

Способы отправки почты из приложения SpringBoot

Ервлева Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Ервлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассмотрен процесс разработки сервиса для отправки сообщений на почту. Разработка производится с использованием языка программирования Java в IDE IntelliJ Idea. Итоговым результатом является рабочее приложение с возможностью отправки сообщений.

Ключевые слова: Spring, Java, Mail

Ways to send mail from the Springboot application

Erovleva Regina Viktorovna

Sholom-Aleichem Priamursky State University

Student

Erovlev Pavel Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the process of developing a service for sending messages to the mail. Development is made using the Java programming language in IDE IntelliJidea. The final result is a working application with the possibility of sending messages.

Keywords: Spring, Java, Mail

В большинстве приложений есть сценарии, с помощью которых можно отправить почту, содержащую некоторую информацию, группе людей. Электронное письмо может быть обычным текстовым сообщением с вложением или может потребоваться отправить письмо в определенном формате.

Цель данной статьи – разработать сервис с возможностью отправки электронных писем.

А.А. Симаков описал разработку трассировки для JVM программ в использовании анализа ПО и обратного проектирования [1]. В своей работе

В.П. Великов, К.С. Добрева рассмотрели проблемы автоматизированной генерации ПО [2]. В своей статье А.А. Шейн описал процесс разработки программы, которая автоматически создает наборы классов в виде нативных объектов Java [3]. А.В. Колойдчук рассмотрела методы программирования кредитных рисков внедрения ИКТ с помощью Java, Pascal и Basic [4]. А.Н. Иванов описал процесс реализации безопасного веб-приложения на Java [5].

Прежде чем перейти к основной части, нужно настроить все необходимые данные в файле «application.properties» (рис.1).

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=sender_mail_id
spring.mail.password=password

spring.mail.toMailID=to_mail_id
spring.mail.fromMailID=from_mail_id
spring.mail.bccMailID=bcc_mail_id

spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.connectiontimeout=50000
spring.mail.properties.mail.smtp.timeout=50000
spring.mail.properties.mail.smtp.writetimeout=50000

spring.mail.properties.mail.smtp.starttls.enable=true
```

Рисунок 1 – application.properties

Для некоторых SMTP-серверов требуется подключение TLS, поэтому включаем его здесь.

Для отправки обычной почты с простыми текстовыми сообщениями можно использовать «SimpleMailMessage» вместе с «JavaMailSender» (рис.2).

```
1  import org.slf4j.Logger;
2  import org.slf4j.LoggerFactory;
3  import org.springframework.beans.factory.annotation.Autowired;
4  import org.springframework.beans.factory.annotation.Value;
5  import org.springframework.mail.SimpleMailMessage;
6  import org.springframework.mail.javamail.JavaMailSender;
7  import org.springframework.stereotype.Component;
8  import org.thymeleaf.spring5.SpringTemplateEngine;
9
10 @Component
11 public class SimpleMailSender {
12
13     Logger log = LoggerFactory.getLogger(EmailSender.class);
14
15     @Autowired
16     private JavaMailSender javaMailSender;
17
18     @Autowired
19     private SpringTemplateEngine templateEngine;
20
21     @Value("${spring.mail.toMailID}")
22     String toMailID;
23
24     @Value("${spring.mail.fromMailID}")
25     String fromMailID;
26
27     @Value("${spring.mail.password}")
28     String password;
29
30     @Value("${spring.mail.bccMailID}")
31     String bccMailID;
32
33     public void sendEmail() {
34         log.info("sending mail from: {} to: {}", fromMailID, toMailID);
35         SimpleMailMessage msg = new SimpleMailMessage();
36         msg.setFrom(fromMailID);
37         msg.setTo(toMailID);
38         msg.setSubject("Simple mail from Spring Boot Application");
39         String body = "Hi Raju,\n\n" +
40             "Please send the candidate details ASAP. " +
41             "\n\nThanks, " +
42             "\nLaxman\n";
43         msg.setText(body);
44         javaMailSender.send(msg);
45         log.info("sent mail successfully from: {} to: {}", fromMailID, toMailID);
46     }
47 }
```

Рисунок 2 – SimpleMailSender

Для отправки письма, содержащего вложение, необходимо иметь вложение в папке ресурсов (рис.3).

