

Networking



■ Bài 8:

Nội dung bài học

- Download tài nguyên mạng
- Gửi email thông qua gmail



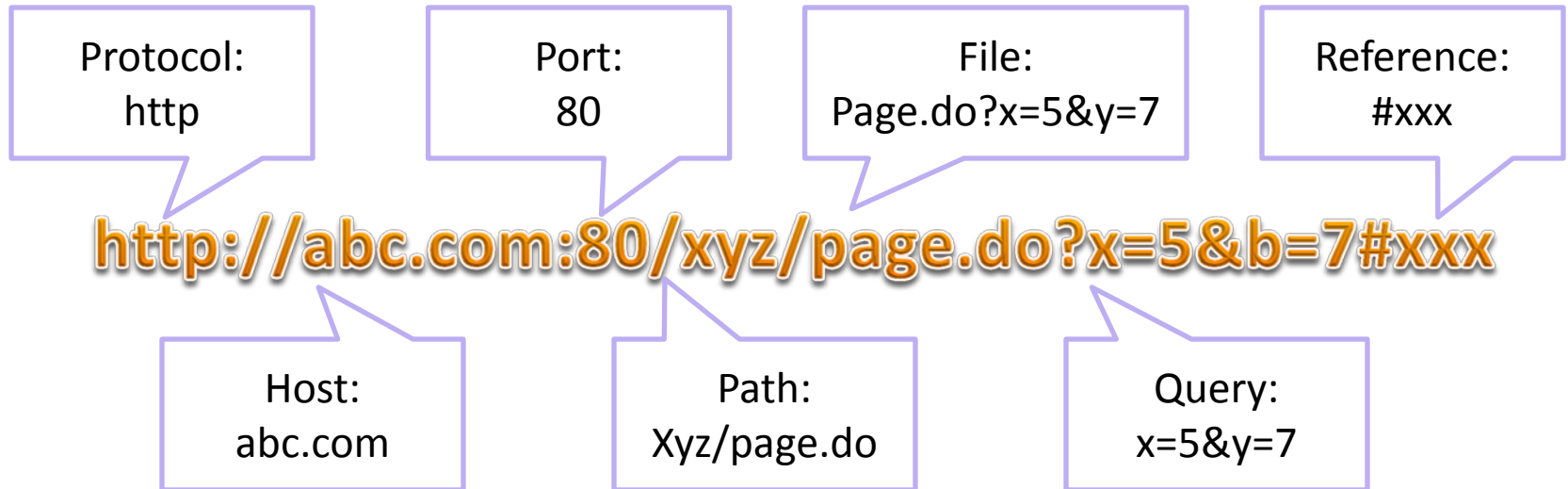
Tài nguyên mạng

- Tài nguyên trên mạng (trang web, hình ảnh, video...) có thể truy cập thông qua địa chỉ URL



Lớp URL

- Java cung cấp lớp `java.net.URL` để có thể gửi/nhận dữ liệu với các URL đó.



URL API

Kiểu trả về	Phương thức	Mô tả
String	<code>getFile()</code>	Lấy tên file
String	<code>getHost()</code>	Lấy tên server
String	<code>getPath()</code>	Lấy đường dẫn
int	<code>getPort()</code>	Lấy cổng
String	<code>getProtocol()</code>	Lấy giao thức
String	<code>getQuery()</code>	Lấy chuỗi truy vấn
String	<code>getRef()</code>	Lấy tham chiếu trong trang
URLConnection	<code>openConnection()</code>	Mở kết nối đến URL
InputStream	<code>openStream()</code>	Lấy luồng dữ liệu từ tài nguyên

Thông tin URL

```
try {  
    URL url = new URL("http://abc.com:80/xyz/page.jsp?x=5&y=7#xxx");  
    System.out.println("Protocol: " + url.getProtocol());  
    System.out.println("Host: " + url.getHost());  
    System.out.println("Port: " + url.getPort());  
    System.out.println("Path: " + url.getPath());  
    System.out.println("File: " + url.getFile());  
    System.out.println("Query: " + url.getQuery());  
    System.out.println("Ref: " + url.getRef());  
}  
catch (MalformedURLException ex) {  
    ex.printStackTrace();  
}
```

