

Nitido Mail Controller API

&

Data Type Definition Specification

Application Interfaces	4
Mail Controller	4
loadFolderList	4
getFolderList	4
loadFolderCount	5
renameFolder	5
addFolder	6
emptyFolder	7
deleteFolder	7
getSelectedFolder	8
getFolderObject	8
loadMessages	8
getMessages	9
loadMessage	10
getMessage	10
deleteMessages	11
moveMessages	11
setMessageFlag	12
getAttachmentURL	13
sendMessage	13
saveAsDraft	14
markSpam	14
getExternalEmail	15
loadUsage	15
getUsage	16

Data Type Definitions	17
Compose Message - used when sending a message or saving a draft	17
Message	18
Message Header	19
Attachment	20
Email Address	20
Folder	21
Usage	21

Application Interfaces

Mail Controller

Mail Controller provides methods to access all server-side mail-related functions and data. The Mail Controller provides an out-of-the-box caching mechanism that caches the currently selected/opened folder.

loadFolderList

Load the currently logged-in user's mail folders into the cache.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *ignoreError* (boolean, optional) - indicates whether to ignore the server response containing an error code.

Returns: N/A

Events Fired:

- MailController.loadFolderList.START_BUSY
- MailController.loadFolderList.STOP_BUSY
- MailController.loadFolderList.EXCEPTION
- MailController.loadFolderList.SUCCESS

Expected Behavior:

- Should load a list of Mail Folders sorted by folder name in ascending order into the controller's cache.

Error Handling: N/A

getFolderList

Returns a list of mail *Folder* objects belonging to the currently logged-in user.

Parameters: *none*

Returns:

- List of Mail Folders (Folder [])

Events Fired: N/A

Expected Behavior:

- Should return a list of mail Folders from the controller cache.

Error Handling: N/A

loadFolderCount

Load the folder count (total/unread messages) of user's mail folders into the cache.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *ignoreError* (boolean, optional) - indicates whether to ignore the server response containing an error code.

Returns: N/A

Events Fired:

- MailController.loadFolderCount.START_BUSY
- MailController.loadFolderCount.STOP_BUSY
- MailController.loadFolderCount.EXCEPTION
- MailController.loadFolderCount.SUCCESS

Expected Behavior:

- Should load the folder counts for each user's mail folder into the controller cache

Error Handling: N/A

renameFolder

Rename the given (existing) folder.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:

- *nfn* (String, required) - New folder name
- *ofid* (String, required) - Folder id
- *pfid* (String, optional) - Folder's Parent id (if folder being renamed is a sub-folder)

Returns: N/A

Events Fired:

- MailController.renameFolder.START_BUSY
- MailController.renameFolder.STOP_BUSY
- MailController.renameFolder.EXCEPTION
- MailController.renameFolder.SUCCESS
- MailController.renameFolder.DUPLICATE_FOLDER_NAME_EXCEPTION

Expected Behavior:

- Should rename the Mail folder.

Error Handling: N/A

addFolder

Add a new mail folder.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fn* (String, required) - Folder name
 - *pfid* (String, optional) - Folder's Parent id (if folder being added is a sub-folder)

Returns: N/A

Events Fired:

- MailController.addFolder.START_BUSY
- MailController.addFolder.STOP_BUSY
- MailController.addFolder.EXCEPTION
- MailController.addFolder.SUCCESS
- MailController.addFolder.DUPLICATE_FOLDER_NAME_EXCEPTION

Expected Behavior:

- Should add a new folder

Error Handling: N/A

emptyFolder

Empty the specified mail folder.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - Folder id

Returns: N/A

Events Fired:

- MailController.emptyFolder.START_BUSY
- MailController.emptyFolder.STOP_BUSY
- MailController.emptyFolder.EXCEPTION
- MailController.emptyFolder.SUCCESS

Expected Behavior:

- Should empty all messages from the specified folder. If the specified folder is Trash, messages are removed from the server; otherwise, messages are moved to the Trash folder.

