



A MITEL  
PRODUCT  
GUIDE

# MiVoice MX-ONE

## Call History REST API - Interface Description

Release 7.8  
66/15519-ANF90114 Uen C

November 2024

## Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks Corporation (MITEL<sup>®</sup>)**. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

## Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC), its affiliates, parents, or subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at [legal@mitel.com](mailto:legal@mitel.com) for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

<sup>®</sup>, <sup>™</sup> Trademark of Mitel Networks Corporation

© Copyright 2024, Mitel Networks Corporation

All rights reserved

# Contents

<b>1 Introduction.....</b>	<b>1</b>
<b>2 Overview, Reading Instructions, Limitation and Scope of the Document.....</b>	<b>2</b>
<b>3 Authentication And Security.....</b>	<b>3</b>
3.1 Authentication Token.....	3
3.1.1 Authentication.....	3
3.2 PBX IP Address.....	4
3.3 HTTPS Server Certificate.....	4
<b>4 Conventions and Terminology.....</b>	<b>5</b>
<b>5 Reference Architecture.....</b>	<b>6</b>
<b>6 Definitions.....</b>	<b>7</b>
6.1 Glossary.....	7
6.2 Acronyms.....	7
<b>7 Call History Log.....</b>	<b>9</b>
7.1 Local Call History Log.....	9
7.2 Central Call History Log.....	9
7.2.1 Enabling/Disabling Central Call Log at Logon.....	9
7.2.2 Creating Central Call Log (at Logon).....	10
7.2.3 Updating Central Call History Log.....	10
7.2.4 Deleting A Central Call History Log Entry.....	10
<b>8 Call History.....</b>	<b>11</b>
8.1 GET Call History for a Directory Number.....	13
8.2 POST Call History Reporting.....	15
8.3 DELETE Call History.....	16
8.4 GET API Specification.....	17

<b>9 Some Use Case Examples.....</b>	<b>18</b>
<b>10 SIP Ports Used.....</b>	<b>19</b>
<b>11 References.....</b>	<b>20</b>
11.1 Internal and CPI Documents.....	20
11.2 Standards, RFCs.....	20

This interface description API document describes the message flows and data content for the Central Call History function. The document relates the data exchange between the Mitel MiCollab infrastructure, primarily the MiCollab Server, and the MiVoice MX-ONE PBXs.

The purpose of the Central Call History feature is to provide every user of a SIP terminal/client with a log of calls received and placed, either answered or missed/failed.

By accessing the Call History Log feature, an end-user can:

- Browse the log
- Make calls to any stored number
- Delete entries that are no longer needed

The Call History Log feature comprises two different functions:

- Call History Log registration, which activates the central Call History Log feature for the extension.
  - The central Call History Log registration for SIP extension is done automatically after login or registration if the feature is enabled on the system. The extension will only accept call logs being pushed to the end-point if the MX-ONE has offered centralized Call History Log.
- Call History Log handling, which means accessing and/or manipulating the central Call History Log data.
  - The central Call History Log handling for SIP end-points is controlled by the terminal/client via keys or menus (depends on the end-point type/model). There are four types of logged calls; incoming, incoming-missed, outgoing, and outgoing-failed.

**Note:**

The system administrator can enable/disable this feature for a SIP terminal/client.

# Overview, Reading Instructions, Limitation and Scope of the Document

## 2

This Interface description presented in this document is not complete and does not show all functions in detail; rather it describes the more important ones with some examples.

MiCollab Server (MCS) uses a REST API interface to get Central Call History Log information from the PBX. The end-points might also have a local Call History function, but that must be turned off when the central Call History is active. Application credentials need to be configured in the PBX to allow MCS to collect Call History Log data for all users.

Authentication between PBX and MCS will use an authentication token. The token will be verified by the PBX and the MCS against the service.

This chapter contains the following sections:

- [Authentication Token](#)
- [PBX IP Address](#)
- [HTTPS Server Certificate](#)

## 3.1 Authentication Token

Authentication is provided by an authorization header.

Digest Authentication is proposed to be used for the authentication (same as for SIP and VDP). The recommended time interval for re-verification is proposed to be 30 minutes.

For more information, see [https://en.wikipedia.org/wiki/Digest\\_access\\_authentication](https://en.wikipedia.org/wiki/Digest_access_authentication), and the RFC 7616.

Example:

HTTP header

```
"Authorization" : " Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IiIsIm5hbWUiOiIgIiwidHlwZSI6IiVzZXIiLCJlbWFpbCI6IiIsInBob3RvVXJsIjoiiIiwiaXV0aFByb3Rpd-Q9KS2JhVYBm8 "
```

The MCS and PBX will try to verify this token for validity. For performance reasons, the token is cached for 30 minutes, before it is re-verified.

The call history can be accessed by authenticating against the user's login credentials, giving only access to the user's call history. With the command `user_authentication`, the administrator can configure a user to access all generic extension call history.

### 3.1.1 Authentication

```
securitySchemes:  
  
digestAuth:  
  
type: http scheme: digest  
description: Authenticate with phone number and phone password
```

## 3.2 PBX IP Address

A different approach could be to check whether the request is actually coming from the configured PBX IP address. The MCS has an active connection to the PBX that uses the same IP address. If token validation is not possible, we could use this approach. This is not the preferred approach.

## 3.3 HTTPS Server Certificate

HTTPS requires a server certificate to be installed on the PBX and the MCS. The same certificate must be uploaded to the MCS web portal and to the PBX portal. This enables the MCS to run HTTPS on port 22228 using the uploaded certificate. Only one certificate can be used for one specific port (22228) in the MCS.