```
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.core.io.ClassPathResource;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.mail.javamail.MimeMessagePreparator;
import org.springframework.stereotype.Component;

import javax.mail.MessagingException;
import javax.mail.internet.MimeMessage;

@Component
public class SendMailWithAttachment {

    @Autowired
    private JavaMailSender javaMailSender;

    @Value("${spring.mail.toMailID}")
    String toMailID;

    @Value("${spring.mail.fromMailID}")
    String fromMailID;

    @Value("${spring.mail.password}")
    String password;

    @Value("${spring.mail.bccMailID}")
    String bccMailID;

    public void sendMailWithAttachment() throws MessagingException {
        log.info("sending mail from: {} to: {} bcc: {}", fromMailID, toMailID, bccMailID);
        ClassPathResource pic = new ClassPathResource("static/doggie.jpeg");
        MimeMessage msg = javaMailSender.createMimeMessage();
        MimeMessagePreparator messagePreparator = mimeMessage -> {
            MimeMessageHelper messageHelper = new MimeMessageHelper(mimeMessage, true);
            messageHelper.setFrom(fromMailID);
            String[] toMailList = {toMailID};
            messageHelper.setTo(toMailList);
            String[] bccMailList = {bccMailID};
            messageHelper.setBcc(bccMailList);
            messageHelper.setSubject("This is a mail with attachment");
            messageHelper.setText("Hi Raju," +
                "<br/> <br/> Please find the attachment for your reference. <br/><br/>" +
                "Thanks,<br/> Laxman", true);
        };
        javaMailSender.send(messagePreparator);
        log.info("sent mail with attachment from: {} to: {} bcc: {}", fromMailID, toMailID, bccMailID);
    }
}
```

Рисунок 3 – sendMailWithAttachment

Для отправки почты в predetermined формате можно использовать «Thymeleaf», чтобы указать формат почты. Этот тип требований обычно возникает, когда нужно отправить почту списку людей для определенных деловых целей (рис.4).

```
5 import org.springframework.core.io.ClassPathResource;
6 import org.springframework.mail.javamail.JavaMailSender;
7 import org.springframework.mail.javamail.MimeMessageHelper;
8 import org.springframework.stereotype.Component;
9 import org.thymeleaf.context.Context;
10 import org.thymeleaf.spring5.SpringTemplateEngine;
11
12 import javax.mail.MessagingException;
13 import javax.mail.internet.MimeMessage;
14 import java.nio.charset.StandardCharsets;
15
16 @Component
17 public class SendMailWithTemplate {
18
19     @Autowired
20     private JavaMailSender javaMailSender;
21
22     @Autowired
23     private SpringTemplateEngine templateEngine;
24
25     @Value("${spring.mail.toMailID}")
26     String toMailID;
27
28     @Value("${spring.mail.fromMailID}")
29     String fromMailID;
30
31     public void sendMailWithTemplate() throws MessagingException {
32         MimeMessage message = javaMailSender.createMimeMessage();
33         MimeMessageHelper helper = new MimeMessageHelper(message,
34             MimeMessageHelper.MULTIPART_MODE_MIXED_RELATED,
35             StandardCharsets.UTF_8.name());
36         helper.addAttachment("template-cat.png", new ClassPathResource("static/cat.jpeg"));
37         Context context = new Context();
38         context.setVariable("name", "Laxman");
39         context.setVariable("department", "Information Technology");
40
41         context.setVariable("catImg", "https://cdn.pixabay.com/photo/2014/11/30/14/11/cat-551554__3");
42
43         String html = templateEngine.process("user-data", context);
44         helper.setFrom(fromMailID);
45         helper.setTo(toMailID);
46         helper.setText(html, true);
47         helper.setSubject("User Details");
48         javaMailSender.send(message);
49     }
50 }
```

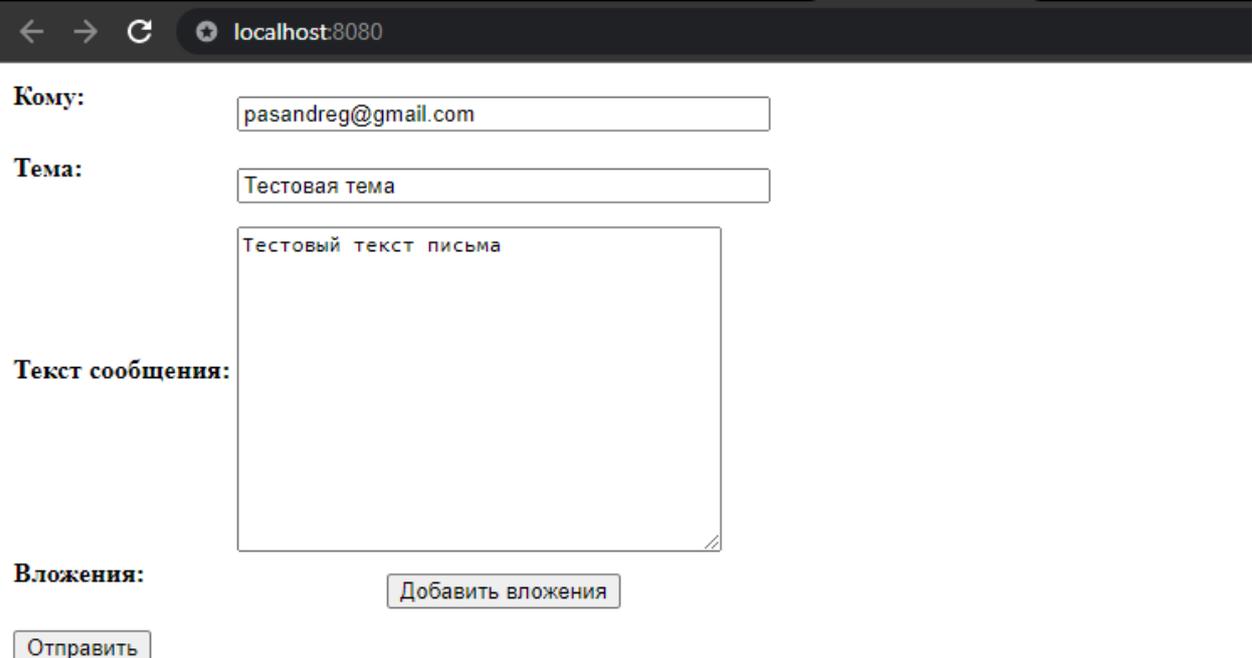
Рисунок 4 – sendMailWithTemplate

Для файла шаблона будем использовать обычный html файл (рис.5).

