

JavaMail

Folder Handling

Operations expecting a **folderPath** argument will treat this the following way:

Folder Path	Description	Example
NULL	This is interpreted as meaning the <Inbox> folder.	
path beginning with /	This is interpreted as a path from the mail connections root folder.	/Archive /Orders
path not beginning with /	This is interpreted as a relative path to the <Inbox> folder, so translates to /<Inbox>/Archive/Orders.	Archive /Orders

The intention of this behavior is to abstract away the fact that <Inbox> has many different names depending on the connection, i.e. IMAP typically uses INBOX, Exchange calls it Inbox, and a German Exchange calls it Posteingang.

Also note that **POP3 does not support folders** other than <Inbox>. Hence when you specify a POP3 connection, **folderPath** parameters other than NULL will raise an exception.

On this Page:

- [Operations](#)
 - [readMessages](#)
 - [readMessagesFiltered](#)
 - [deleteMessage](#)
 - [deleteMessages](#)
 - [deleteMessagesFiltered](#)
 - [moveMessage](#)
 - [sendMessage](#)
 - [sendPgpMessage](#)
 - [listFolders](#)
 - [existsFolderPath](#)
 - [createFolderPath](#)
 - [deleteFolderPath](#)
 - [loadCertificateStore](#)
 - [loadPgpPublicKeyRing](#)
 - [setPgpKeyProvider](#)
- [Types](#)
 - [Connection](#)
 - [ReadFlags](#)
 - [Filter](#)
 - [ReceivedMail](#)
 - [Attachment](#)
 - [MailReceiveStatus](#)
 - [SignatureVerificationResult](#)
 - [MailHeader](#)
 - [Mail](#)

Related Pages:

- [JavaMail Adapter](#)

Operations

readMessages



Optional: IPgpKeyProvider Implementation

For PGP decryption to work, be sure to register your implementation of IPgpKeyProvider (see [Keys and Certificates](#)). If not provided, PGP encrypted messages will still be retrieved, with two attachments that contain the encrypted message information.

Parameter	Type	Direction	Description
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).
flags	ReadFlags	in	Specify an instance of ReadFlags to set some mail reading flags.
folderPath	String	in	Specify a path to a folder to read from. See note regarding folder handling .

readMessagesFiltered



Optional: IPgpKeyProvider Implementation

For PGP decryption to work, be sure to register your implementation of IPgpKeyProvider (see [Keys and Certificates](#)). If not provided, PGP encrypted messages will still be retrieved, with two attachments that contain the encrypted message information.

Parameter	Types	Direction	Description
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).
flags	ReadFlags	in	Specify an instance of ReadFlags (see below).
folderPath	String	in	Specify a path to a folder to read from. See note regarding folder handling .

filter	Filter	in	Specify an instance of Filter containing the filter criteria.
---------------	------------------------	----	--

deleteMessage

Parameter	Types	Direction	Description
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).
mail	ReceivedMail	in	Provide a previously received mail object.

deleteMessages

Parameter	Types	Direction	Description
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).
mail	Array of ReceivedMail	in	Provide an array previously received mail object.

deleteMessagesFiltered

Parameter	Types	Direction	Description	Allowed Values / Example
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).	
folderPath	String	in	Specify a path to a folder to delete from. See note regarding folder handling .	orders/incoming
filter	Filter	in	Specify an instance of Filter containing the filter criteria.  If no filter criteria are set, i.e. if you provide an empty filter, all messages will be deleted.	

