Date : 26/09/2020

Spring Boot 7AM

Mr. RAGHU

----------------------------------------------

**Spring Boot Email**

**(spring-boot-starter-mail)**

--Steps to send Email as a User(Human)--

1. Goto Mail Service Provider (host/port) ex: http://gmail.com

2. Login with Host (username/password)

3. Click on Compose to create new Message (MimeMessage)

4. Fill details in Message (to,subject,text,cc,bcc,attachments..)

5. Finally Send Operations

----------------------------------------

\*) JavaMailSenderImpl(C) is AutoConfigured by Spring boot that performs

Step#1 and Step#2. We need to pass properties only(host,port,un,pwd...)

\*) Define one class (EmailUtil) with one method (send())

that performs (step#3,4,5)

-> create empty message using JavaMailSender(I)

-> Fill details to empty message using MimeMessageHelper(C)

-> Send Message using JavaMailSender(I)

\*) Note: Read a File into Spring application

Resource r1 = new ClassPathResource("abcd.jpg"); //file inside src/main/resources folder

Resource r2 = new FileSystemResource("E:/images/abcd.jpg"); //file inside System Drive

Resource r3 = new FileUrlResource("http://imagescloud.net/myicon"); //file URL

Gmail:

<https://developers.google.com/gmail/imap/imap-smtp>

\*)Note:

a. Enable Less Secure Apps in Gmail Account

> Goto Gmail Account > Click on Top right corner symbol

> manage your google account

> Security

> Goto Less Secure Apps Access

> Turn on Less Secure Apps

> Click ON

b. Stop Antivirus for few minutes

Q) What is difference between CC and BCC?

A)

Actual Mail Receiver : To

Additional Users : CC

Additional Blind Users: BCC

If a User Email inside BCC, they can see other BCC users only.

Not a User who is in CC.

--Code Samples-------------------

//1. Create Empty Message by using JavaMailSender

MimeMessage message = sender.createMimeMessage();

//2. Use Helper class object -> message, isAttachmentExist?then true/false

MimeMessageHelper helper = new MimeMessageHelper(message, file!=null?true:false);

//set details

helper.setTo(to);

helper.setSubject(subject);

helper.setText(text);

if(cc!=null && cc.length>0)

helper.setCc(cc);

if(bcc!=null && bcc.length>0)

helper.setBcc(bcc);

if(file!=null)

helper.addAttachment(file.getFilename(), file);

//3. send message

sender.send(message);

-----------------------------------

\*)Note:

setText(String text) => This method will send data as Plain text by default

setText(String text, boolean html) : This method will send data as

HTML format if 2nd param is true, else plain text if 2nd param is false.

=======FULL CODE====================================

1. Create one Starter Project

Name : SpringBoot2EmailApp

Dep : java mail sender

2. application.properties

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.username=javaraghu2018

spring.mail.password=0044javaraghu2200

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

3. EmailUtil.java

package in.nareshit.raghu.util;

import javax.mail.internet.MimeMessage;

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

import org.springframework.core.io.Resource;

import org.springframework.mail.javamail.JavaMailSender;

import org.springframework.mail.javamail.MimeMessageHelper;

import org.springframework.stereotype.Component;

@Component

public class EmailUtil {

@Autowired

private JavaMailSender sender; // HAS-A

public boolean send(

String to,

String cc[],

String bcc[],

String subject,

String text,

Resource file

)

{

boolean flag = false;

try {

//1. Create Empty Message

MimeMessage message = sender.createMimeMessage();

//2. Use Helper class object -> message, isAttachmentExist?

MimeMessageHelper helper = new MimeMessageHelper(message, file!=null?true:false);

//set details

helper.setTo(to);

helper.setSubject(subject);

//helper.setText(text);//false-send as text

helper.setText(text,true); //send as HTML data

if(cc!=null && cc.length>0)

helper.setCc(cc);

if(bcc!=null && bcc.length>0)

helper.setBcc(bcc);

if(file!=null)

helper.addAttachment(file.getFilename(), file);

//3. send message

sender.send(message);

flag = true; //success

} catch (Exception e) {

flag = false; //failed

e.printStackTrace();

}

return flag;

}

}

4. Runner class

package in.nareshit.raghu.runner;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.core.io.FileSystemResource;

import org.springframework.stereotype.Component;

import in.nareshit.raghu.util.EmailUtil;

@Component

public class TestEmailRunner implements CommandLineRunner {

@Autowired

private EmailUtil emailUtil;

@Override

public void run(String... args) throws Exception {

FileSystemResource file = new FileSystemResource("F:\\Images\\SpringBoot7AM\_25092020.png");

boolean flag = emailUtil.send(

"abcdef@gmail.com",

new String[] {

"dollygour612@gmail.com",

"swaroopkumarbh@gmail.com",

"hemantpundir72@gmail.com",

},

new String[] {

"santhugmcastar@gmail.com",

"gantyalau@gmail.com",

"swaruphippergekar@gmail.com"

},

"SAMPLE DATA ONE",

"<html><body><h1>Hello Data ABC!</h1>"

+"<img src='https://s3-ap-southeast-1.amazonaws.com/tv-prod/member/photo/745494-medium190ap.jpg'/> "

+ "</body></html>",

file);

if(flag) {

System.out.println("SUCCESS");

} else {

System.out.println("FAIL");

}

}

}