**[JavaMail을 메일 해석에 사용](http://gimmehide.tistory.com/261)**

JavaMail은 통상 메일을 송신하거나 메일을 수신하기 위한 API이지만, 메일 데이터를 스트림으로 읽어들여 해석하는 등의 용도에도 이용할 수 있다. 예를 들어 수신한 메일 데이터의 행선지 주소를 뽑아낸다거나, 본문의 내용을 가공한다거나 하는 경우이다. 스트림으로부터 받아 들인다는 것은 표준 입력으로부터의 데이터도 해석할 수 있는 것으로, 예를 들어 .forward로 메일을 프로그램에 전송 해 거기에서 어떠한 처리를 할 때 등 표준 입력으로 프로그램에 메일 데이터를 넘길 때 JavaMail의 API를 이용할 수 있다.  
  
우선은 표준 입력이 아니라 파일로부터 스트림을 작성하고, 메일을 해석. 메일 샘플 파일은 텍스트로 JIS로 보존해 둔다.

|  |
| --- |
| public class FileMain {   public static void main(String[] args) throws IOException {     try {       InputStream in = new FileInputStream(new File("mail.dummy.txt"));       Session session = Session.getDefaultInstance(           new java.util.Properties(), null);       MimeMessage message = new MimeMessage(session, in);       print(message);     } catch (MessagingException e) {       e.printStackTrace();     } catch (FileNotFoundException e) {       e.printStackTrace();     }   }    private static void print(MimeMessage message) throws MessagingException,       IOException {     System.out.println("---- 모든 헤더 리스트 ----");     Enumeration<Header> allHeaders = message.getAllHeaders();     while (allHeaders.hasMoreElements()) {       Header header = allHeaders.nextElement();       System.out.println(header.getName() + " : " + header.getValue());     }      System.out.println("---- 헤더명을 지정해서 취득(Received) ----");     String[] headers = message.getHeader("Received");     for (String header : headers) {       System.out.println(header);     }      System.out.println("---- 헤더명을 지정해서 취득(Return-Path) ----");     headers = message.getHeader("Return-Path");     for (String header : headers) {       System.out.println(header);     }      System.out.println("---- 헤더명을 지정해서 취득(Subject) ----");     headers = message.getHeader("Subject");     for (String header : headers) {       System.out.println(header);     }     System.out.println("↑디코드 되지 않음");      System.out.println("---- 전용 메소드로 취득(Subject) ----");     System.out.println(message.getSubject());     System.out.println("↑디코드 됨");      System.out.println("---- 전용 메소드로 취득(To) ----");     Address[] allRecipients = message.getAllRecipients();     for (Address address : allRecipients) {       System.out.println(address);     }      System.out.println("---- 메일 본문을 취득 ----");     System.out.println(message.getContent());     System.out.println("--------");   } } |

다음은 파이프 된 표준 입력으로부터 메일 읽어들이기

|  |
| --- |
| public static void main(String[] args) {   try {     InputStream in = System.in;     Session session = Session.getDefaultInstance(         new java.util.Properties(), null);     MimeMessage message = new MimeMessage(session, in);     print(message);   } catch (MessagingException e) {     e.printStackTrace();   } catch (IOException e) {     e.printStackTrace();   } } |

스트림으로서 표준 입력을 전달하고 있다.   
Unix상에서 파이프로 표준 입력을 전달할 경우는 다음과 같이 하면 된다.

|  |
| --- |
| [hoge@www Test]$ export CLASSPATH=lib/mail.jar:lib/sample.jar:lib/activation.jar [hoge@www Test]$ cat mail.dummy.txt | java test.sample.Main |

메일 본문이 첨부파일 + 텍스트 형태인 멀티파트로 되어있는 경우 취득방법

|  |
| --- |
| public static void getMessage(Message message) {         try {             //발신 어드레스를 취득             String mailFrom = ((InternetAddress)message.getFrom()[0]).getAddress();                    //수신 어드레스를 취득             String mailTo = ((InternetAddress)message.getRecipients(Message.RecipientType.TO)[0]).getAddress();               //메일 타이틀 취득             String subject = message.getSubject();                          //메일 본문 취득             Object objContent = message.getContent();              //메일이 멀티파트일 경우             MimeMultipart multipart = null;             String mailContent = "";             if ( objContent instanceof Multipart ) {                 multipart = (MimeMultipart) objContent;                 //멀티파트의 개수를 취득                 int bodyCount = multipart.getCount();                 //멀티파트 개수만큼 반복하며 본문 내용을 취득                 for (int i = 0; i < bodyCount; i++) {                     BodyPart bodypart = multipart.getBodyPart(i);                     Object obj = bodypart.getContent();                                          //첨부파일이 있는 경우 파일을 저장                     if (obj instanceof BASE64DecoderStream) {                         BASE64DecoderStream newObj = (BASE64DecoderStream) obj;                         String filename = MimeUtility.decodeText(bodypart.getFileName());                         File attacheFile = new File(filename);                          FileOutputStream fos = new FileOutputStream(attacheFile);                         byte[] buffer = new byte[1024];                          while (0 < newObj.read(buffer)) {                             fos.write(buffer);                         }                         newObj.close();                         fos.close();                     //메일의 텍스트부분                     } else {                         mailContent = (String)obj;                     }                 }             //텍스트메일인 경우             } else {                 mailContent = (String)objContent;             }                 }catch(Exception ex) {             ex.getMessage();         }     } |