Error Handling: N/A

deleteFolder

Delete the specified mail folder.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - Folder id

Returns: N/A

Events Fired:

- MailController.deleteFolder.START_BUSY
- MailController.deleteFolder.STOP_BUSY
- MailController.deleteFolder.EXCEPTION
- MailController.deleteFolder.SUCCESS

Expected Behavior:

Error Handling: N/A

getSelectedFolder

Returns the currently selected/opened *Folder* object.

Parameters: *none*

Returns:

- Selected folder (Folder)

Events Fired: N/A

Expected Behavior:

- Should return the currently selected folder object

Error Handling: N/A

getFolderObject

Returns a *Folder* object for the specified folder id.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - Folder id

Returns:

- Specified folder (Folder) or *false* if none found.

Events Fired: N/A

Expected Behavior:

- Should return the currently selected folder object

Error Handling: N/A

loadMessages

Load messages for the given folder from *start* to *end* indices. However, if *ignoreCache* is true, a

request to load the messages from the server is made regardless of whether the specified messages exist in the cache.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - Folder id
 - *start* (Number, required) - Start position/index of the messages to load
 - *end* (Number, required) - End position/index of the messages to load
 - *sc* (String, required) - Sort column
 - *so* (String, required) - Sort order (one of 2 values: *asc* or *desc*)
 - *sk* (Object, optional) - Search keyword(s). Uses the following format:
 - Ex: { "subject": "test", "sender": "wau@nitido.com" }
 - *check* (boolean, optional) - Indicates whether to just perform a “check mail” operation
 - *pop* (boolean, optional) - Indicates whether to perform
 - *ignoreCache* (boolean, optional) -

Returns: N/A

Events Fired:

- MailController.loadMessages.START_BUSY
- MailController.loadMessages.STOP_BUSY
- MailController.loadMessages.EXCEPTION
- MailController.loadMessages.SUCCESS

Expected Behavior:

- Should load a sorted list of mail Messages into the controller’s cache. Sort order and field are based on specified parameters.

Error Handling: N/A

getMessages

Returns a list of Mail Messages (from *start* to *end* index) for the specified folder.

Parameters:

- *options* (Object, required) – **NB:** This is the same configuration object passed into *loadMessages*.

Returns:

- List of Messages (Message [])

Events Fired: N/A

Expected Behavior:

- Should return a list of Mail from the controller cache.

Error Handling: N/A

loadMessage

Load the specified message. If the specified message exists in the controller cache, it is returned instead of re-requesting it from the server.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - The folder id
 - *mid* (String, required) - The message id

Returns: N/A

Events Fired:

- MailController.loadMessage.START_BUSY
- MailController.loadMessage.STOP_BUSY
- MailController.loadMessage.EXCEPTION
- MailController.loadMessage.SUCCESS

Expected Behavior:

- Should load a Message into the controller's cache.

Error Handling: N/A

getMessage

Returns the specified message from the controller's cache.

Parameters:

- *options* (Object, required) – **NB:** This is the same configuration object passed into *loadMessages*.

Returns:

- Message object (Message)

Events Fired: N/A

Expected Behavior:

- Should return a Message from the controller cache.

Error Handling: N/A

deleteMessages

Delete the given messages for the specified folder id.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) - The folder id
 - *mid* (String [], required) - The message id(s)

Returns: N/A

Events Fired:

- MailController.deleteMessages.START_BUSY
- MailController.deleteMessages.STOP_BUSY
- MailController.deleteMessages.EXCEPTION
- MailController.deleteMessages.SUCCESS

Expected Behavior:

- Should delete the specified messages

Error Handling: N/A

moveMessages

Move the specified messages from the source folder to the target folder.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *sfid* (String, required) - The source folder id
 - *tfid* (String, required) - The target folder id
 - *mid* (String [], required) - The message id(s)

Returns: N/A

Events Fired:

- MailController.moveMessages.START_BUSY
- MailController.moveMessages.STOP_BUSY
- MailController.moveMessages.EXCEPTION
- MailController.moveMessages.SUCCESS

Expected Behavior:

- Should move the specified messages from the source folder to the target folder

Error Handling: N/A

setMessageFlag

Set the read/unread status for the given messages.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *flag* (Number, required) - This parameter can have one of the following values:
 - **0** - to mark messages as unread
 - **1** - to mark messages as read
 - **2** - Mark message as “flagged”
 - **3** - Unmarked message as “flagged”
 - *on|<tagname>* - Adds a user-defined tag to the message. *<tagname>* is a custom user-defined tag. *|* is the Unix pipe character.
 - *off|<tagname>* - Removes a user-defined tag from the message. *<tagname>* is a custom user-defined tag. *|* is the Unix pipe character.
 - *fid* (String, required) - Folder id
 - *mid* (String [], required) - The message id(s)

Returns: N/A

Events Fired:

- MailController.setMessageFlag.START_BUSY
- MailController.setMessageFlag.STOP_BUSY
- MailController.setMessageFlag.EXCEPTION
- MailController.setMessageFlag.SUCCESS

Expected Behavior:

- Should change the status of the messages to the specified flag value

Error Handling: N/A

getAttachmentURL

Takes an attachment object (property of a Message object) and returns a URL to the attachment.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *attachment* (Object) - An Attachment object to generate a URL for

Returns:

- URL to download attachment (String)

Events Fired: N/A

Expected Behavior:

- Should return the URL to the specified attachment

Error Handling: N/A

sendMessage

Send a message. At least one recipient must be specified in order to successfully send the message.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *msg* (Compose Message, required) - Message object
 - *action* (String, optional) - Indicates if email being sent is a reply or a forward to an original email. Can have one of the following two values:
 - *f* - Indicates this is a forward action
 - *r* - Indicates this is a reply action
 - *omid* (String, optional) - The original message id (when replying to, or forwarding, a message)
 - *fid* (String, optional) - The original folder id (when replying to, or forwarding, a message)

Returns: N/A

Events Fired:

- MailController.sendMessage.START_BUSY
- MailController.sendMessage.STOP_BUSY

- MailController.sendMessage.EXCEPTION
- MailController.sendMessage.SUCCESS

Expected Behavior:

- Should send the message to the specified recipients

Error Handling: N/A

saveAsDraft

Save the given message as a draft.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *msg* (Compose Message, required) - Message object

Returns: N/A**Events Fired:**

- MailController.saveDraft.START_BUSY
- MailController.saveDraft.STOP_BUSY
- MailController.saveDraft.EXCEPTION
- MailController.saveDraft.SUCCESS

Expected Behavior:

- Should save the message in the currently logged-in user's Drafts folder

Error Handling: N/A

markSpam

Mark the given message as spam/not spam

Parameters:

- *options* (Object, required) – A configuration object with the following properties:
 - *fid* (String, required) – Folder id the message belongs to
 - *mid* (String [], required) - The message id(s)

Returns: N/A

Events Fired:

- MailController.markSpam.START_BUSY
- MailController.markSpam.STOP_BUSY
- MailController.markSpam.EXCEPTION
- MailController.markSpam.SUCCESS

Expected Behavior:

- Should mark/unmark the specified messages as *spam*.

Error Handling: N/A**getExternalEmail**

POP Account Aggregation. Initiates a POP account aggregation request.

Parameters: *none*

Returns: N/A

Events Fired:

- MailController.getExternalEmail.START_BUSY
- MailController.getExternalEmail.STOP_BUSY
- MailController.getExternalEmail.EXCEPTION
- MailController.getExternalEmail.SUCCESS

Expected Behavior:

- Should initiate a request to load messages from external accounts into their specified folders (the folder external messages are loaded into is a user-defined configuration).

Error Handling: N/A**loadUsage**

Load the mail quota from the server for the currently logged-in user.

Parameters:

- *options* (Object, required) – A configuration object with the following properties:

- *ignoreError* (boolean, optional) - indicates whether to ignore the server response containing an error code.

Returns: N/A

Events Fired:

- MailController.loadUsage.START_BUSY
- MailController.loadUsage.STOP_BUSY
- MailController.loadUsage.EXCEPTION
- MailController.loadUsage.SUCCESS

Expected Behavior:

- Should load the mail quota from the server for the currently active user.

Error Handling: N/A

getUsage

Get the mail quota for the currently logged-in user.

Parameters: *none*

Returns:

- Usage object (Usage)

Events Fired: N/A

Expected Behavior:

- Should load the mail quota from the server for the currently active user.

