

## MỤC TIÊU

- ✓ Nắm vững kỹ thuật gửi email qua gmail
- ✓ Biết cách queue email để tránh nghẽn

### BÀI 1: Xây dựng thư viện gửi email thông qua GMail

Để thuận tiện cho việc gửi mail sau này, bạn cần xây dựng cho mình một thư viện gửi email sao cho thuận tiện và dễ dùng nhất, hiệu quả nhất.

Trong bài này phần thư viện bạn cần tạo 2 lớp là

- ✓ JMailSenderService: đây là thread chạy phía hậu trường để gửi email một cách tuần tự tránh tranh chấp tài nguyên bộ nhớ và CPU khi có nhiều hoạt động gửi email.
- ✓ JMailSender: chứa các phương thức gửi email từ đơn giản đến phức tạp

Mã nguồn của lớp JmailSenderService

```
package com.poly.util;

import java.util.ArrayList;
import java.util.List;
import javax.mail.MessagingException;
import javax.mail.Transport;
import javax.mail.internet.MimeMessage;

public class JMailSenderService extends Thread{
    static {
        JMailSenderService sender = new JMailSenderService();
        sender.start();
    }
    static final List<MimeMessage> queue = new ArrayList<>();

    public static void queue(MimeMessage mail){
        synchronized(queue){
            queue.add(mail);
            queue.notify();
        }
    }

    @Override
    public void run() {
        while(true){
            try {
```

```
synchronized(queue){
    if(queue.size() > 0){
        try {
            MimeMessage mail = queue.remove(0);
            Transport.send(mail);
            System.out.println("The mail was sent.");
        }
        catch (MessagingException e) {
            System.out.println("Unable to send mail.");
        }
    }
    else{
        queue.wait();
    }
}
catch (InterruptedException e) {
    break;
}
}
```

Mã nguồn của lớp JMailSender

```
package com.poly.util;

import java.io.File;
import java.util.Date;
import java.util.Properties;
import javax.activation.DataHandler;
import javax.activation.FileDataSource;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.PasswordAuthentication;
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.MimeMultipart;

public class JMailSender
{
```

```

private static final String email = "user@gmail.com";
private static final String password = "songlong";
private static final Properties config = new Properties();
static{
    // Các thông số gmail
    config.setProperty("mail.smtp.host", "smtp.gmail.com");
    config.setProperty("mail.smtp.port", "465");
    config.setProperty("mail.smtp.starttls.enable","true");
    config.setProperty("mail.smtp.auth", "true");
    config.setProperty("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    config.setProperty("mail.smtp.socketFactory.fallback", "false");
}

public static Session getSession() {
    // Đăng nhập gmail
    Authenticator authenticator = new Authenticator(){
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(email, password);
        }
    };
    Session session = Session.getInstance(config, authenticator);
    return session;
}

/**
 * Gửi email từ hệ thống
 *
 * @param to danh sách email của người nhận, cách nhau bởi dấu phẩy hoặc chấm phẩy.
 * @param subject tiêu đề của email
 * @param body nội dung email
 * @exception RuntimeException lỗi tạo email
 */
public static void send(String to, String subject, String body)
{
    String from = String.format("EStore Web Master <%s>", email);
    JMailSender.send(from, to, subject, body);
}

/**
 * Gửi email với thông tin đơn giản
 *
 * @param from email của người gửi (ví dụ: <b>Nguyen Nghiem <nnghiem@yahoo.com>&lt;/b>
 hoặc <b>nnghiem@yahoo.com</b>)
 * @param to danh sách email của người nhận, cách nhau bởi dấu phẩy hoặc chấm phẩy.
 * @param subject tiêu đề của email
 * @param body nội dung email
 * @exception RuntimeException lỗi tạo email

```

```

*/
public static void send(String from, String to, String subject, String body)
{
    String cc = "", bcc = "", attach = "";
    JMailSender.send(from, to, cc, bcc, subject, body, attach);
}

/**
 * Gửi email đầy đủ thông tin
 *
 * @param from email của người gửi (ví dụ: <b>Nguyen Nghiem <ngnghiem@yahoo.com></b>
 hoặc <b>ngnghiem@yahoo.com</b>)
 * @param to danh sách email của người nhận, cách nhau bởi dấu phẩy hoặc chấm phẩy.
 * @param cc danh sách email của những người đồng nhận, cách nhau bởi dấu phẩy hoặc chấm phẩy.
 * @param bcc giống cc nhưng các email này không hiển thị trên mail.
 * @param subject tiêu đề? của email
 * @param body nội dung email
 * @param attachments danh sách đường dẫn các tập tin đính kèm, cách nhau bởi dấu phẩy hoặc
 chấm phẩy.
 * @exception RuntimeException lỗi tạo email
 */
public static void send(String from, String to, String cc, String bcc, String subject, String body, String
attachments)
{
    try{
        MimeMessage mail = new MimeMessage(getSession());

        String[] addresses = from.split("[<>]");
        String name = addresses[0].trim();
        String email = (addresses.length > 1 ? addresses[1] : addresses[0]).trim();
        InternetAddress fromAddress = new InternetAddress(email, name, "utf8");

        mail.setFrom(fromAddress);
        mail.setReplyTo(new InternetAddress[]{fromAddress});

        final String toEmails = to.trim().replaceAll("[,;\\s]+", ",");
        mail.addRecipients(Message.RecipientType.TO, toEmails);

        if(cc != null && cc.trim().length() > 0)
        {
            final String ccEmails = cc.trim().replaceAll("[,;\\s]+", ",");
            mail.addRecipients(Message.RecipientType.CC, ccEmails);
        }

        if(bcc != null && bcc.trim().length() > 0)
        {
            final String bccEmails = bcc.trim().replaceAll("[,;\\s]+", ",");
            mail.addRecipients(Message.RecipientType.BCC, bccEmails);
        }
    }
}

