14-邮件发送的整合与使用

1.基础协议及邮件配置整合

一、名词概念解释

- 什么是POP3、SMTP和IMAP?
  简单的说：POP3和IMAP是用来从服务器上下载邮件的。SMTP适用于发送或中转信件时找到下一个目的地。所以我们发送邮件应该使用SMTP协议。
  IMAP和POP3有什么区别?
- 什么是免费邮箱客户端授权码功能?
  邮箱客户端授权码是为了避免您的邮箱密码被盗后，盗号者通过客户端登录邮箱而独特设计的安防功能。可以理解为客户端授权码为邮件发送的二次密码。

二、 整合邮件发送功能

2.1. 引入依赖

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

2.2.邮箱配置

QQ系邮箱配置

官方配置说明：参考官方帮助中心

获取客户端授权码：参考官方帮助中心

详细的配置如下：

```yaml
spring:
  mail:
    host: smtp.qq.com #发送邮件服务器
    username: xx@qq.com #QQ邮箱
    password: xxxxxxxxxxx #客户端授权码
    protocol: smtp #发送邮件协议
    properties.mail.smtp.auth: true
    properties.mail.smtp.port: 465 #端口号465或587
    properties.mail.display.sendmail: Javen #可以任意
    properties.mail.display.sendname: Spring Boot Guide Email #可以任意
    properties.mail.smtp.starttls.enable: true
    properties.mail.smtp.starttls.required: true
    properties.mail.smtp.ssl.enable: true
    default-encoding: utf-8
```

> **说明：开启SSL时使用587端口时无法连接QQ邮件服务器**

网易系(126/163/yeah)邮箱配置

网易邮箱客户端授码:参考官方帮助中心

客户端端口配置说明:参考官方帮助中心

详细的配置如下：

```yaml
spring:
  mail:
    host: smtp.126.com
    username: xx@126.com
    password: xxxxxxxx
    protocol: smtp
    properties.mail.smtp.auth: true
    properties.mail.smtp.port: 994 #465或者994
    properties.mail.display.sendmail: Javen
    properties.mail.display.sendname: Spring Boot Guide Email
    properties.mail.smtp.starttls.enable: true
    properties.mail.smtp.starttls.required: true
    properties.mail.smtp.ssl.enable: true
    default-encoding: utf-8
    from: xx@126.com
```

特别说明:

- 126邮箱SMTP服务器地址:smtp.126.com,端口号:465或者994
- 163邮箱SMTP服务器地址:smtp.163.com,端口号:465或者994
- yeah邮箱SMTP服务器地址:smtp.yeah.net,端口号:465或者994

> **有的邮件服务器接受使用客户端授权码发邮件，有的邮件服务器接受使用邮箱密码来发送邮件，所以password的配置不能一概而论。客户端授权码不行，就试试用邮箱密码；邮箱密码不行，就试试客户端授权码。**

三、发送简单邮件

这里的简单邮件就是指邮件的内容只是普通文字的这种邮件。

```java
@Service
public class MailService {
    @Resource
    private JavaMailSender mailSender;

    @Value("${spring.mail.username}")
    private String fromEmail;

    /**
     * 发送文本邮件
     */
    public void sendSimpleMail(String to, String subject, String content)
{
        SimpleMailMessage message = new SimpleMailMessage();
        message.setFrom(fromEmail);
        message.setTo(to);
        message.setSubject(subject);
        message.setText(content);
        mailSender.send(message);
    }

}
```

sendSimpleMail的三个参数依次是：邮件的发送目标，邮件的标题，邮件的内容。

测试代码:

```java
@SpringBootTest(webEnvironment =
SpringBootTest.WebEnvironment.DEFINED_PORT)
public class MailServiceTest {

    @Resource
    MailService mailService;

    @Test
    public void sendSimpleMail() {
        mailService.sendSimpleMail("16422802@qq.com",
                "普通文本邮件",
                "普通文本邮件内容测试");
    }
}
```

