

Gửi Email trong Java

Gửi một Email sử dụng Ứng dụng Java là đơn giản, và để bắt đầu, bạn nên có **JavaMail API** và **Java Activation Framework (JAF)** được cài đặt trên thiết bị của bạn.

- Bạn có thể tải phiên bản mới nhất của [JavaMail \(Version 1.2\)](#) từ Website chuẩn của Java.
- Bạn có thể tải phiên bản mới nhất của [JAF \(Version 1.1.1\)](#) từ Website chuẩn của Java.

Download và unzip những file này, trong các thư mục cấp độ cao nhất đã được tạo mới nhất, bạn sẽ tìm thấy một số jar file cho cả hai ứng dụng. Bạn cần thêm các file là **mail.jar** và **activation.jar** vào trong CLASSPATH của bạn.

Gửi một Email đơn giản trong Java

Dưới đây là ví dụ để gửi một Email đơn giản từ thiết bị của bạn. Giả sử rằng **localhost** của bạn được kết nối internet và có khả năng để gửi một Email.

```
// File Name SendEmail.java

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;

public class SendEmail
{
    public static void main(String [] args)
    {
        // Recipient's email ID needs to be mentioned.
        String to = "abcd@gmail.com";

        // Sender's email ID needs to be mentioned
        String from = "web@gmail.com";

        // Assuming you are sending email from localhost
```

```
String host = "localhost";

// Get system properties
Properties properties = System.getProperties();

// Setup mail server
properties.setProperty("mail.smtp.host", host);

// Get the default Session object.
Session session = Session.getDefaultInstance(properties);

try{
    // Create a default MimeMessage object.
    MimeMessage message = new MimeMessage(session);

    // Set From: header field of the header.
    message.setFrom(new InternetAddress(from));

    // Set To: header field of the header.
    message.addRecipient(Message.RecipientType.TO,
                          new InternetAddress(to));

    // Set Subject: header field
    message.setSubject("This is the Subject Line!");

    // Now set the actual message
    message.setText("This is actual message");

    // Send message
    Transport.send(message);

    System.out.println("Sent message successfully....");
}catch (MessagingException mex) {
    mex.printStackTrace();
}
```

```
    }  
}  
}
```

Biên dịch và chạy chương trình trên sẽ gửi một Email đơn giản trong Java:

```
$ java SendEmail  
Sent message successfully....
```

Nếu bạn muốn gửi một Email tới nhiều người nhận, thì phương thức sau sẽ được sử dụng để xác định nhiều Email ID:

```
void addRecipients(Message.RecipientType type,  
                  Address[] addresses)  
throws MessagingException
```

Miêu tả về tham số của phương thức trên trong Java:

- **type:** Nó sẽ được thiết lập là TO, CC hoặc BCC. Với CC biểu diễn Carbon Copy và BCC biểu diễn Black Carbon Copy. Ví dụ: *Message.RecipientType.TO*
- **addresses:** Đây là mảng của Email ID. Bạn sẽ cần sử dụng phương thức *InternetAddresss()* trong khi xác định các Email ID.

Gửi một HTML Email trong Java

Ví dụ dưới đây để gửi một HTML Email trong Java từ thiết bị của bạn. Giả sử rằng **localhost** của bạn được kết nối internet và có khả năng để gửi một Email.

Ví dụ này tương tự như ví dụ trước, ngoại trừ, ở đây chúng ta sử dụng phương thức *setContent()* để thiết lập content của tham số thứ hai là "text/html" để xác định rằng nội dung HTML được bao trong thông báo.

Sử dụng ví dụ này, bạn có thể gửi nội dung HTML lớn bao nhiêu tùy thích.

```
// File Name SendHTMLEmail.java  
  
import java.util.*;  
import javax.mail.*;
```

```
import javax.mail.internet.*;
import javax.activation.*;

public class SendHTMLEmail
{
    public static void main(String [] args)
    {

        // Recipient's email ID needs to be mentioned.
        String to = "abcd@gmail.com";

        // Sender's email ID needs to be mentioned
        String from = "web@gmail.com";

        // Assuming you are sending email from localhost
        String host = "localhost";

        // Get system properties
        Properties properties = System.getProperties();

        // Setup mail server
        properties.setProperty("mail.smtp.host", host);

        // Get the default Session object.
        Session session = Session.getDefaultInstance(properties);

        try{
            // Create a default MimeMessage object.
            MimeMessage message = new MimeMessage(session);

            // Set From: header field of the header.
            message.setFrom(new InternetAddress(from));
```

```
// Set To: header field of the header.
message.addRecipient(Message.RecipientType.TO,
                      new InternetAddress(to));

// Set Subject: header field
message.setSubject("This is the Subject Line!");

// Send the actual HTML message, as big as you like
message.setContent("<h1>This is actual message</h1>",
                  "text/html" );

// Send message
Transport.send(message);
System.out.println("Sent message successfully....");
}catch (MessagingException mex) {
    mex.printStackTrace();
}
}
```

Biên dịch và chạy chương trình này để gửi một HTML Email:

```
$ java SendHTMLEmail
Sent message successfully....
```

Gửi đính kèm trong Email trong Java

Ví dụ này để gửi một Email với đính kèm (attachment) trong Java từ thiết bị của bạn. Giả sử rằng **localhost** của bạn được kết nối internet và có khả năng để gửi một Email.

```
// File Name SendFileEmail.java

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;
```

```
public class SendFileEmail
{
    public static void main(String [] args)
    {

        // Recipient's email ID needs to be mentioned.
        String to = "abcd@gmail.com";

        // Sender's email ID needs to be mentioned
        String from = "web@gmail.com";

        // Assuming you are sending email from localhost
        String host = "localhost";

        // Get system properties
        Properties properties = System.getProperties();

        // Setup mail server
        properties.setProperty("mail.smtp.host", host);

        // Get the default Session object.
        Session session = Session.getDefaultInstance(properties);

        try{
            // Create a default MimeMessage object.
            MimeMessage message = new MimeMessage(session);

            // Set From: header field of the header.
            message.setFrom(new InternetAddress(from));

            // Set To: header field of the header.
            message.addRecipient(Message.RecipientType.TO,
```

```
        new InternetAddress(to));

// Set Subject: header field
message.setSubject("This is the Subject Line!");

// Create the message part
BodyPart messageBodyPart = new MimeBodyPart();

// Fill the message
messageBodyPart.setText("This is message body");

// Create a multipart message
Multipart multipart = new MimeMultipart();

// Set text message part
multipart.addBodyPart(messageBodyPart);

// Part two is attachment
messageBodyPart = new MimeBodyPart();
String filename = "file.txt";
DataSource source = new FileDataSource(filename);
messageBodyPart.setDataHandler(new DataHandler(source));
messageBodyPart.setFileName(filename);
multipart.addBodyPart(messageBodyPart);

// Send the complete message parts
message.setContent(multipart );

// Send message
Transport.send(message);

System.out.println("Sent message successfully....");
}catch (MessagingException mex) {
    mex.printStackTrace();
}
```

```
    }  
  }  
}
```

Biên dịch và thực thi chương trình này để gửi một Email với đính kèm:

```
$ java SendFileEmail  
Sent message successfully....
```

Xác nhận người dùng trong Java

Nếu nó được yêu cầu để cung cấp ID và Password người sử dụng tới một Email Server với mục đích xác nhận, thì bạn có thể thiết lập các thuộc tính này như sau:

```
props.setProperty("mail.user", "myuser");  
props.setProperty("mail.password", "mypwd");
```

Phần còn lại của kỹ thuật gửi Email sẽ như phần giải thích ở trên.