Protocol: http
Host: abc.com
Port: 80
Path: /xyz/page.jsp
File: /xyz/page.jsp?x=5&y=7
Query: x=5&y=7
Ref: xxx

Download từ Internet

```
ByteArrayOutputStream buffer = new ByteArrayOutputStream();
```

```
URL url = new URL("http://www.vnexpress.net");
```

Chứa dữ liệu download

```
InputStream is = url.openStream();
```

```
byte[] block = new byte[4*1024]; // 4KB
```

Mở luồng dữ liệu đến tài nguyên

```
while(true){
```

```
    int n = is.read(block);
```

Chỉ nên đọc 1 lần tối đa 4KB

```
    if(n <= 0){
```

```
        break;
```

Dừng khi đã hết dữ liệu

```
    }
```

```
        buffer.write(block, 0, n);
```

Tích lũy dữ liệu đọc được

```
    }
```

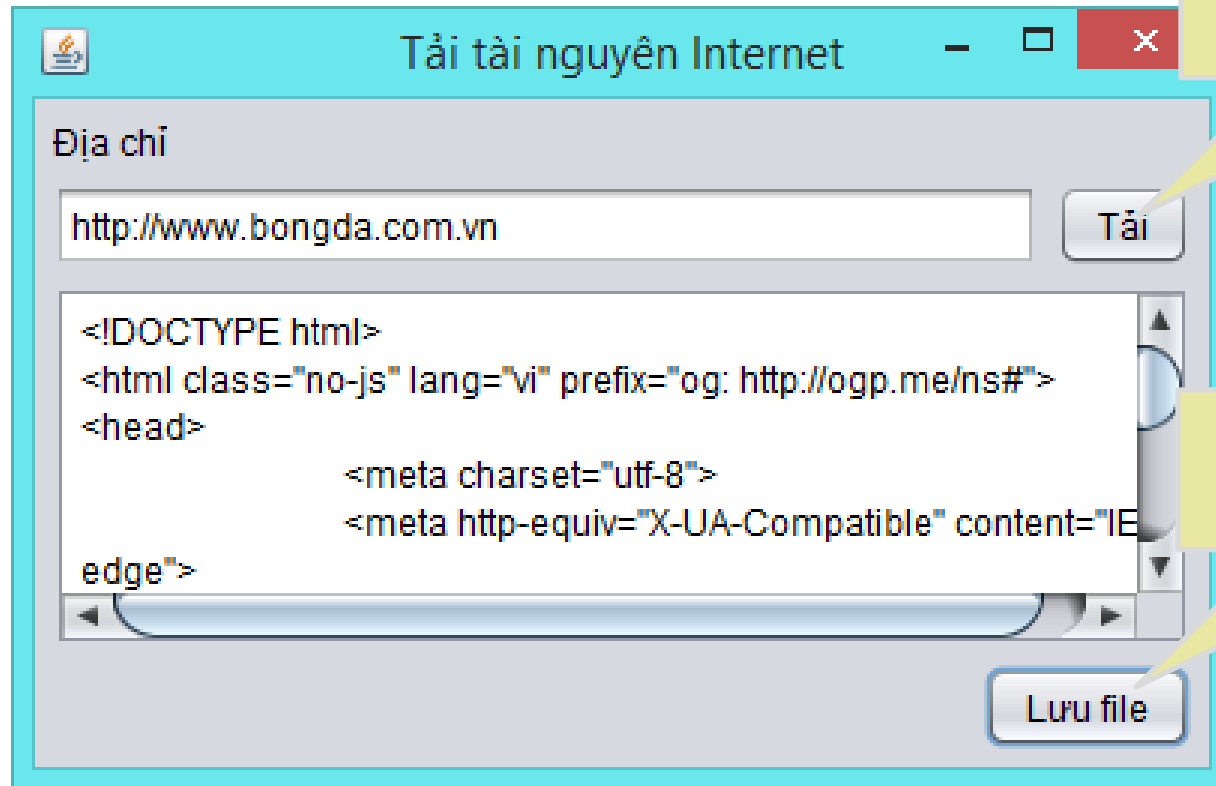
```
is.close();
```

```
byte[] data = buffer.toByteArray();
```

