

[Back to: CUMI API Overview](#)

Contents

- [1 About Mailboxes and Folders](#)
- [2 Mailbox Folder Operations](#)
 - ◆ [2.1 Inbox Folder Operations](#)
 - ◆ [2.2 Sent Items Folder Operations](#)
 - ◆ [2.3 Deleted Items Folder Operations](#)
 - ◆ [2.4 Offset and Limit](#)
- [3 Sorting](#)
- [4 Filtering](#)
 - ◆ [4.1 Examples](#)

About Mailboxes and Folders

The root of Cisco Unity Connection Messaging Interface (CUMI) API is the Mailbox resource that is associated with each user. This contains some general information about the user's mailbox, and also contains a Folders resource that lists the folders for the mailbox. This list is currently fixed, although it is possible that folders may be added in the future.

Beginning with Cisco Unity Connection 10.5 and later, when one or more tenants are configured on a single installation of Cisco Unity Connection, a user with **Mailbox Access Delegate Account** role and belonging to a particular tenant will be able to list messages of all the users within the same tenant only.

GET operation on the Mailbox returns properties of the mailbox (for example, quotas) as well as a link to the Folders element for the mailbox:

```
GET /vmrest/mailbox
<xs:complexType name="Mailbox">
<xs:all>
<xs:element name="DisplayName" type="xs:string" />
<xs:element name="CurrentSizeInBytes" type="xs:long" />
<xs:element name="IsPrimary" type="xs:boolean" />
<xs:element name="IsStoreMounted" type="xs:boolean" />
<xs:element name="IsStoreOverFlowed" type="xs:boolean" />
<xs:element name="IsMailboxMounted" type="xs:boolean" />
<xs:element name="IsWarningQuotaExceeded" type="xs:boolean" />
<xs:element name="IsReceiveQuotaExceeded" type="xs:boolean" />
<xs:element name="IsSendQuotaExceeded" type="xs:boolean" />
<xs:element name="WarningQuota" type="xs:long" />
<xs:element name="ReceiveQuota" type="xs:long" />
<xs:element name="SendQuota" type="xs:long" />
<xs:element name="IsDeletedFolderEnabled" type="xs:boolean" />
<xs:element name="FoldersURI" type="xs:anyURI" />
</xs:all>
</xs:complexType>
```

Mailbox Folder Operations

There are three folders currently supported on a Unity Connection Mailbox -

- Inbox
- Sent Items
- Deleted Items

Performing GET operation on the folders returns the fixed list of folders:

```
GET /vmrest/mailbox/folders
```

Following are the properties associated with each folder:

- DisplayName
- MessageCount
- Unique Serial Number (Unity Connection 11.5 and later)
- UIDValidity (Unity Connection 11.5 and later)

A GET operation on a folder returns the associated properties.

```
GET /vmrest/mailbox/folders/<folder_name>
```

Following is the response of above GET request.

```
<Folder>
<DisplayName></DisplayName>
<MessageCount></MessageCount>
<USN></USN>
<UIDValidity></UIDValidity>
</Folder>
```

Note: Each time an operation is performed on Inbox folder or Delete folder, the Unique Serial Number (USN) of both the folders changes.

Inbox Folder Operations

All folder operations can be executed by a user when connecting with his/her credentials. When using the administrative credentials, using the userobjectid parameter will allow administrators to do the same operations on the users mailbox. Unity Connection provides a field "Session Id" for Video Messages in the Inbox folder . This Session Id acts as a unique Identity number for Video Messages.