```

```

    }
    mail.setSubject(subject, "utf-8");
    mail.setContent(body, "text/html; charset=utf-8");
    mail.setSentDate(new Date());

    if(attachments != null && attachments.trim().length() > 0)
    {
        MimeMultipart multiPart = new MimeMultipart();

        MimeBodyPart mailBodyPart = new MimeBodyPart();
        mailBodyPart.setContent(body, "text/html; charset=utf8");
        multiPart.addBodyPart(mailBodyPart);

        String[] paths = attachments.split("[,;]+");
        for(String path : paths)
        {
            File file = new File(path.trim());
            MimeBodyPart attachBodyPart = new MimeBodyPart();
            FileDataSource fds = new FileDataSource(file);
            attachBodyPart.setDataHandler(new DataHandler(fds));
            attachBodyPart.setFileName(file.getName());
            multiPart.addBodyPart(attachBodyPart);
        }
        mail.setContent(multiPart);
    }

    //Transport.send(mail);
    JMailSenderService.queue(mail);
    }
    catch (Exception e) {
        throw new RuntimeException(e);
    }
}
}

```

Với 2 lớp đã viết ở trên bây giờ bạn đã có thể chạy thử một chương trình gửi email đơn giản như sau

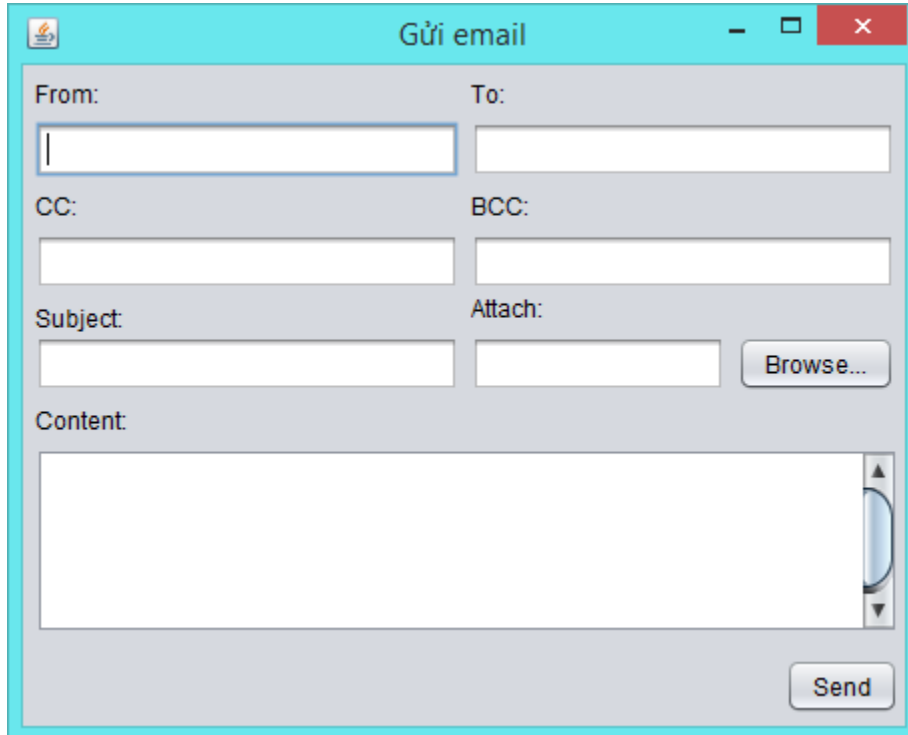
```

String from = "sender@gmail.com";
String to = "a@gmail.com ";
String subject = "Tiêu đề mail";
String body = "Nội dung mail";
JmailSender.send(from, to, subject, body);

```

## **BÀI 2:** Gửi email đầu đủ

Sử dụng thư viện đã xây dựng trong bài 1 hãy viết ứng dụng gửi email có giao diện như sau



#### HƯỚNG DẪN THỰC HIỆN

- ✓ Chú ý 2 thư viện gửi email là: mail.jar và activation.jar
- ✓ [Send]

```
String from = txtFrom.getText();  
String to = txtTo.getText();  
String cc = txtCC.getText();  
String bcc = txtBCC.getText();  
String subject = txtSubject.getText();  
String body = txtContent.getText();  
String attach = txtAttach.getText();
```

```
JMailSender.send(from, to, cc, bcc, subject, body, attach);
```

#### BAI 3: Giáo viên cho thêm