前提：

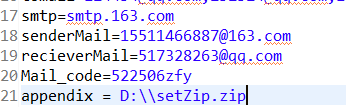
1. reportNG配置完成,所需包完成导入，
2. 参考网址：<https://www.cnblogs.com/chengtch/p/6071322.html>。
3. javamail所需包完成导入。

思路：

在testNG下完成一些测试用例的执行，reportNG在指定目录生成测试报告（一系列web文件），利用压缩包类将生成的测试报告（一系列web文件）所在的文件夹打包，将打包文件作为附件利用javamail完成发送。

## 源代码1（包含附件的sendMail类）

相应内容在config文件里添加



**import** java.util.Properties;

**import** javax.activation.DataHandler;

**import** javax.activation.DataSource;

**import** javax.activation.FileDataSource;

**import** javax.mail.Address;

**import** javax.mail.BodyPart;

**import** javax.mail.Multipart;

**import** javax.mail.Session;

**import** javax.mail.Transport;

**import** javax.mail.internet.InternetAddress;

**import** javax.mail.internet.MimeBodyPart;

**import** javax.mail.internet.MimeMessage;

**import** javax.mail.internet.MimeMessage.RecipientType;

**import** javax.mail.internet.MimeMultipart;

**import** com.webtest.utils.ReadProperties;

**public** **class** sendMail {

**private** String smtp;

**private** String sender;

**private** String reciever;

**private** String code;

**private** String title;

**private** String NGappendix;

**private** String body;

**public** sendMail() **throws** Exception{

//均从config.properties根据value值读取相应内容

smtp = ReadProperties.*getPropertyValue*("smtp");//

sender = ReadProperties.*getPropertyValue*("senderMail");//发送者邮箱地址

reciever = ReadProperties.*getPropertyValue*("recieverMail");//接收者邮箱地址

code = ReadProperties.*getPropertyValue*("Mail\_code"); //发送者邮件授权码

NGappendix = ReadProperties.*getPropertyValue*("appendix");//附件路径

}

**public** **void** setTitle\_body(String title,String body) {//设置邮件的标题和正文

**this**.title = title;

**this**.body = body;

}

**public** **void** Mail() **throws** Exception{

Properties props = **new** Properties();

props.put("mail.smtp.host", smtp);//指定邮件的发送服务器地址

props.put("mail.smtp.auth", "true");//服务器是否要验证用户的身份信息

Session session = Session.*getInstance*(props);//得到Session

session.setDebug(**true**);//代表启用debug模式，可以在控制台输出smtp协议应答的过程

//创建一个MimeMessage格式的邮件

MimeMessage message = **new** MimeMessage(session);

//设置发送者

Address fromAddress = **new** InternetAddress(sender);//邮件地址

message.setFrom(fromAddress);//设置发送的邮件地址

//设置接收者

Address toAddress = **new** InternetAddress(reciever);//要接收邮件的邮箱

message.setRecipient(RecipientType.***TO***, toAddress);//设置接收者的地址

//设置邮件的主题

message.setSubject(title);

/\* //设置邮件的内容

message.setText("practice trianing");\*/

BodyPart messageBodyPart = **new** MimeBodyPart();

messageBodyPart.setText(body);

Multipart multipart = **new** MimeMultipart();

multipart.addBodyPart(messageBodyPart);

//附件

messageBodyPart = **new** MimeBodyPart();

String filename = NGappendix;//附件路径

DataSource source = **new** FileDataSource(filename);

messageBodyPart.setDataHandler(**new** DataHandler(source));

messageBodyPart.setFileName(filename);

multipart.addBodyPart(messageBodyPart);

message.setContent(multipart);

//保存邮件

message.saveChanges();

//得到发送邮件的服务器(这里用的是smtp服务器)

Transport transport = session.getTransport("smtp");

//发送者的账号连接到smtp服务器上 @163.com可以不写

transport.connect(smtp,sender,code);

//发送信息

transport.sendMessage(message, message.getAllRecipients());

//关闭服务器通道

transport.close();

}

}

## 源代码2压缩指定文件夹类

**import** java.io.BufferedInputStream;

**import** java.io.BufferedOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.util.zip.ZipEntry;

**import** java.util.zip.ZipOutputStream;

**public** **class** buildZip {

**public** buildZip() {

//构造函数

}

/\*\*

\* 将存放在sourceFilePath目录下的源文件，打包成fileName名称的zip文件，并存放到zipFilePath路径下

\* **@param** sourceFilePath :待压缩的文件路径

\* **@param** zipFilePath :压缩后存放路径

\* **@param** fileName :压缩后文件的名称

\* **@return**

\*/

**public** **static** **boolean** fileToZip(String sourceFilePath,String zipFilePath,String fileName){

**boolean** flag = **false**;

File sourceFile = **new** File(sourceFilePath);

FileInputStream fis = **null**;

BufferedInputStream bis = **null**;

FileOutputStream fos = **null**;

ZipOutputStream zos = **null**;

**if**(sourceFile.exists() == **false**){

System.***out***.println("待压缩的文件目录："+sourceFilePath+"不存在.");

sourceFile.mkdir(); // 新建目录

}

**try** {

File zipFile = **new** File(zipFilePath + "/" + fileName +".zip");

**if**(zipFile.exists()){

System.***out***.println(zipFilePath + "目录下存在名字为:" + fileName +".zip" +"打包文件.");

}**else**{

File[] sourceFiles = sourceFile.listFiles();

**if**(**null** == sourceFiles || sourceFiles.length<1){

System.***out***.println("待压缩的文件目录：" + sourceFilePath + "里面不存在文件，无需压缩.");

}**else**{

fos = **new** FileOutputStream(zipFile);

zos = **new** ZipOutputStream(**new** BufferedOutputStream(fos));

**byte**[] bufs = **new** **byte**[1024\*10];

**for**(**int** i=0;i<sourceFiles.length;i++){

//创建ZIP实体，并添加进压缩包

ZipEntry zipEntry = **new** ZipEntry(sourceFiles[i].getName());

zos.putNextEntry(zipEntry);

//读取待压缩的文件并写进压缩包里

fis = **new** FileInputStream(sourceFiles[i]);

bis = **new** BufferedInputStream(fis, 1024\*10);

**int** read = 0;

**while**((read=bis.read(bufs, 0, 1024\*10)) != -1){

zos.write(bufs,0,read);

}

}

flag = **true**;

}

}

} **catch** (FileNotFoundException e) {

e.printStackTrace();

**throw** **new** RuntimeException(e);

} **catch** (IOException e) {

e.printStackTrace();

**throw** **new** RuntimeException(e);

} **finally**{

//关闭流

**try** {

**if**(**null** != bis) bis.close();

**if**(**null** != zos) zos.close();

} **catch** (IOException e) {

e.printStackTrace();

**throw** **new** RuntimeException(e);

}

}

**return** flag;

}

}

## 源代码3执行打包文件作为附件发送邮件的main方法

**public** **class** reportNG\_buildZip\_sendMail {

**public** **static** **void** main(String[] args) **throws** Exception{

//压缩指定文件对象

String file = "D:\\g3softwaretesting\\practical\_training\\test-output\\html";//也可以是文件夹路径

//待生成的zip包名

String zipName = "setZip";

//待生成的zip保存路径

String zipFilePath = "D:\\";

//压缩

buildZip.*fileToZip*(file , zipFilePath , zipName);

//将打包文件作为附件发送邮件。

sendMail s1 = **new** sendMail();

s1.setTitle\_body("标题","正文");

s1.Mail();

}

}