Lấy nội dung trang web

```
buffer.close();
```

Đề mô



Tải và hiển thị trang web
lên ô nội dung bên dưới

Tải và lưu vào file được
nhập từ JFileChooser

Xử lý nút [Tải]

```
ByteArrayOutputStream buffer = new ByteArrayOutputStream();
```

```
URL url = new URL(txtUrl.getText());  
InputStream is = url.openStream();  
byte[] block = new byte[4*1024]; // 4KB  
while(true){  
    int n = is.read(block);  
    if(n <= 0){  
        break;  
    }  
    buffer.write(block, 0, n);  
}  
is.close();
```

```
txtNoiDung.setText(buffer.toString("utf-8"));
```

```
buffer.close();
```

Xử lý nút [Lưu file]

```
if(jFileChooser1.showSaveDialog(this) == JFileChooser.APPROVE_OPTION){  
    File file = jFileChooser1.getSelectedFile();  
    FileOutputStream fos = new FileOutputStream(file);  
  
    URL url = new URL(txtUrl.getText());  
    InputStream is = url.openStream();  
    byte[] block = new byte[4*1024]; // 4KB  
    while(true){  
        int n = is.read(block);  
        if(n <= 0){  
            break;  
        }  
        fos.write(block, 0, n);  
    }  
    is.close();  
    fos.close();  
}
```

Gửi email

- Gửi email là hoạt động thường xuyên xảy ra của 1 ứng dụng. Có thể người dùng gửi cho 1 ai đó, hoặc có thể là thông báo của hệ thống
- Để gửi email trong Java bạn cần 2 thư viện
 - Mail.jar
 - Activation.jar

