

[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 the 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.

Doing a GET 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

## Cisco\_Unity\_Connection\_Messaging\_Interface\_(CUMI)\_API\_--\_Using\_the\_CUMI\_API

- Sent Items
- Deleted Items

Doing a GET on the Folders returns the fixed list of folders:

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

A Folder consists of a small set of properties (DisplayName and MessageCount) and a collection of Messages.

Doing a GET on a Folder returns a small folder resource object and a reference to its list of Messages:

```
<Folder>  
<DisplayName></DisplayName>  
<MessageCount></MessageCount>  
</Folder>
```

### 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.

A GET API call on the folder returns a message list:

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

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

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

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

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

A DELETE API call on Messages in the 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 API call on the folder returns a message list:

## Mailbox Folder Operations

## Cisco Unity Connection Messaging Interface (CUMI) API -- Using the CUMI API

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

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

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

A PUT API call on Messages in the Sent Items folder can 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 API call 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 API call on the folder returns a message list:

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

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

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

A PUT API call can be used 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 API call 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 API call 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
```

### Sent Items Folder Operations

## 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, voice, fax and dispatch.

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

## 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
```

---

**Back to: [CUMI API Overview](#)**