 抄送人

## 文本邮件测试类

```

@RunWith(SpringRunner.class)
@SpringBootTest
public class DemoApplicationTests {
    @Autowired
    private MailUtil mailUtil;

    //接收人
    private static final String RECIPIENT = "2734572804@qq.com";

    /**
     * 发送文本邮件
     */
    @Test
    public void sendSimpleMail() {
        Mail mailBean = new Mail();
        mailBean.setRecipient(RECIPIENT);
        mailBean.setSubject("SpringBootMail之这是一封文本的邮件");
        mailBean.setContent("SpringBootMail发送一个简单格式的邮件, 时间为: " + DateUtil.format(new Date()));
        mailUtil.sendSimpleMail(mailBean);
    }
}

```

## HTML格式邮件发送

与文本格式邮件代码对比, 富文本HTML邮件发送使用**MimeMessageHelper**类, 把setText()方法的消息文本设置为html, 并将第二个参数设置为true, 表示这是html的富文本。**MimeMessageHelper**支持发送复杂邮件模板, 支持文本、附件、HTML、图片等。

```

public void sendHtmlmail(Mail mail){
    MimeMessage mimeMessage = null;
    try {
        mimeMessage= javaMailSender.createMimeMessage();
        //true 表示需要创建一个multipart message
        MimeMessageHelper mimeMessageHelper = new
MimeMessageHelper(mimeMessage, true);
        mimeMessageHelper.setFrom(MAIL_SENDER);
        mimeMessageHelper.setTo(mail.getRecipient());
        mimeMessageHelper.setSubject(mail.getSubject());
        //邮件抄送
        //mimeMessageHelper.addCc("抄送人");
        mimeMessageHelper.setText(mail.getContent(),true);
        javaMailSender.send(mimeMessage);
    }catch (Exception e){
        logger.error("邮件发送失败", e.getMessage());
    }
}

```

## HTML格式邮件测试类

```

@Test
public void sendHtmlMail(){
    Mail mail = new Mail();
    mail.setRecipient(RECIPIENT);
    mail.setSubject("SpringBootMailHTML之这是一封HTML格式的邮件");
    StringBuilder sb = new StringBuilder();
    sb.append("<h2>springboot测试html5</h2>")
        .append("<p style='text-align:left'>这是一封HTML邮件...")
        .append("<p> 时间为: " + DateUtil.format(new Date()) + ")")
        .append("</p>");
    mail.setContent(sb.toString());
    mailUtil.sendHtmlmail(mail);
}

```

## 附件格式邮件发送