```
1 <html xmlns:th="https://thymeleaf.org">
2 <table>
3   <tr>
4     <td><h4>User Name: </h4></td>
5     <td><h4 th:text="${name}"></h4></td>
6   </tr>
7   <tr>
8     <td><h4>Department: </h4></td>
9     <td><h4 th:text="${department}"></h4></td>
10  </tr>
11  <tr>
12    <td><h4>Photo:</h4></td>
13    <td> </td>
14  </tr>
15 </table>
16 </html>
```

Рисунок 5 – Шаблон

Теперь при запуске приложения будет происходить перенаправление в окно браузера, где можно будет указать получателя, тема письма, написать текст, а также добавить вложение (рис.6).



← → ↻ localhost:8080

Кому:

Тема:

Текст сообщения:

Вложения:

Рисунок 6 – Рабочая программа в браузере

Из соображений безопасности может понадобиться прочитать почтовые идентификаторы и пароли из базы данных. Это можно сделать, создав классы «Entity» и вызвав методы интерфейса «JpaRepository».

В данной статье был рассмотрен процесс создания сервисов с возможностью отправки электронных писем.

Библиографический список

1. Симаков А.А. Java tracer. Программное средство для трассировки java программ // Заметки по информатике и математике. 2019. №3. С. 51-57.
2. Великов В.П., Добрева К.С. Генератор из диаграмм классов java в исходный код java // Информационные системы и технологии: управление и безопасность. 2014. №3. С. 14-23.
3. Шейн А.А. Генератор исходного кода на языке java по описанию бортовых компонентов decode // Современные проблемы науки и образования. 2016 №8. С. 18-25.
4. Колойдчук А.В. Программирование инвестиционных и кредитных рисков введение ИКТ с помощью pascal и visual basic // Формирование рыночных отношений в Украине. 2020 №4(227). С. 97-101.
5. Иванов А.Н. Разработка приложения с использованием веб-технологий java // Ученые заметки ТОГУ. 2020. №4. С. 261-266.