affectedMessages	Integer	out	Returns the number of deleted messages.	This might be lower than the total number of messages matching the filter if that number is huge, as mail servers may restrict the maximum number of messages processed in one call. If you want to be sure that all messages matching the filter are actually deleted, call this operation in a loop until affectedMessages becomes 0.
-------------------------	---------	-----	---	---

moveMessage

Parameter	Types	Direction	Description	Allowed Values / Example
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).	
mail	Received Mail	in	Provide an array previously received mail object.	
folderPath	String	in	Specify a path to a specific sub-folder to move the message to. See note regarding folder handling . POP3 connections will cause an exception to be thrown here, as POP3 does not support the concept of folders.	orders /incoming /accepted

sendMessage

Parameter	Types	Direction	Description
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).
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 Connection object defining the mail server connection parameters to use (see Mail Server Connection).		
mail	Mail	in	Prepare a mail object to send.		
encryptionMode	String	in	Specify one of the encryption modes.	P R E F ER	Apply encryption when PGP keys can be retrieved (see Keys and Certificates), otherwise do not apply any encryption.
				E N F O R CE	Apply encryption when PGP keys can be retrieved, otherwise throw an exception.
				N O NE	Do not apply any encryption (default, same behavior as sendMessage()).

listFolders

Parameter	Types	Direction	Description	Allowed Values / Example	
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).		
parentFolder	String	in	Specify the path to the parent folder. See note regarding folder handling .	NULL	List folders below <Inbox>.
				\	List folders below root folder.
				appath	List folders below given path.
folders	Array of String	return	Returns a recursive list of sub-folder paths.	- orders - orders/incoming - orders/incoming/accepted	

existsFolderPath

Parameter	Types	Direction	Description	Allowed Values / Example	
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).		
folderPath	String	in	Specify the path to the folder to check. See note regarding folder handling .	orders/outgoing	
exists	Boolean	return	Returns if the given folder is existent on the mail server.	true	Folder exists.
				false	Folder does not exist.

createFolderPath

Parameter	Types	Direction	Description	Allowed Values / Example	
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).		
folderPath	String	in	Specify the path to the folder to be created. See note regarding folder handling .	orders/outgoing	

deleteFolderPath

Parameter	Types	Direction	Description	Allowed Values / Example	
connection	Connection	in	Specify the Connection object defining the mail server connection parameters to use (see Mail Server Connection).		

folderPath	String	in	Specify the path to the folder to be deleted recursively. See note regarding folder handling .	- orders/incoming/accepted Delete folder accepted from folder incoming. - orders Delete folder orders and all sub-folders and contents.
-------------------	--------	----	--	--

loadCertificateStore

Parameter	Types	Direction	Description	Allowed Values / Example
keyStoreLocation	String	in	Specify the path to the keystore file.	opt/bridge/crypto/mailservercerts.jks
keyStorePassword	String	in	Specify the password of the keystore.	

loadPgpPublicKeyRing



to be discontinued

Future versions of the library will rely on `setPgpKeyProvider()` solely (see below), this operation will disappear. Until then it is still required if you want to verify signatures of unencrypted emails.

Parameter	Types	Direction	Description	Allowed Values / Example
keyRingLocation	String	in	Specify the path to the keyring file.	opt/bridge/crypto/pubkeys.asc

setPgpKeyProvider

Parameter	Types	Direction	Description	Allowed Values / Example
provider	IPgpKeyProvider	in	Instance of a class implementing the IPgpKeyProvider interface	

Types

Connection

Attribute Name	Type	Description	Allowed Values	
debug	Boolean	Specify whether to write additional debug information to the bridgeserver log.	true	Additional debug information is written to the bridgeserver log .
			false	No additional logs (default).
exchangeVersion	String	Specify the Microsoft Exchange version (optional).	2007_SP1	Exchange 2007 Service Pack 1
			2010	Exchange 2010
			2010_SP1	Exchange 2010 Service Pack 1
			2010_SP2	Exchange 2010 Service Pack 2 and higher (default)