附录：QQ邮箱发邮件设置

1.开启SMTP服务

2.在配置开启SMTP之后，会返回给我们一个客户端授权码。这个授权码就是上文中用来发邮件的password。

2.发送html和基于模板的邮件

一、发送html邮件服务

sendHtmlMail函数的第一个参数是发送目标邮箱，第二个参数是邮件标题，第三个参数是邮件的正文(html)。

- 上一节中发送普通的文本文件邮件，使用的是SimpleMailMessage
- 下文代码中发送的正文是HTML的邮件，使用的是MimeMessage

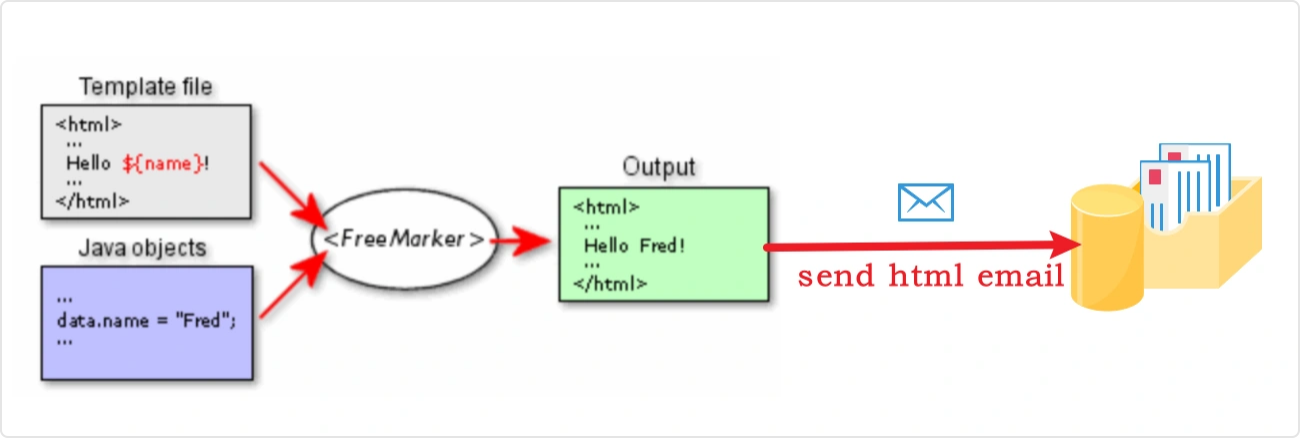
```java
/**
 * 发送html邮件
 */
public void sendHtmlMail(String to, String subject, String content)
throws MessagingException {
    //注意这里使用的是MimeMessage
    MimeMessage message = mailSender.createMimeMessage();
    MimeMessageHelper helper = new MimeMessageHelper(message, true);
    helper.setFrom(fromEmail);
    helper.setTo(to);
    helper.setSubject(subject);
    //第二个参数是否是html，true表示发送的邮件正文是html文本
    helper.setText(content, true);

    mailSender.send(message);
}
```

测试用例，我们将HTML以字符串拼接的方式写在Java代码里面，这样对于开发者而言不太友好。可以结合Java模板引擎，如Freemarker来来发送HTML邮件。

```java
@Test
public void sendHtmlMail() throws MessagingException {
    mailService.sendHtmlMail(
        "16422802@qq.com",
        "一封HTML测试邮件",
        """
        <body>
        <div style="width:600px;height:400px;margin:
auto;background:#9e98f9;color:#fff;text-align:center;">
        <h3>开到荼蘼</h3>
        <img src="https://avatars.githubusercontent.com/u/42235689?v=4">
        <p>
        <a style="text-decoration: none;color:#fff;"
href="https://github.com/mqxu" target="_bank" >
        <strong>我的Github主页</strong>
        </a>
        </p>
        </div>
        </body>""");
        }
```