Error Handling: N/A

Data Type Definitions

Compose Message - *used when sending a message or saving a draft*

Property	Type	Required	Description
mid	String		Current Message id (ex: if this is an edit of a currently drafted message)
sender	String	X	Sender's email address
to	String []	X	Recipient email address(es)
cc	String []		Carbon-Copied email address(es)
bcc	String []		Blind Carbon-Copied email address(es)
subject	String		Message subject
body	String		Message body
htmlFormat	Boolean		Specifies whether message body is in HTML format (false indicates plain text)
priority	Number	X	Message priority. Can be any value from 1-5.
reqReceipt	Boolean	X	Request return receipt
attachment	String []		<p>There may be more than one "attachment" parameter when sending multiple attachments of the following two formats:</p> <p><code>message:<part-index>:<mid>:<folder-name></code></p> <p>or</p> <p><code>upload:<mime>:<actual-filename>:<original-filename></code></p> <p>The "message" and "upload" component of the attachment parameter describe whether the attachment comes from another message (ex: forwarded message) or from an uploaded file.</p> <p>Each component must be separated by a colon (:). Each component must be Base64 encoded so that all special characters (ie: colon, etc.) are removed.</p>

Message

Property	Type	Description
mid	String	Message's id
fn	String	Message's folder id
sender	Email Address	Sender's <i>Email Address</i>
to	Email Address []	Recipient(s) list
cc	Email Address []	Carbon-Copied list of <i>Email Address(es)</i>
bcc	Email Address []	Blind Carbon-Copied list of <i>Email Address(es)</i>
subject	String	Message subject
bodyhtml	String	Message body
attachments	Attachment []	List of <i>Attachment(s)</i>
date	Number	The message date (represented in <i>Long</i> format)
hasBlockedImages	Boolean	Indicates whether email's images are blocked (to prevent spam)
isCertified	Boolean	Indicates whether email has been signed as "certified" (ie: GoodMail)
midNext	String	
midPrev	String	
msgheaders	Object []	List of message headers in the format of: <code>{ key: value }</code>
priority	Number	Message priority. Can be any value from 1-5.
reqReceipt	Boolean	Indicates if the sender wants to receive a notification when the message is viewed by recipient.
timezone	Number	Represents the time zone to be used for converting the date field (for display purposes).

Message Header

Property	Type	Description
mid	String	Message's id
fn	String	Message's folder id
sender	String	Sender's <i>Email Address</i>
subject	String	Message subject
size	Number	Message size (in KB)
date	Number	The message date (represented in Long format)
flagged	Boolean	Indicates if the message has been "flagged"
priority	Boolean	Indicates if the message has been flagged as important by the sender.
reqReceipt	Boolean	Request return receipt
attachments	Boolean	Indicates whether the Message has any attachments
isFwd	Boolean	Indicates if the message has been forwarded. It is used for display purposes on the client.
isRe	Boolean	Indicates if the user replied to the sender . It is used for display purposes on the client.
isCertified	Boolean	Indicates if the message has been certified.
unread	Boolean	Indicates if the message has been flagged as read/unread. We keep this for backward compatibility.
flags	String []	Holds an array of user flags set on the message.
msgheaders	Object []	List of message headers in the format of: { key: value }

Attachment

Property	Type	Description
attachmentSize	Number	Attachment's file size (in bytes)
disp	String	Attachment's disposition. Two possible values: <ul style="list-style-type: none">• <i>inline</i>• <i>attachment</i>
fin	String	Filename
fn	String	Folder id
mid	String	Message id
mime	String	
partIndex	String	Part Index

Email Address

Property	Type	Description
address	String	Sender's email address
personal	String	Sender's display name

Folder

Property	Type	Description
sys	Boolean	Indicates whether this is a system folder
totalMsgNum	Number	Total messages in this folder
totalUnreadMsgNum	Number	Total unread messages in this folder
fid	String	Folder id
fn	String	Folder name (display name)
allowSub	Boolean	Indicates whether this folder allows sub-folders under it

Usage

Property	Type	Description
percentage	Number	Total usage represented as a percentage
total	Number	Total usage available to the user