host	String	Specify the host name/IP address of the mail server.	a valid host name or IP address	for <i>exchange</i> connections this can be left NULL, triggering Autodiscover (which might fail) for <i>office365</i> connections this is ignored
microsoftOnlineCredentials	Object	Specify OAuth login credentials for Microsoft Online connections	a valid set of credentials	either this or usernamePasswordCredentials must be set
port	Integer	Specify the port of the mail server (optional)	a valid port number	If omitted (NULL), the standard port for the chosen protocol and security is used.
protocol	String	Specify the protocol used for e-mail retrieval and storage. Values are not case sensitive.	imap	Use the IMAP protocol.
			imaps	Use IMAP over SSL.
			pop3	Use POP3.
			smtp	Use the SMTP protocol.
			smtps	Use the SMTPS protocol.
			exchange	Use Exchange Web Service (EWS) API with a hosted Exchange Server
			office365	Use Exchange Web Service (EWS) API with a Office 365 Server (will ignore host).
security	String	Specify the encryption. Values are not case sensitive. exchange and office365 protocols ignore this setting and use ssl anyway.	none	No encryption used. exchange and office365 protocol (see attribute protocol) use SSL in any case, this cannot be overridden.
			ssl	Use SSL encryption.
			starttls	Use TLS encryption. exchange and office365 protocol (see attribute protocol) use SSL in any case, this cannot be overridden. Make sure to pick imap, pop3, or smtp with this setting, not imaps, pop3s, or smtps.
timeoutSeconds	Integer	Specify the number of seconds after which a request (send or receive) times out.	any positive integer Default is 0, which translates to the following defaults: - infinite timeout for imap(s), pop3(s), and smtp(s) 100 seconds for exchange	
usernamePasswordCredentials	Object	Specify the user's credentials	a valid set of credentials	either this or microsoftOnlineCredentials must be set
verifyHost	Boolean	Specify whether for SSL connections the mail server host shall be verified or not (optional).	true	Verify the mail server's SSL certificate (default, see also security).
			false	Do not verify the mail server.
Username and Password Credentials				
usernamePasswordCredentials.username	String	Specify the user account on the mail server.	a valid user name	Some SMTP servers allow unauthorized sending, in these cases you may leave it as NULL.
usernamePasswordCredentials.password	String	Specify the user's password.	a valid password	Some SMTP servers allow unauthorized sending, in these cases you may leave it as NULL.
Microsoft Online Credentials (only applicable to office365 protocol)				
microsoftOnlineCredentials.clientId	String	client id of your app registration	For details on registering EWS access via OAuth please follow the steps as outlined here: https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/how-to-authenticate-an-ews-application-by-using-oauth Make sure to follow the path for "app-only authentication"	

microsoftOnlineCredentials.clientSecret	String	client secret of your app registration		
microsoftOnlineCredentials.tenantId	String	tenant id of your microsoft online organization		
microsoftOnlineCredentials.impersonatedUser	String	name (email address) of the mailbox you want to impersonate	a valid email address	Provide the email address of the mailbox you want to interact with (e.g. read messages from)
Proxy Settings (only applicable to HTTP based connections, i.e. exchange or office365 protocol)				
proxy.host	String	Specify the host name/IP address of the proxy server.	a valid host name or IP address	
proxy.port	Integer	Specify the port of the proxy server.	a valid port number	
proxy.domain	String	Specify the user domain on the proxy server (optional). Some Windows-based proxy servers might require this for authentication.	a valid ActiveDirectory domain name	
proxy.user	String	Specify the user account on the proxy server (optional).	a valid user	
proxy.password	String	Specify the user's password (optional).	a valid password	

ReadFlags

Attribute Name	Type	Description	Allowed Values	Comment
maxResultCount	Integer	Specify a limit for the number of messages to be returned. If you do not provide a filter as parameter, it makes sense to provide a meaningful limit, as the complete result set will be kept in memory.	any positive integer	Default is NULL, meaning no limit is imposed.
deleteOnServer	Boolean	Specify whether successfully read messages will automatically be deleted on the server.	true	Delete messages on server after successful read. Messages that cannot be read successfully will not be deleted.
			false	Do not delete messages on server (default).
markAsRead	Boolean	Specify whether successfully read messages will automatically be marked as read on the server.	true	Mark messages as read on server after successful read. Messages that cannot be read successfully will not be marked read.
			false	Do not mark messages read on server (default).
withContent	Boolean	Specify whether the message body (HTML and/or plain text) should be populated.	true	Populate message body.
			false	Message body is empty (default).
withAttachments	Boolean	Specify whether attachments should be retrieved.	true	Retrieve attachments.
			false	Do not retrieve attachments (default).
withEmbeddedMails	Boolean	Specify whether attachments with mime type message/rfc822 should be retrieved as embedded messages. Requires withAttachments to be true to have an effect.	true	Retrieve attachments with mime type message/rfc822 as embedded messages.