二、基于freemarker模板的邮件



基于freemarker模板邮件本质上，还是发送html邮件，只不过是有一个把模板转换成html字符串的过程。

- 先添加freemarker依赖

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

- application.yml配置freemarker

```yaml
spring:
  freemarker:
  cache: false # 缓存配置 开发阶段应该配置为false 因为经常会改
  suffix: .ftl # 模版文件后缀名
  charset: UTF-8 # 文件编码
  template-loader-path: classpath:/templates/
```

- Article实体类

```java
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
public class Article {
    private Long id;
    private String author;
    private String title;
    private String content;
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss", timezone = "GMT+8")
    private Date createTime;
}
```

- 编写模板邮件发送测试代码

```java
@Test
public void sendTemplateMail() throws Exception{
    //添加动态数据，替换模板里面的占位符
    List<Article> articles = new ArrayList<>();
    articles.add(new Article(1L, "mqxu", "SpringBoot", "内容一", new Date()));
    articles.add(new Article(2L, "mqxu", "Vue.js", "内容二", new Date()));
    Template template = freeMarkerConfigurer.getConfiguration().getTemplate("freemarker-temp.ftl");
    //将模板文件及数据渲染完成之后，转换为html字符串
    Map<String, Object> model = new HashMap<>();
    model.put("articles", articles);
    String templateHtml = FreeMarkerTemplateUtils.processTemplateIntoString(template, model);
    //发送邮件
    mailService.sendHtmlMail("16422802@qq.com", "这是一封freemarker模板的html测试邮件", templateHtml);
}
```

- templates目录新建 freemarker-temp.ftl 模板文件

HTML | 复制代码

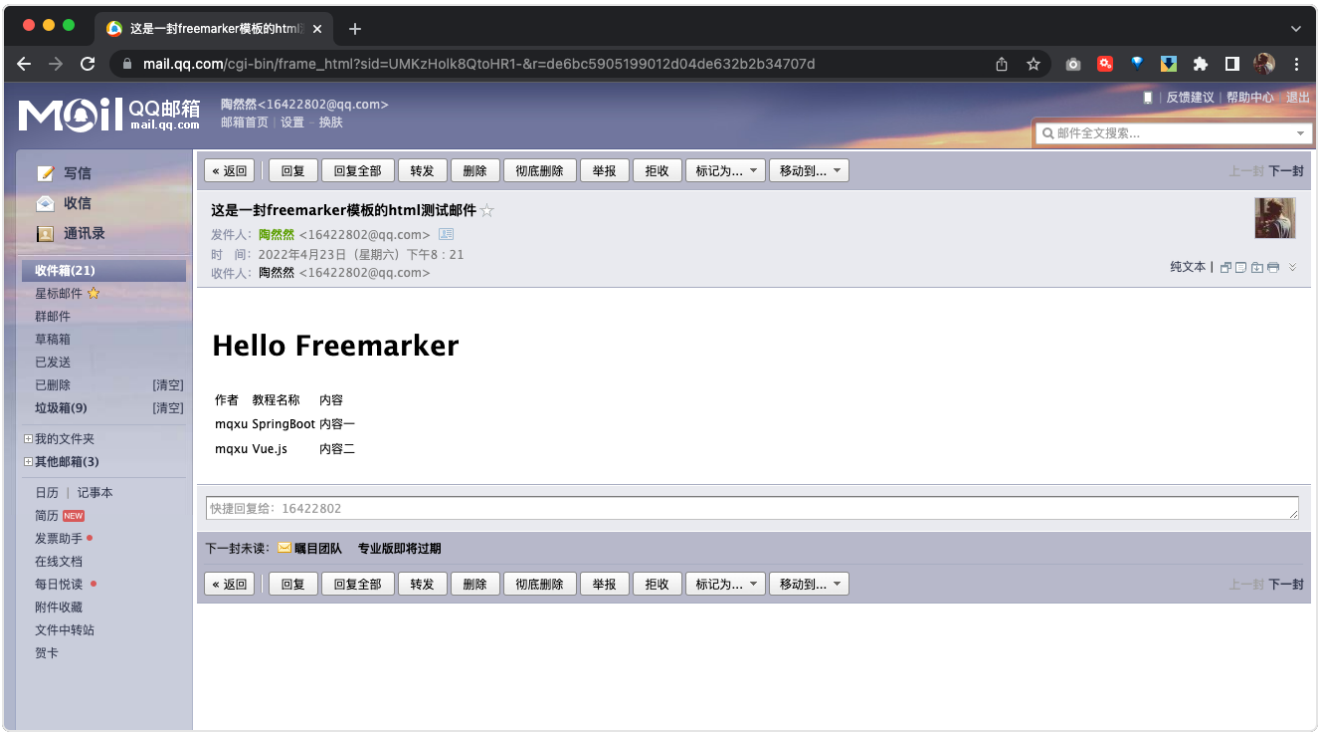
```html
<!DOCTYPE html>
<html lang="en">
<head lang="en">
    <meta charset="UTF-8"/>
    <title>freemarker简单示例</title>
</head>
<body>
<h1>Hello Freemarker</h1>

<table class="">
    <tr>
        <td>作者</td>
        <td>教程名称</td>
        <td>内容</td>
    </tr>

    <#list articles as article>
        <tr>
            <td>${article.author}</td>
            <td>${article.title}</td>
            <td>${article.content}</td>
        </tr>
    </#list>

</table>
</body>
</html>
```

