

# Spring Boot系列（十三） Spring Boot 发送邮件

2017-11-06 Java知音

这一篇介绍Spring Boot 发送邮件需要springboot实战完整视频教程的，[点击这里](#)！

Spring框架使用JavaMailSender接口为发送邮件提供了一个简单的抽象，并且Spring Boot也为它提供了自动配置和一个starter模块。

如果spring.mail.host和相关的库（通过spring-boot-starter-mail定义）都存在，一个默认的JavaMailSender将被创建。该sender可以通过spring.mail命名空间下的配置项进一步自定义，下面本站素文宅博客具体讲述一下Spring Boot如何实现发送邮件。

引入spring-boot-starter-mail依赖，在pom.xml配置文件中增加如下内容（基于之前章节“Spring Boot 构建框架”中的pom.xml文件）：

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

## 应用发送邮件案例

在application.properties配置文件中加入如下配置（注意替换自己的用户名和密码）：

```
spring.mail.host=smtp.qq.com
spring.mail.username=用户名    //发送方的邮箱
spring.mail.password=密码    //对于qq邮箱而言 密码指的就是发送方的授权码
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.starttls.required=true
```

邮件service服务代码，具体如下：

```
@Service
public class MailService {
    private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @Autowired
    private JavaMailSender sender;

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

    /**
     * 发送纯文本的简单邮件
     * @param to
     * @param subject
     * @param content
     */
}
```

```
    */  
    public void sendSimpleMail(String to, String subject, String content) {  
        SimpleMailMessage message = new SimpleMailMessage();  
        message.setFrom(from);  
        message.setTo(to);  
        message.setSubject(subject);  
        message.setText(content);  
  
        try {  
            sender.send(message);  
            logger.info("简单邮件已经发送。");  
        } catch (Exception e) {  
            logger.error("发送简单邮件时发生异常!", e);  
        }  
    }  
  
    /**  
     * 发送html格式的邮件  
     * @param to  
     * @param subject  
     * @param content  
     */  
    public void sendHtmlMail(String to, String subject, String content) {  
        MimeMessage message = sender.createMimeMessage();  
  
        try {  
            //true表示需要创建一个multipart message  
            MimeMessageHelper helper = new MimeMessageHelper(message, true);  
            helper.setFrom(from);  
            helper.setTo(to);  
            helper.setSubject(subject);  
            helper.setText(content, true);  
  
            sender.send(message);  
            logger.info("html邮件已经发送。");  
        } catch (MessagingException e) {  
            logger.error("发送html邮件时发生异常!", e);  
        }  
    }  
  
    /**  
     * 发送带附件的邮件  
     * @param to  
     * @param subject  
     * @param content  
     * @param filePath  
     */  
    public void sendAttachmentsMail(String to, String subject, String content, String filePath) {  
        MimeMessage message = sender.createMimeMessage();
```

```

    try {
        //true表示需要创建一个multipart message
        MimeMessageHelper helper = new MimeMessageHelper(message, true);
        helper.setFrom(from);
        helper.setTo(to);
        helper.setSubject(subject);
        helper.setText(content, true);

        FileSystemResource file = new FileSystemResource(new File(filePath));
        String fileName = filePath.substring(filePath.lastIndexOf(File.separator));
        helper.addAttachment(fileName, file);

        sender.send(message);
        logger.info("带附件的邮件已经发送。");
    } catch (MessagingException e) {
        logger.error("发送带附件的邮件时发生异常!", e);
    }
}

/**
 * 发送嵌入静态资源（一般是图片）的邮件
 * @param to
 * @param subject
 * @param content 邮件内容， 需要包括一个静态资源的id， 比如: 
 * @param rscPath 静态资源路径和文件名
 * @param rscId 静态资源id
 */
public void sendInlineResourceMail(String to, String subject, String content, String rscPath,
    MimeMessage message = sender.createMimeMessage();

    try {
        //true表示需要创建一个multipart message
        MimeMessageHelper helper = new MimeMessageHelper(message, true);
        helper.setFrom(from);
        helper.setTo(to);
        helper.setSubject(subject);
        helper.setText(content, true);

        FileSystemResource res = new FileSystemResource(new File(rscPath));
        helper.addInline(rscId, res);

        sender.send(message);
        logger.info("嵌入静态资源的邮件已经发送。");
    } catch (MessagingException e) {
        logger.error("发送嵌入静态资源的邮件时发生异常!", e);
    }
}
}

```

简单测试代码如下：

```
public class MailTests extends BasicUtClass{  
    @Autowired  
    private MailService mailService;  
  
    private String to = "xujijun@mail.cn";  
  
    @Test  
    public void sendSimpleMail() {  
        mailService.sendSimpleMail(to, "主题：简单邮件", "测试邮件内容");  
    }  
}
```

来源：素文宅博客

链接：<https://blog.yoodb.com/yoodb/article/detail/1412>

#### 推荐阅读

[加入Java学习与交流群](#)

[Javaweb练手项目源码下载](#)

[精选Javaweb相关视频教程](#)

[100套IT类简历模板下载](#)

## Java知音


推送Java相关优质文章，热点咨询及视频资源



微信号: Java\_friends



长按识别二维码关注

 Java知音

[阅读原文](#)