			false	Do not retrieve attachments with mime type message/rfc822 as embedded messages, but as regular attachments (default).
withHeaders	Boolean	Specify whether messaging headers should be populated.	true	Populate messaging headers.
			false	Messaging headers are empty (default).
withRawContents	Boolean	Specify whether the raw message source should be retrieved.	true	Retrieve raw message source. This is similar to "view source" on most email clients, hence the output can be huge.
			false	Do not retrieve raw message source (default).
verifySignatures	Boolean	Specify whether for any signed messages a verification of the signature should be attempted.	true	Verify signature. To successfully verify PGP signatures, a public key ring must be provided with the connection parameter.
			false	Do not verify signature (default).

Filter

Attribute Name	Type	Description	Allowed Values	
unreadOnly	Boolean	Specify whether only unread messages should be returned.	true	Return unread messages only.
			false	Return all messages (default).
fromTimestamp	DateTime	Specify the beginning of a date range that should be used to retrieve messages (see also toTimestamp below). Make sure that fromTimestamp and toTimestamp do not contradict each other as this is not checked.	any DateTime	
toTimestamp	DateTime	Specify the end of a date range that should be used to retrieve messages (see also fromTimestamp below). Make sure that fromTimestamp and toTimestamp do not contradict each other as this is not checked.	any DateTime	
subjectSubstring	String	Specify a filter substring for the email subject. Only messages containing this string as a literal substring are returned. subjectSubstring is case sensitive.	any String with printable characters (case sensitive)	
sender	String	Specify a dedicated sender you want to retrieve messages from. Only message sent from this address are returned	any valid email address (case insensitive)	

senderRegex	String	Specify a regex filter to be applied to the mail sender. Only messages with the sender matching the filter expression are returned.	any valid regular expression (Java style)
subjectRegex	String	Specify a regex filter to be applied to the mail subject. Only messages with the subject matching the filter expression are returned.	any valid regular expression (Java style)
attachmentNameRegex	String	Specify a regex filter to be applied to the mail attachments. Only messages with at least one attachment's name matching the filter expression are returned. Make sure that you are using withAttachments = true when reading the mails (see ReadFlags).	any valid regular expression (Java style)

ReceivedMail

ReceivedMail inherits from [Mail](#), so contains all attributes of Mail plus a few more.

Attribute Name	Type	Description	Possible Values
sender	String	Contains the email address of the sender.	
toRecipients	Array of String	Contains an array of TO recipient's email addresses.	
ccRecipients	Array of String	Contains an array of CC recipient's email addresses.	
bccRecipients	Array of String	Contains an array of BCC recipient's email addresses.	
subject	String	Contains the email subject.	
attachments	Array of Attachment	Contains an array of email attachments. If this is not populated, you may have set withAttachments = false when reading the mails (see ReadFlags).	
plainTextContent	String	Contains the plain text message if available. Mime messages may contain HTML, Plain Text, or both.	
htmlTextContent	String	Contains the HTML message if available. Mime messages may contain HTML, Plain Text, or both.	
id	String	Contains the unique ID of the message within a folder on the mail server.	
receiveDate	DateTime	Contains a timestamp indicating when the message arrived on the server.	
sentDate	DateTime	Contains a timestamp indicating when the message has been sent.	
headers	Array of MailHeader	Contains an array of internet email headers. If this is not populated, you may have set withHeaders = false when reading the mails (see ReadFlags).	

