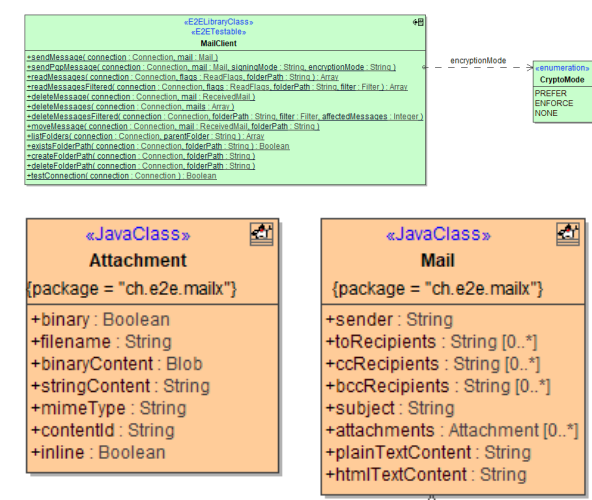


# Sending Messages

Sending messages requires you to assemble an instance of Mail and feeding it to the `sendMessage()` operation.




## Operations

### sendMessage

| Parameter  | Types      | Direction | Description   |
|------------|------------|-----------|---|
| connection | Connection | in        | Specify the <b>Connection</b> object defining the mail server connection parameters to use (see <a href="#">Mail Server Connection</a> ). |
| mail       | Mail       | in        | Prepare a mail object to send.  |

### sendPgpMessage

 **Requires IPgpKeyProvider Implementation**

For PGP encryption to work, be sure to register your implementation of `IPgpKeyProvider` (see [Keys and Certificates](#)).

| Parameter      | Types      | Direction | Description   | Allowed Values / Example        |  |
|----------------|------------|-----------|---|---------------------------------|--|
| connection     | Connection | in        | Specify the <b>Connection</b> object defining the mail server connection parameters to use (see <a href="#">Mail Server Connection</a> ). |                                 |  |
| mail           | Mail       | in        | Prepare a mail object to send.  |                                 |  |
| encryptionMode | String     | in        | Specify one of the encryption modes.  | P<br>R<br>E<br>F<br>E<br>R      | Apply encryption when PGP keys can be retrieved (see <a href="#">Keys and Certificates</a> ), otherwise do not apply any encryption. |
|                |            |           |   | E<br>N<br>F<br>O<br>R<br>C<br>E | Apply encryption when PGP keys can be retrieved, otherwise throw an exception.   |
|                |            |           |   | N<br>O<br>N<br>E                | Do not apply any encryption (default, same behavior as <a href="#">sendMessage()</a> ).  |

#### On this Page:

- Operations
  - [sendMessage](#)
  - [sendPgpMessage](#)
- Types
  - [Mail](#)
  - [Attachment](#)

#### Related Pages:

- [Mail Server Connection](#)
- [JavaMail Library Reference](#)

# Types

## Mail

| Attribute Name          | Type                                | Description  |
|-------------------------|-------------------------------------|--|
| <b>sender</b>           | String                              | Provide the email address of the sender. <div><b>Sending Messages</b><p>Some email servers (especially Exchange servers) require this to be the address of the user specified in the connection. They will raise an imposter exception if this is not the case.</p></div>  |
| <b>toRecipients</b>     | Array of String                     | Provide an array of TO recipient's email addresses.  |
| <b>ccRecipients</b>     | Array of String                     | Provide an array of CC recipient's email addresses.  |
| <b>bccRecipients</b>    | Array of String                     | Provide an array of BCC recipient's email addresses.   |
| <b>subject</b>          | String                              | Provide the email subject.   |
| <b>attachments</b>      | Array of <a href="#">Attachment</a> | Provide an array of email attachments.   |
| <b>plainTextContent</b> | String                              | Provide a plain text message if applicable. <div>Mime messages allow for multipart/alternative parts. If you specify both <b>plainTextContent</b> and <b>htmlTextContent</b>, the JavaMail library will send both as a multipart/alternative when connecting via SMTP. EWS does not allow this. So, when sending through Exchange, the library will pick the HTML content if provided.</div> |
| <b>htmlTextContent</b>  | String                              | Provide an HTML message if applicable. <div>Mime messages allow for multipart/alternative parts. If you specify both <b>plainTextContent</b> and <b>htmlTextContent</b>, the JavaMail library will send both as a multipart/alternative when connecting via SMTP. EWS does not allow this. So, when sending through Exchange, the library will pick the HTML content if provided.</div>      |

## Attachment

### Inline Images

Adding inline images to HTML emails is an art form. Be prepared to see different results depending on the client application receiving your message. Make sure to test your HTML code thoroughly to get good results.

Also, sending large images inline is considered bad style as it blows up message size considerably and could be replaced with links to online locations of the image (e.g. a CDN or publicly accessible web server).

| Attribute Name | Type    | Description  | Possible Values  |
|----------------|---------|--|--|
| <b>binary</b>  | Boolean | Indicates whether <b>binaryContent</b> or <b>stringContent</b> | true Attachment contents can be found in, or shall be taken from, <b>binaryContent</b> . |

|                      |         |  |   |   |
|----------------------|---------|--|---|---|
|                      |         | nt is used.  | false   | Attachment contents can/shall be found in, or shall be taken from, <b>stringContent</b> . |
| <b>filename</b>      | String  | The filename of the attachment.  | <ul style="list-style-type: none"> <li>• can be NULL</li> </ul> <div> <b>Sending Messages</b><br/><br/> This is the name of the file as you want it to appear at the recipient's side, or vice versa.<br/><br/> Do not provide a local file name here and expect the library to load the attachment from that file. Use the <a href="#">File System Adapter</a> instead, and provide the attachment as binary blob content (see <b>binaryContent</b>). </div> |   |
| <b>binaryContent</b> | Blob    | Contains the binary content.   | any Blob<br><br><div> Either one of <b>binaryContent</b> or <b>stringContent</b> is provided, depending on <b>binary</b>. </div>  |   |
| <b>stringContent</b> | String  | Contains the non-binary content.   | any String<br><br><div> Either one of <b>binaryContent</b> or <b>stringContent</b> is provided, depending on <b>binary</b>. </div>  |   |
| <b>mimeType</b>      | String  | Contains the mime type (content type) of the attachment.                                   | any valid MIME content-type, e.g. text/plain, image/png, application/octet-stream etc.<br><br><b>mimeType</b> is unreliable upon receiving emails, i.e. can be NULL. Especially when reading from Exchange servers using <b>exchange</b> or <b>office365</b> connections.   |   |
| <b>contentId</b>     | String  | The content id of the attachment, a unique id across all attachments of the same message.  | When set to e.g. myUniqueId, you can refer to this attachment from HTML message content like  |   |
| <b>inline</b>        | Boolean | Controls whether the attachment is flagged with the corresponding content disposition tag. | true  | Attachment is flagged, and will be displayed inline within a HTML message.                |
|                      |         |  | false   | Attachment is not flagged.  |