- 测试结果：



3.发送带附件和内联附件邮件

一、发送带附件的邮件

```java
/**
 * 发送带附件的邮件
 */
public void sendAttachmentsMail(String to, String subject, String content, String filePath) throws MessagingException {
    MimeMessage message = mailSender.createMimeMessage();
    //带附件第二个参数true
    MimeMessageHelper helper = new MimeMessageHelper(message, true);
    helper.setFrom(fromEmail);
    helper.setTo(to);
    helper.setSubject(subject);
    helper.setText(content, true);
    //添加附件资源
    FileSystemResource file = new FileSystemResource(new File(filePath));
    String fileName = filePath.substring(filePath.lastIndexOf(File.separator));
    helper.addAttachment(fileName, file);
    //发送邮件
    mailSender.send(message);
}
```

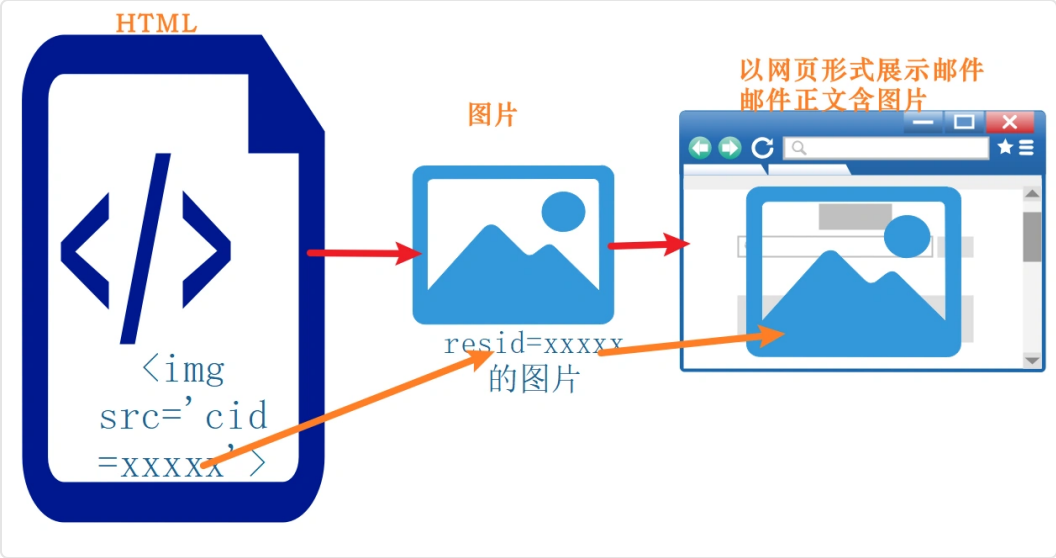
sendAttachmentsMail的第一个参数是发送目标邮箱，第二个参数是邮件的内容，第三个参数是邮件的附件。

