**《Spring Boot基础教程》**

**第24节**

**发送邮件-使用模板邮件并实现多账号轮询发送**

1. **添加依赖**

<!-- mail -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-mail</artifactId>

</dependency>

1. **配置**

# mail

spring.mail.host: smtp.exmail.qq.com

spring.mail.username:fengyw@roncoo.com,service@roncoo.com,education@roncoo.com

spring.mail.password:

spring.mail.properties.mail.smtp.auth: true

**# 企业qq的邮箱或者是163这类，不建议使用私人qq**

1. **代码实现**

**实现多账号**

/\*\*

\* 实现多账号，轮询发送

\*

\* @author wujing

\*/

@Configuration

@EnableConfigurationProperties(MailProperties.class)

public class RoncooJavaMailSenderImpl extends JavaMailSenderImpl implements JavaMailSender {

private ArrayList<String> usernameList;

private ArrayList<String> passwordList;

private int currentMailId = 0;

private final MailProperties properties;

public RoncooJavaMailSenderImpl(MailProperties properties) {

this.properties = properties;

// 初始化账号

if (usernameList == null)

usernameList = new ArrayList<String>();

String[] userNames = this.properties.getUsername().split(",");

if (userNames != null) {

for (String user : userNames) {

usernameList.add(user);

}

}

// 初始化密码

if (passwordList == null)

passwordList = new ArrayList<String>();

String[] passwords = this.properties.getPassword().split(",");

if (passwords != null) {

for (String pw : passwords) {

passwordList.add(pw);

}

}

}

@Override

protected void doSend(MimeMessage[] mimeMessage, Object[] object) throws MailException {

super.setUsername(usernameList.get(currentMailId));

super.setPassword(passwordList.get(currentMailId));

// 设置编码和各种参数

super.setHost(this.properties.getHost());

super.setDefaultEncoding(this.properties.getDefaultEncoding().name());

super.setJavaMailProperties(asProperties(this.properties.getProperties()));

super.doSend(mimeMessage, object);

// 轮询

currentMailId = (currentMailId + 1) % usernameList.size();

}

private Properties asProperties(Map<String, String> source) {

Properties properties = new Properties();

properties.putAll(source);

return properties;

}

@Override

public String getUsername() {

return usernameList.get(currentMailId);

}

}

**实现发送功能**

/\*\*

\*

\* **@author** wujing

\*/

@Component

**public** **class** RoncooJavaMailComponent {

**private** **static** **final** String ***template*** = "mail/roncoo.ftl";

@Autowired

**private** FreeMarkerConfigurer freeMarkerConfigurer;

@Autowired

**private** RoncooJavaMailSenderImpl javaMailSender;

**public** **void** sendMail(String email) {

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("email", email);

**try** {

String text = getTextByTemplate(***template***, map);

send(email, text);

} **catch** (IOException | TemplateException e) {

e.printStackTrace();

} **catch** (MessagingException e) {

e.printStackTrace();

}

}

**private** String getTextByTemplate(String template, Map<String, Object> model) **throws** TemplateNotFoundException, MalformedTemplateNameException, ParseException, IOException, TemplateException {

**return** FreeMarkerTemplateUtils.*processTemplateIntoString*(freeMarkerConfigurer.getConfiguration().getTemplate(template), model);

}

**private** String send(String email, String text) **throws** MessagingException, UnsupportedEncodingException {

MimeMessage message = javaMailSender.createMimeMessage();

MimeMessageHelper helper = **new** MimeMessageHelper(message, **true**, "UTF-8");

InternetAddress from = **new** InternetAddress();

from.setAddress(javaMailSender.getUsername());

from.setPersonal("龙果学院", "UTF-8");

helper.setFrom(from);

helper.setTo(email);

helper.setSubject("测试邮件");

helper.setText(text, **true**);

javaMailSender.send(message);

**return** text;

}

}

1. **测试**

**flt代码：**

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

</head>

<body>

<div style="width: 600px; text-align: left; margin: 0 auto;">

<h1 style="color: #005da7;">龙果学院</h1>

<div style="border-bottom: 5px solid #005da7; height: 2px; width: 100%;"></div>

<div style="border: 1px solid #005da7; font-size: 16px; line-height: 50px; padding: 20px;">

<div>${email}，您好！</div>

<div>

这是个测试

</div>

<div style="border-bottom: 2px solid #005da7; height: 2px; width: 100%;"></div>

<div>扫一扫，关注龙果学院微信公共号，里面更多精彩推荐</div>

<div>

<img src="http://account.roncoo.com/images/qrcode.png" alt="龙果学院公众号二维码" />

</div>

<div>

想了解更多信息，请访问 <a href="http://www.roncoo.com">http://www.roncoo.com</a>

</div>

</div>

</div>

</body>

</html>

**html代码：**

<input type=*"text"* name=*"email"* id=*"email"* />

<button id=*"send"*>发送邮件</button>

**js代码:**

$(**function**(){

$('#send').click(**function**(){

**var** email = $('#email').val();

$.ajax({

url:'/api/mail',

type:'post',

data:{'email':email},

success:**function**(msg){

alert(msg);

}

});

});

})

**java代码:**

@Autowired

**private** RoncooJavaMailComponent component;

@RequestMapping(value = "mail")

**public** String mail(String email) {

component.sendMail(email);

**return** "success";

}

**更多课程信息，请关注 龙果学院 官方网站**[**http://www.roncoo.com/**](http://www.roncoo.com/)

**或关注 龙果 微信公众号 RonCoo\_com**

****