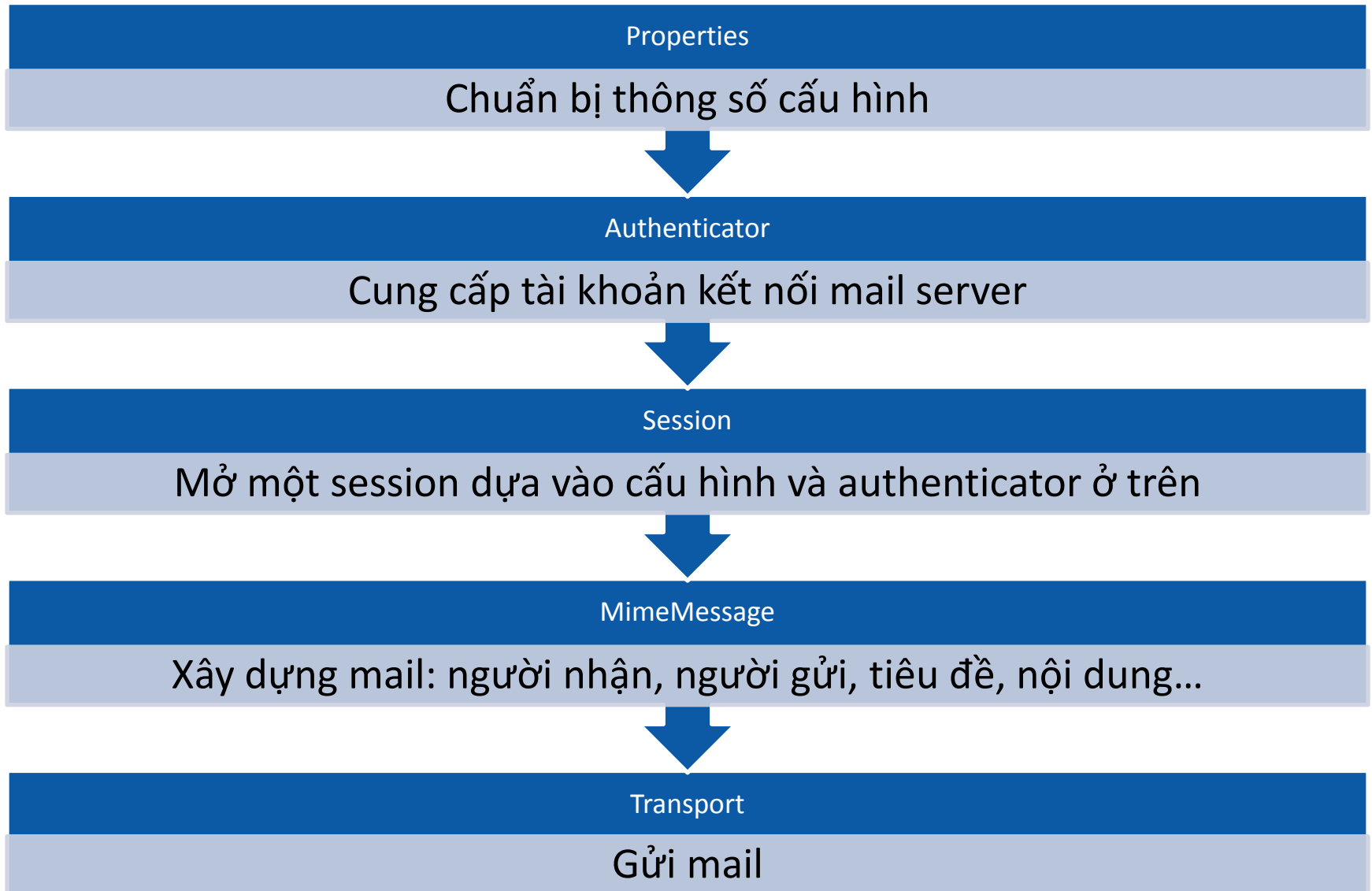


Mail API

- Các phương thức được sử dụng của MimeMessage trong việc gửi email.

Phương thức	Mô tả
setFrom(InternetAddress)	Cung cấp địa chỉ email người gửi
setReplyTo(InternetAddress[])	Cung cấp địa chỉ email người nhận phản hồi
addRecipients(RecipientType, Address[])	Cung cấp danh sách địa chỉ email người cùng nhận
setSubject(String)	Cung cấp tiêu đề mail
setContent(String, String)	Cung cấp nội dung mail
setSentDate(Date)	Cung cấp ngày gửi mail

Giải thuật gửi mail



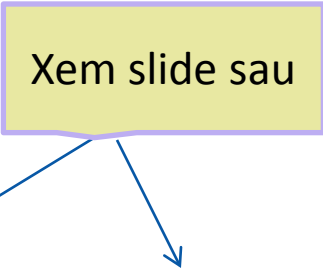
Code gửi email

```
try{
    String fromName = "Nguyễn Văn Sender";
    String fromEmail = "sender@gmail.com";
    String toEmails = "a@gmail.com,b@fpt.edu.vn";
    String subject = "Tiêu đề mail";
    String body = "Nội dung mail";

    Session session = Session.getInstance(config, authenticator);
    MimeMessage mail = new MimeMessage(session);

    InternetAddress sender = new InternetAddress(fromEmail, fromName, "utf-8");
    mail.setFrom(sender);
    mail.setReplyTo(new InternetAddress[]{sender});
    mail.addRecipients(Message.RecipientType.TO, toEmails);
    mail.setSubject(subject, "utf-8");
    mail.setContent(body, "text/html; charset=utf-8");
    mail.setSentDate(new Date());

    Transport.send(mail);
}
catch (Exception e) {
    throw new RuntimeException(e);
}
```



Code gửi email qua Gmail

```
// Các thông số gmail
Properties config = new Properties();
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");

// Đăng nhập gmail
Authenticator authenticator = new Authenticator() {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        String email = "user@gmail.com";
        String password = "songlong";
        return new PasswordAuthentication(email, password);
    }
};
```