运行如下的测试用例进行测试：

```java
@Test
    public void sendAttachmentsMailTest() throws MessagingException {
        String filePath = "/Users/mqxu/Desktop/1.png";
        mailService.sendAttachmentsMail("16422802@qq.com",
                                        "这是一封带附件的邮件——来自mqxu",
                                        "邮件中有附件，请注意查收！",
filePath);
    }
```

二、发送内联附件的邮件

所谓的内联附件就是附件文件在邮件正文内显示，通常是一图片资源。



Java | 复制代码

```java
/**
* 发送正文中有静态资源的邮件
*/
public void sendResourceMail(String to, String subject, String content,
String rscPath, String rscId) throws MessagingException {
    MimeMessage message = mailSender.createMimeMessage();

    MimeMessageHelper helper = new MimeMessageHelper(message, true);
    helper.setFrom(fromEmail);
    helper.setTo(to);
    helper.setSubject(subject);
    helper.setText(content, true);
    //添加内联附件，指定一个资源id:rscId
    FileSystemResource res = new FileSystemResource(new File(rscPath));
    helper.addInline(rscId, res);

    mailSender.send(message);
}
```

sendResourceMail方法的参数说明：

- 参数一：发送邮件的目标邮箱
- 参数二：文件的标题
- 参数三：邮件的正文：html（含图片资源id：rscId）
- 参数四：图片资源文件本地磁盘路径res
- 参数五：图片资源文件的资源Id：rscId

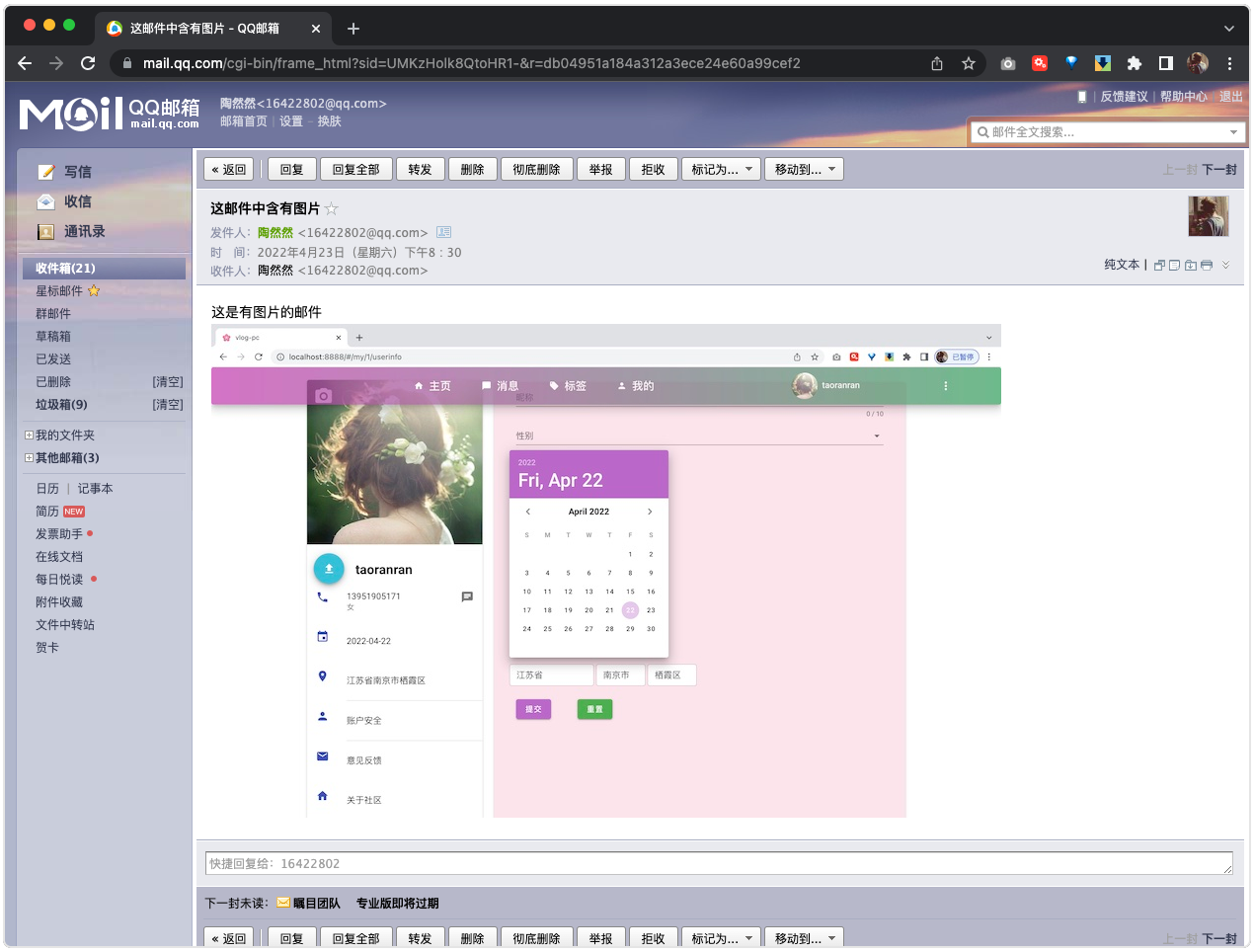
参数三HTML文本发现正文中包含<img src=cid: rscId>,就会根据参数五helper.addInline(rscId, res);，找到参数四对应的资源文件res，并渲染到HTML里面。

