[**第二十一篇： springboot集成JMS**](http://blog.csdn.net/forezp/article/details/71024024)

参考：<http://blog.csdn.net/forezp/article/details/71024024>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/71024024>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

springboot对JMS提供了很好的支持，对其做了起步依赖。

构架工程

创建一个springboot工程，在其pom文件加入：

<dependency>

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

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

</dependency>

添加配置

spring.mail.host=smtp.163.com

spring.mail.username=miles02@163.com

spring.mail.password=

spring.mail.port=25

spring.mail.protocol=smtp

spring.mail.default-encoding=UTF-8

在password 中填写自己的邮箱密码。

测试发邮件

测试代码清单如下：

package com.forezp;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.core.io.FileSystemResource;

import org.springframework.mail.SimpleMailMessage;

import org.springframework.mail.javamail.JavaMailSenderImpl;

import org.springframework.mail.javamail.MimeMessageHelper;

import org.springframework.test.context.junit4.SpringRunner;

import javax.mail.internet.MimeMessage;

import java.io.File;

@RunWith(SpringRunner.class)

@SpringBootTest

public class SpringbootJmsApplicationTests {

@Test

public void contextLoads() {

}

@Autowired

private JavaMailSenderImpl mailSender;

/\*\*

\* 发送包含简单文本的邮件

\*/

@Test

public void sendTxtMail() {

SimpleMailMessage simpleMailMessage = new SimpleMailMessage();

// 设置收件人，寄件人

simpleMailMessage.setTo(new String[] {"miles02@163.com"});

simpleMailMessage.setFrom("miles02@163.com");

simpleMailMessage.setSubject("Spring Boot Mail 邮件测试【文本】");

simpleMailMessage.setText("这里是一段简单文本。");

// 发送邮件

mailSender.send(simpleMailMessage);

System.out.println("邮件已发送");

}

/\*\*

\* 发送包含HTML文本的邮件

\* @throws Exception

\*/

@Test

public void sendHtmlMail() throws Exception {

MimeMessage mimeMessage = mailSender.createMimeMessage();

MimeMessageHelper mimeMessageHelper = new MimeMessageHelper(mimeMessage);

mimeMessageHelper.setTo("miles02@163.com");

mimeMessageHelper.setFrom("miles02@163.com");

mimeMessageHelper.setSubject("Spring Boot Mail 邮件测试【HTML】");

StringBuilder sb = new StringBuilder();

sb.append("<html><head></head>");

sb.append("<body><h1>spring 邮件测试</h1><p>hello!this is spring mail test。</p></body>");

sb.append("</html>");

// 启用html

mimeMessageHelper.setText(sb.toString(), true);

// 发送邮件

mailSender.send(mimeMessage);

System.out.println("邮件已发送");

}

/\*\*

\* 发送包含内嵌图片的邮件

\* @throws Exception

\*/

@Test

public void sendAttachedImageMail() throws Exception {

MimeMessage mimeMessage = mailSender.createMimeMessage();

// multipart模式

MimeMessageHelper mimeMessageHelper = new MimeMessageHelper(mimeMessage, true);

mimeMessageHelper.setTo("miles02@163.com");

mimeMessageHelper.setFrom("miles02@163.com");

mimeMessageHelper.setSubject("Spring Boot Mail 邮件测试【图片】");

StringBuilder sb = new StringBuilder();

sb.append("<html><head></head>");

sb.append("<body><h1>spring 邮件测试</h1><p>hello!this is spring mail test。</p>");

// cid为固定写法，imageId指定一个标识

sb.append("<img src=\"cid:imageId\"/></body>");

sb.append("</html>");

// 启用html

mimeMessageHelper.setText(sb.toString(), true);

// 设置imageId

FileSystemResource img = new FileSystemResource(new File("E:/1.jpg"));

mimeMessageHelper.addInline("imageId", img);

// 发送邮件

mailSender.send(mimeMessage);

System.out.println("邮件已发送");

}

/\*\*

\* 发送包含附件的邮件

\* @throws Exception

\*/

@Test

public void sendAttendedFileMail() throws Exception {

MimeMessage mimeMessage = mailSender.createMimeMessage();

// multipart模式

MimeMessageHelper mimeMessageHelper = new MimeMessageHelper(mimeMessage, true, "utf-8");

mimeMessageHelper.setTo("miles02@163.com");

mimeMessageHelper.setFrom("miles02@163.com");

mimeMessageHelper.setSubject("Spring Boot Mail 邮件测试【附件】");

StringBuilder sb = new StringBuilder();

sb.append("<html><head></head>");

sb.append("<body><h1>spring 邮件测试</h1><p>hello!this is spring mail test。</p></body>");

sb.append("</html>");

// 启用html

mimeMessageHelper.setText(sb.toString(), true);

// 设置附件

FileSystemResource img = new FileSystemResource(new File("E:/1.jpg"));

mimeMessageHelper.addAttachment("image.jpg", img);

// 发送邮件

mailSender.send(mimeMessage);

System.out.println("邮件已发送");

}

}

测试已全部通过，没有坑。

参考资料

<http://blog.720ui.com/2017/springboot_07_othercore_javamail/>

源码下载

<https://github.com/forezp/SpringBootLearning>

优秀文章推荐：

* 更多springboot 教程：[springBoot非官方教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70341818)
* 更多springcoud 教程：[史上最简单的 SpringCloud 教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70148833)