Gửi email có CC và BCC

- Để gửi mail cho nhiều người cùng nhận bạn chỉ cần bổ sung các email ấy vào CC hoặc BCC.
- Sự khác biệt giữa CC và BCC là CC là những email sẽ được người nhận nhìn thấy trên mail còn BCC thì không.
- Mã bổ sung email cùng nhận:

```
String ccEmails = "cc1@gmail.com,cc2@yahoo.com,cc3@hotmail.com";  
mail.addRecipients(Message.RecipientType.CC, ccEmails);
```

```
String bccEmails = "bcc1@gmail.com,bcc2@yahoo.com";  
mail.addRecipients(Message.RecipientType.BCC, bccEmails);
```

```
Transport.send(mail);
```


Gửi mail có đính kèm file

```
// Phần nội dung mail chính
MimeBodyPart contentPart = new MimeBodyPart();
contentPart.setContent(body, "text/html; charset=utf-8");

// Phần file đính kèm
MimeBodyPart filePart = new MimeBodyPart();
File file = new File("c:/temp/a.gif");
FileDataSource fds = new FileDataSource(file);
filePart.setDataHandler(new DataHandler(fds));
filePart.setFileName(file.getName());

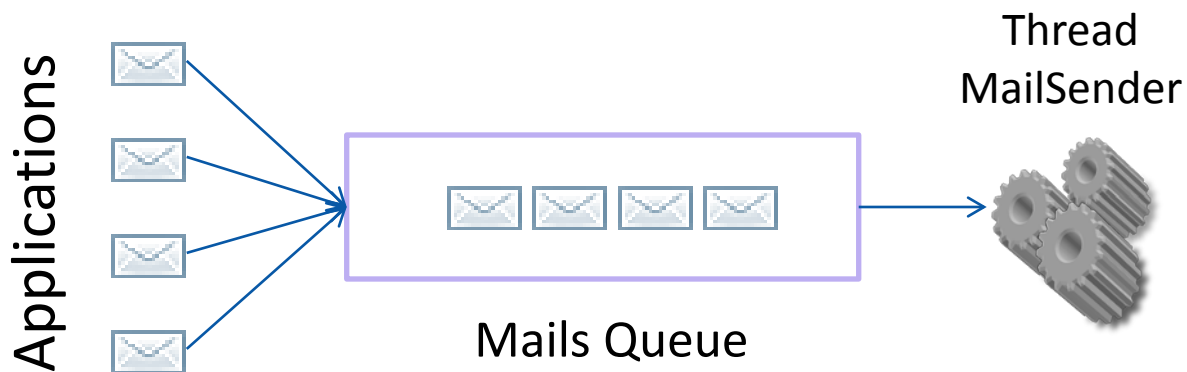
// Bổ sung các phần vào multi-part
MimeMultipart multiPart = new MimeMultipart();
multiPart.addBodyPart(contentPart);
multiPart.addBodyPart(filePart);

// Thiết lập nội dung mail là multi-part
mail.setContent(multiPart);

Transport.send(mail);
```

Queue mail

- Chi phí (CPU và bộ nhớ) dành cho gửi mail là rất cao. Đôi khi hệ thống phải thực hiện gửi nhiều email một lúc dẫn đến tài nguyên cạn kiệt dễ gây tê liệt hệ thống.
- Giải pháp khắc phục là tại queue nắm giữ các mail và có 1 thread chuyển trách gửi tuần tự từng email.



MailSender Thread

```
public class MailSender extends Thread{
    static {
        MailSender sender = new MailSender();
        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() { ...24 lines }
}
```

Xem slide sau

MailSender Thread

```
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;  
    }  
}
```

Sử dụng MailSender Thread

```
Session session = Session.getInstance(config, authenticator);
MimeMessage mail = new MimeMessage(session);

InternetAddress sender = new InternetAddress(fromEmail, fromName, "utf-8");
mail.setFrom(sender);
mail.setReplyTo(new InternetAddress[]{sender});
mail.addRecipients(Message.RecipientType.TO, toEmails);
mail.setSubject(subject, "utf-8");
mail.setContent(body, "text/html; charset=utf-8");
mail.setSentDate(new Date());

//Transport.send(mail);
MailSender.queue(mail);
```

Thay vì gọi ~~Transport.send(mail)~~ để gửi email thì sử dụng **MailSender.queue(mail)** để bổ sung email vào queue và sẽ được gửi bởi 1 Thread khác



KẾT THÚC