测试用例，执行之后看结果

Java | 复制代码

```java
@Test
public void sendResourceMail() throws MessagingException {
    String rscId = "mqxu";
    String content = "<html><body>这是有图片的邮件<br/><img src='cid:" +
rscId + "' ></body></html>";
    String imgPath = "/Users/mqxu/Desktop/1.jpg";
    mailService.sendResourceMail("16422802@qq.com",
                                 "这邮件中含有图片",
                                 content,
                                 imgPath,
                                 rscId);
}
```

测试结果


这邮件中含有图片 - QQ邮箱
mail.qq.com/cgi-bin/frame_html?sid=UMKzHolk8QtoHR1-&r=db04951a184a312a3ece24e60a99cef2
QQ邮箱
mail.qq.com
陶然然<16422802@qq.com>
邮箱首页 | 设置 | 换肤
反馈建议 | 帮助中心 | 退出
邮件全文搜索...
写信
收信
通讯录
收件箱(21)
星标邮件
群邮件
草稿箱
已发送
已删除 [清空]
垃圾箱(9) [清空]
我的文件夹
其他邮箱(3)
日历 | 记事本
简历 NEW
发票助手
在线文档
每日悦读
附件收藏
文件中转站
贺卡
« 返回
回复
回复全部
转发
删除
彻底删除
举报
拒收
标记为...
移动到...
上一封 下一封
这邮件中含有图片
发件人：陶然然 <16422802@qq.com>
时 间：2022年4月23日（星期六）下午8：30
收件人：陶然然 <16422802@qq.com>
纯文本
这是有图片的邮件
localhost:8888/#/my/1/userinfo
主页
消息
标签
我的
taoranran
性别
2022
Fri, Apr 22
April 2022
taoranran
13951905171
女
2022-04-22
江苏省南京市栖霞区
账户安全
意见反馈
关于社区
江苏省
南京市
栖霞区
提交
重置
快捷回复给：16422802
下一封未读：瞩目团队 专业版即将过期