A GET operation on a folder returns a message list:

```
GET /vmrest/mailbox/folders/inbox/messages
```

Response:

```
<Messages total="1">
<Message>
<Subject>Message from user1 (1001)</Subject>
<Read>true</Read>
<Dispatch>>false</Dispatch>
<Secure>>false</Secure>
<Priority>Normal</Priority>
```

Cisco_Unity_Connection_Messaging_Interface_(CUMI)_API_--_Using_the_CUMI_API

```
<Sensitivity>Normal</Sensitivity>
<SessionId>122162627c12b61</SessionId>
<URI>/vmrest/messages/0:b4ffbaa5-19ad-4f4f-b95a-933a32e00d64</URI>
<MsgId>0:b4ffbaa5-19ad-4f4f-b95a-933a32e00d64</MsgId>
<From>
<DisplayName>user1</DisplayName>
<SmtpAddress>connectionserver</SmtpAddress>
<DtmfAccessId>1001</DtmfAccessId>
</From>
<CallerId>
<CallerNumber>1001</CallerNumber>
<CallerName>user1</CallerName>
</CallerId>
<ArrivalTime>1522069679000</ArrivalTime>
<Size>37291</Size>
<Duration>3340</Duration>
<IMAPUid>22</IMAPUid>
<FromSub>true</FromSub>
<MsgType>Voice</MsgType>
</Message>
</Messages>
```

A GET operation on messages in inbox folder returns a particular message by providing message Id:

```
GET/vmrest/messages/<messageid>
```

Response

```
<Message>
<Subject>Message from user1 (1001)</Subject>
<Read>true</Read>
<Dispatch>false</Dispatch>
<Secure>false</Secure>
<Priority>Normal</Priority>
<Sensitivity>Normal</Sensitivity>
<Attachments>
<Attachment>
<URI>/vmrest/messages/0:b4ffbaa5-19ad-4f4f-b95a-933a32e00d64/attachments/0</URI>
<contentType>audio/wav; name=voicemail.wav</contentType>
<contentTransferEncoding/>
<contentDisposition>inline; filename=VoiceMessage.wav; voice=Voice-Message</contentDisposition>
</Attachment>
</Attachments>
<SessionId>122162627c12b61</SessionId>
<Recipients>
<Recipient>
<Type>TO</Type>
<Address>
<DisplayName/>
<SmtpAddress>connectionserver</SmtpAddress>
<DtmfAccessId>1009</DtmfAccessId>
</Address>
</Recipient>
</Recipients>
<URI>/vmrest/messages/0:b4ffbaa5-19ad-4f4f-b95a-933a32e00d64</URI>
<MsgId>0:b4ffbaa5-19ad-4f4f-b95a-933a32e00d64</MsgId>
<From>
<DisplayName>user1</DisplayName>
<SmtpAddress>user1@ucbu-aricent-vm550.cisco.com</SmtpAddress>
<DtmfAccessId>1001</DtmfAccessId>
</From>
<CallerId>
<CallerNumber>1001</CallerNumber>
```

Cisco_Unity_Connection_Messaging_Interface_(CUMI)_API_--_Using_the_CUMI_API

```
<CallerName>user1</CallerName>
</CallerId>
<ArrivalTime>1522069679000</ArrivalTime>
<Size>37291</Size>
<Duration>3340</Duration>
<IMAPUid>22</IMAPUid>
<FromSub>true</FromSub>
<MsgType>Voice</MsgType>
</Message>
```

A user with administrative privileges can list the messages of another users folder by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/inbox/messages?userobjectid=<userobjectid>
```

A user with administrative privileges can list the messages of another users folder having USN value greater than specified and by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/inbox/messages?userobjectid=<userobjectid>&usngreaterthan=<value>
```

A PUT operation on messages in Inbox folder can update the Subject or the Read field of the Messages. No other parameter of a message can be changed:

```
PUT /vmrest/messages/<messageid>
<Message>
  <Subject>New subject</Subject>
</Message>
```

A DELETE operation on messages in Inbox folder can delete the message from the folder. Whether the message is soft or hard deleted is dependent on the settings of the system.

```
DELETE /vmrest/messages/<message-id>
```

A hard or soft delete can be forced by passing in the harddelete parameter:

```
DELETE /vmrest/messages/<message-id>?userobjectid=<userobjectid>&harddelete=<true or false>
```

Sent Items Folder Operations

All folder operations can be executed by a user when connecting with his/her credentials. When using the administrative credentials, using the userobjectid parameter will allow administrators to do the same operations on the users mailbox.

