## MimeMessage encryptedSignedMessage = encryptMessage(session, signMessage Transport.send(encryptedSignedMessage);

and the [SmimeKey][SmimeKey] to sign it with:

}

}

like this:

.. to sign a message

.. to encrypt a message

private MimeMessage encryptMessage(Session session, MimeMessage message, String

X509Certificate certificate = getCertificateForRecipient(to);

return SmimeUtil.encrypt(session, message, certificate);

Just use the [SmimeUtil][SmimeUtil] with the [MimeMessage][MimeMessage] to be

encrypted and the [X509certificate][X509Certificate] to encrypt it with:

SmimeKey smimeKey = getSmimeKeyForSender(from);

return SmimeUtil.sign(session, message, smimeKey);

Just use the [SmimeUtil][SmimeUtil] with the [MimeMessage][MimeMessage] to be signed

private MimeMessage signMessage(Session session, MimeMessage message, String from

```
Using the library as a receiver
We will assume that you already know how to create a POP or IMAP [Session] [Session] and
how receive a MIME Message with JavaMail, but here is a minimal example how one could read
messages:
```

```
You can then use the [SmimeUtil][SmimeUtil] check the messages content type and find out
if it has a [SmimeState][SmimeState] of ENCRYPTED, SIGNED or NEITHER like this:
```

Store store = session.getStore();

inbox.open(Folder.READ\_ONLY);

store.connect(host, port, user, password);

for (int i = 1, n = inbox.getMessageCount(); i <= n; i++) {</pre>

Folder inbox = store.getFolder("Inbox");

SmimeState smimeState = SmimeUtil.getStatus(mimePart);

If the messages S/MIME state is ENCRYPTED, you can use the [SmimeUtil] [SmimeUtil] with

the encrypted [MimeMessage] [MimeMessage] and the [SmimeKey] [SmimeKey] to decrypt

MimeMessage decryptedMessage = SmimeUtil.decrypt(session, mimeMessage, getSmimek

MimeMessage mimeMessage = (MimeMessage) inbox.getMessage(i);

```
If the messages S/MIME state is SIGNED (the contains a MIME multipart with exactly two body
parts: the signed content and the signature), you can use the [SmimeUtil][SmimeUtil] to
check whether the signature is valid for the signed content and retrieve the signed content like
this:
```

boolean validSignature = SmimeUtil.checkSignature(mimePart)

MimeBodyPart signedContent = SmimeUtil.getSignedContent(mimePart);

If the messages S/MIME state is NEITHER it just means that the message is neither S/MIME encrypted nor S/MIME signed. It may be encrypted or signed by some other means.

© 2022 GitHub, Inc. **Terms** Privacy Security Status Docs Contact GitHub Pricing API Training Blog

**About** 

provided.

약 Fork 4

Sign in

Star 9

Sign up

**M** Readme

☆ 9 stars 4 watching

4 forks

Releases 2 > v2.0.1 ( Latest

+ 1 release **Packages** 

on Jan 31

bbottema Benny Bottema RockyMM Rade Martinović toKrause Torsten Krause

Java 100.0%