embeddedMails	Array of ReceivedMail	Contains an array of nested/embedded mail objects (attached email messages, e.g. .eml or .msg attachments). If this is not populated, you may have set withAttachments = false when reading the mails (see ReadFlags).		
receiveStatus	MailReceiveStatus	Contains detailed information about the receive status of the email.		
folder	String	Contains the folder the message was read from.	NULL	<Inbox>.
			folder path	The folder path below <Inbox>.
signed	Boolean	Contains the signing status of the message.	true	Message is signed.
			false	Message is not signed.
encrypted	Boolean	Contains the encryption status of the message. detection is supported, however receiveStatus.success will always be false, as decryption is not yet supported	true	Message is encrypted.
			false	Message is not encrypted.
signatureVerificationResult	SignatureVerificationResult	Contains detailed information about the verification of signatures.		
rawMessageText	Blob	Contains the raw message. If this is not populated, you may have set withRawContents = false when reading the mails (see ReadFlags).		

Attachment

Attribute Name	Type	Description	Possible Values	
binary	Boolean	Indicates whether binaryContent or stringContent is used.	true	Attachment contents can be found in, or shall be taken from, binaryContent .
			false	Attachment contents can/shall be found in, or shall be taken from, stringContent .
filename	String	The filename of the attachment.	can be NULL Sending Messages This is the name of the file as you want it to appear at the recipient's side, or vice versa. Do not provide a local file name here and expect the library to load the attachment from that file. Use the File System Adapter instead, and provide the attachment as binary blob content (see binaryContent).	
binaryContent	Blob	Contains the binary content.	any Blob Either one of binaryContent or stringContent is provided, depending on binary .	

stringContent	String	Contains the non-binary content.	any String Either one of binaryContent or stringContent is provided, depending on binary .	
mimeType	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. mimeType is unreliable upon receiving emails, i.e. can be NULL. Especially when reading from Exchange servers using exchange or office365 connections.	
contentId	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	
inline	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.

MailReceiveStatus

Attribute Name	Type	Description	Possible Values	
success	Boolean	Returns whether the entire message could be read with all requested attributes (see ReadFlags). A message will not be deleted if success is false - regardless of what has been specified with deleteOnServer (see ReadFlags).	true	Message could be read with all requested attributes.
			false	Message could not be read or some requested attributes are missing.
deletedOnServer	Boolean	Returns whether the message has been deleted after retrieval. A message will not be deleted if success is false - regardless of what has been specified with deleteOnServer (see ReadFlags).	true	Message has been deleted.
			false	Message has not been deleted.
markedAsRead	Boolean	Returns whether the message has been marked as read after retrieval. A message will not be marked as read if success is false - regardless of what has been specified with markAsRead (see ReadFlags).	true	Message has been marked as read.
			false	Message has not been marked as read.
errorMessage	String	Returns further information on the issue if success is false.		
stacktrace	Array of String	Returns a stacktrace if the problem was caused by an unexpected Java exception.		
protocol	Array of String	Returns the steps performed while receiving the message as a protocol. This can give a hint in case of problems.		

SignatureVerificationResult

Attribute Name	Type	Description	Possible Values	
verified	Boolean	Returns whether the signature is valid or not.	true	Message signature is valid.

			false	Message signature is invalid, or no verification has been requested (see verifySignatures in ReadFlags).
errorMessage	String	Returns details of a failed verification.		

MailHeader

Attribute Name	Type	Description	Possible Values
name	String	Name of the header.	
value	String	Value of the header.	

Mail

Attribute Name	Type	Description
sender	String	Provide the email address of the sender. Sending Messages 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.
toRecipients	Array of String	Provide an array of TO recipient's email addresses.
ccRecipients	Array of String	Provide an array of CC recipient's email addresses.
bccRecipients	Array of String	Provide an array of BCC recipient's email addresses.
subject	String	Provide the email subject.
attachments	Array of Attachment	Provide an array of email attachments.
plainTextContent	String	Provide a plain text message if applicable. Mime messages allow for multipart/alternative parts. If you specify both plainTextContent and htmlTextContent, 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.
htmlTextContent	String	Provide an HTML message if applicable. Mime messages allow for multipart/alternative parts. If you specify both plainTextContent and htmlTextContent, 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.