A GET operation on the folder returns a message list:

```
GET /vmrest/mailbox/folders/sent/messages
```

A user with administrative privileges can list the messages of another users folder by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/sent/messages?userobjectid=<userobjectid>
```

A user with administrative privileges can list the messages of another users folder having USN value greater than specified and by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/sent/messages?userobjectid=<userobjectid>&usngreaterthan=<value>
```

Cisco_Unity_Connection_Messaging_Interface_(CUMI)_API_--_Using_the_CUMI_API

A PUT operation on messages in the Sent Items folder can update the Subject of messages. No other parameter on a message can be changed:

```
PUT /vmrest/messages/<messageid>
<Message>
  <Subject>New subject</Subject>
</Message>
```

A DELETE operation on messages in the Sent Items folder can delete the message from the folder. Whether the message is soft or hard deleted is dependent on the settings of the system.

```
DELETE /vmrest/messages/<message-id>
```

A hard or soft delete can be forced by passing in the harddelete parameter:

```
DELETE /vmrest/messages/<message-id>?userobjectid=<userobjectid>&harddelete=<true or false>
```

Deleted Items Folder Operations

All folder operations can be executed by a user when connecting with his/her credentials. When using the administrative credentials, using the userobjectid parameter will allow administrators to do the same operations on the users mailbox.

A GET operation on the folder returns a message list:

```
GET /vmrest/mailbox/folders/deleted/messages
```

A user with administrative privileges can list the messages of another users folder by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/deleted/messages?userobjectid=<userobjectid>
```

A user with administrative privileges can list the messages of another users folder having USN value greater than specified and by passing the userobjectid of the other user:

```
GET /vmrest/mailbox/folders/deleted/messages?userobjectid=<userobjectid>&usngreaterthan=<value>
```

A PUT operation on messages in the Deleted Items folder to update the Subject on the messages. No other parameter on a message can be changed:

```
PUT /vmrest/messages/<messageid>
<Message>
  <Subject>New subject</Subject>
</Message>
```

A DELETE operation on messages in the Deleted Items folder will delete the message from the folder. This is a hard (permanent) delete.

```
DELETE /vmrest/messages/<message-id>
```

A POST operation on the Deleted Items folder can be used to empty the whole folder. The messages are hard deleted.

```
POST /vmrest/mailbox/folders/deleted/messages?method=empty
```

Offset and Limit

Each of the folders will accept the parameters "pagenumber" and "rowsperpage" to specify which messages to retrieve:

```
/vmrest/mailbox/folders/inbox/messages?pagenumber=1&rowsperpage=10
```

Sorting

Initially, server-side sorting will be limited to what can be done efficiently by the database, and will default to placing new messages first, followed by read messages, and sorted within each by ArrivalTime.

As recommended in the VTG REST guidelines, sorting will be controlled via "sortkey" and "sortorder" parameters, although initially only the following sort orders will be supported by the server:

Sort Description	Sort Parameters
Newest first	no parameters (default) or sortkey=arrivaltime&sortorder=descending
Oldest first	sortkey=arrivaltime&sortorder=ascending
Urgent first	sortkey=priority&sortorder=descending

Filtering

Filtering can be done on the folders by read, priority, dispatch, type, and USN of the message.

```
read={true|false}
priority={urgent|normal|low}
dispatch={true|false}
type={voice|fax|email|receipt}
usngreaterthan={Integer}
```

Examples

To get a list of unheard voice messages:

```
GET /vmrest/mailbox/folders/inbox/messages?read=false&type=voice
```

To get a list of unheard urgent messages:

```
GET /vmrest/mailbox/folders/inbox/messages?read=false&priority=urgent
```

To get a list of saved (deleted) messages:

```
GET /vmrest/mailbox/folders/deleted/messages
```

To get a list of messages with USN greater than 10:

```
GET /vmrest/mailbox/folders/inbox/messages?usngreaterthan=10
```

Back to: [CUMI API Overview](#)