

TECNOLOGIA WEB

Envió y recepción de correo con JavaMail

JavaMail

- JavaMail
 - Es una expansión de Java que facilita el envío y recepción de e-mail desde código java
 - JavaMail implementa el protocolo SMTP (Simple Mail Transfer Protocol) así como los distintos tipos de conexión con servidores de correo -TLS, SSL, autenticación con usuario y password.
 - JavaMail no se incluye en la JDK ni en la JRE, sino que debe conseguirse como un paquete externo.

Enviado Correo – Texto Simple

```
Properties propiedadesEnvio = System.getProperties();
propiedadesEnvio.put("mail.smtp.host", "ip o nombre FQDN smtp");
propiedadesEnvio.put("mail.smtp.starttls.enable", "true");
Session sesionCorreo = Session.getDefaultInstance(propiedadesEnvio);
Message mensajeEnviar = new MimeMessage(sesionCorreo);
mensajeEnviar.setFrom(new InternetAddress("cuentaorigen@dominio"));
mensajeEnviar.addRecipient(Message.RecipientType.TO,
    new InternetAddress("cuenta1@dominio"));
mensajeEnviar.addRecipient(Message.RecipientType.CC,
    new InternetAddress("cuenta2@dominio"));
mensajeEnviar.setSubject("Asunto del correo");
mensajeEnviar.setText("Mensaje a enviar");
Transport.send(mensajeEnviar);
```

Enviado Correo – Texto Simple como multiparte

```
Properties propiedadesEnvio = System.getProperties();
propiedadesEnvio.put("mail.smtp.host", "ip o nombre FQN smtp");
propiedadesEnvio.put("mail.smtp.starttls.enable", "true");
Session sesionCorreo = Session.getDefaultInstance(propiedadesEnvio);
Message mensajeEnviar = new MimeMessage(sesionCorreo);
mensajeEnviar.setFrom(new InternetAddress("cuentaorigen@dominio"));
mensajeEnviar.addRecipient(Message.RecipientType.TO,
    new InternetAddress("cuenta1@dominio"));
mensajeEnviar.addRecipient(Message.RecipientType.CC,
    new InternetAddress("cuenta2@dominio"));
mensajeEnviar.setSubject("Asunto del correo");
MimeBodyPart cuerpoDelMensaje = new MimeBodyPart();
cuerpoDelMensaje.setText("Mensaje del correo");
// Por lo menos setear a vacio
Multipart multiPartes = new MimeMultipart();
multiPartes.addBodyPart(cuerpoDelMensaje);
mensajeEnviar.setContent(multiPartes);
Transport.send(mensajeEnviar);
```

Enviado Correo – Con adjunto

```
Properties propiedadesDeEnvio = System.getProperties();
propiedadesDeEnvio.put("mail.smtp.host", "ip o nombre servidor smtp");
propiedadesDeEnvio.put("mail.smtp.starttls.enable", "true");
MimeBodyPart cuerpoDelMensaje = new MimeBodyPart();
cuerpoDelMensaje.setText("Texto del cuerpo del mensaje");
MimeBodyPart adjunto1 = new MimeBodyPart();
adjunto1.attachFile("nombre Completo Del Adjunto");
Multipart multiparte = new MimeMultipart();
multiparte.addBodyPart(cuerpoDelMensaje);
multiparte.addBodyPart(adjunto1);
Session sesionCorreo = Session.getDefaultInstance(propiedadesDeEnvio);
Message mensajeEnviar = new MimeMessage(sesionCorreo);
mensajeEnviar.setFrom(new InternetAddress("cuentaorigen@dominio"));
mensajeEnviar.addRecipient(Message.RecipientType.TO,
    new InternetAddress("cuentadestino@dominio"));
mensajeEnviar.setSubject("Asunto del correo");
mensajeEnviar.setContent(multiparte);
Transport.send(mensajeEnviar);
```

Lectura de correo

```
Properties propiedadesDeLectura = System.getProperties();
Session sesionLectura = Session.getInstance(propiedadesDeLectura);
Store almacenDeMensajes = sesionLectura.getStore("imaps");
almacenDeMensajes.connect(
    "ip o nombre FQN del servidor IMAP",
    "cuenta", "contraseña");
Folder bandejaDeEntrada = almacenDeMensajes.getFolder("inbox");
if (bandejaDeEntrada != null && bandejaDeEntrada.exists()) {
    //Leyendo la bandeja de entrada
}
```

Leyendo la bandeja de entrada

```
Message mensajes[];
try {
    bandejaDeEntrada.open(Folder.READ_WRITE);
} catch (MessagingException ex) {
    bandejaDeEntrada.open(Folder.READ_WRITE);
}
FlagTerm opcionesDeLectura = new FlagTerm(new Flags(Flags.Flag.SEEN), false);
mensajes = bandejaDeEntrada.search(opcionesDeLectura);
for (Message mensaje : mensajes) {
    System.out.println("----- Mensaje " + mensaje.getMessageNumber() + " -----");
    System.out.println("Fecha de envio : " + mensaje.getSentDate());
    System.out.println("De : " + mensaje.getFrom()[0]);
    System.out.println("Asunto : " + mensaje.getSubject());
    procesaMensaje(mensaje);
    mensaje.setFlag(Flags.Flag.SEEN, true);
    System.out.println();
}
bandejaDeEntrada.close(true);
almacenDeMensajes.close();
```

Procesando el mensaje leido

```
void procesaMensaje(Message mensaje) throws IOException, MessagingException {  
    if (mensaje.isMimeType("multipart/*")) {  
        Multipart multipartes = (Multipart) mensaje.getContent();  
        for (int i = 0; i < multipartes.getCount(); i++) {  
            Part parte = multipartes.getBodyPart(i);  
            procesarPartes(parte, i);  
        }  
    } else {  
        System.out.println("Mensaje : " + mensaje.getContent());  
    }  
}
```


Procesando partes de un mensaje

```
void procesarPartes(Part parte,  
    int i) throws MessagingException, IOException {  
    if (parte.isMimeType("multipart/*")) {  
        Multipart subMultipartes = (Multipart) parte.getContent();  
        for (int j = 0; j < subMultipartes.getCount(); j++) {  
            Part subParte = subMultipartes.getBodyPart(j);  
            System.out.println("Mensaje parte1-" + i + "-"  
                + j + ": " + subParte.getContent());  
        }  
    } else {  
        System.out.println("Mensaje parte2-" + i + ": "  
            + parte.getContent());  
    }  
}
```

Excepciones manejadas

- En el envío de correo se debe manejar la excepción `MessagingException`
- En la lectura de correo también se debe manejar la excepción `MessagingException` y `IOException` al leer el cuerpo de las partes de un mensaje.