# Conventions and Terminology

## 4

The following conventions and terminology will be used in this document: Data will be added using JSON formatted objects and GET, PUT, POST and DELETE methods.

This is the base URL that is implemented in the REST API: `https://<mxone-micollab>:22228/api/v1/mxoneCallHistoryApi/pbx/`

GET method must be used for reading data. POST method must be used for creating data. PUT method must be used for changing data. DELETE method must be used for deleting data.

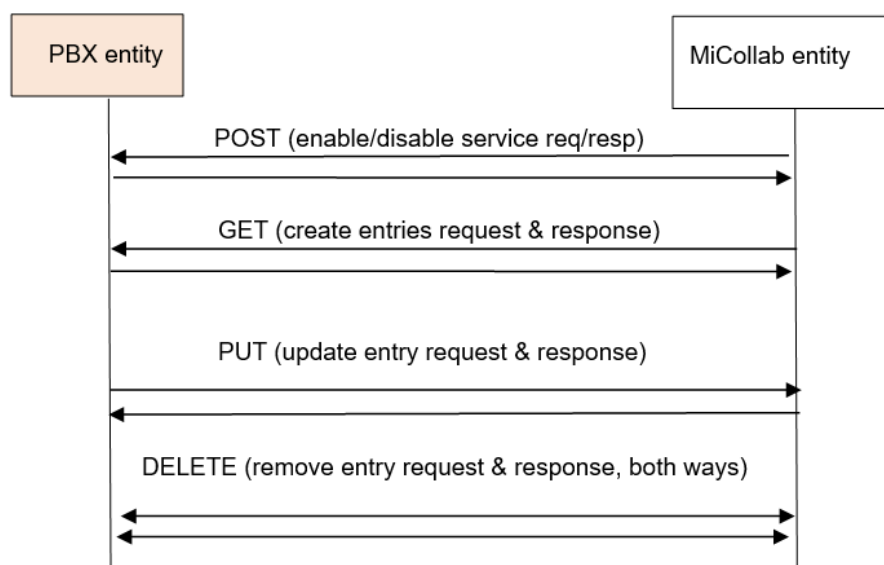
All requests will return a standard HTTP response, for example:

- 200 OK, the request was performed
- 400 Bad request; the data could not be parsed
- 401 Authentication failed (unauthorized)
- 403 Authentication rejected
- 404 User number not found
- 500 Internal Server Error
- 503 Service Unavailable

GET requests return additional data in the response body.

The messages shown in the following figure are used.

Figure 1: Central Call History Messages

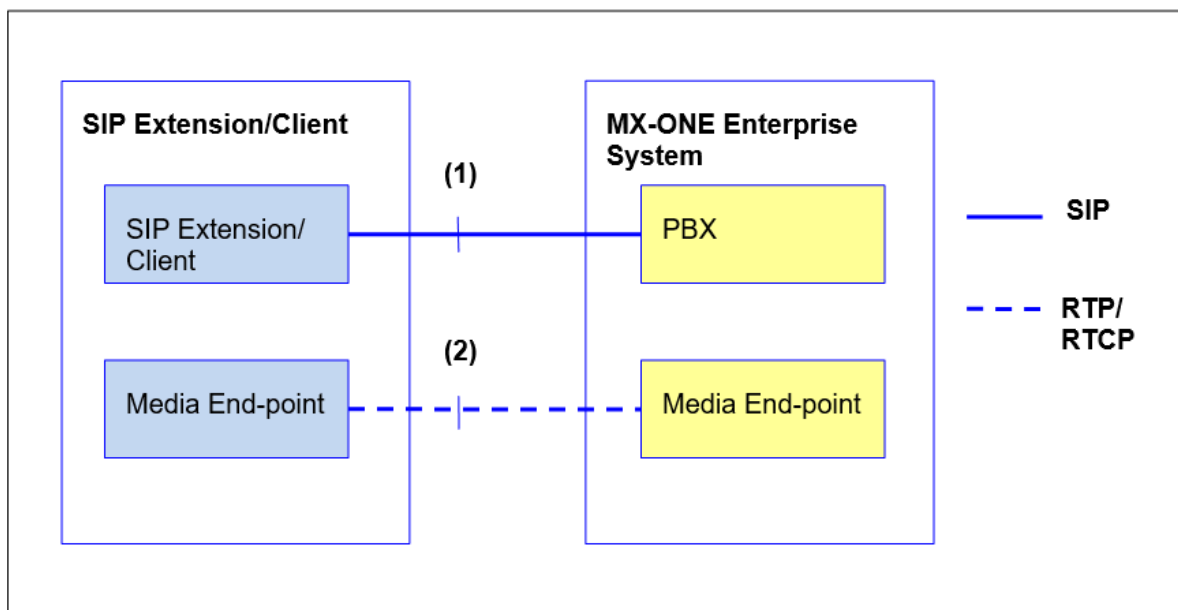


# Reference Architecture

## 5

The reference architecture diagram in the following figure shows the common functional elements required to support the interface specification outlined by this document. The figure shows two reference points between the Enterprise system and the SIP based extension/client; reference point (1) and reference point (2). The Enterprise system has a SIP Proxy Server/Registrar and media gateways. The Central Call History API is part of reference point (1), and conveys data for the Central Call History log for extensions/clients.

Figure 2: Reference Architecture



Note that a single SIP-PBX may serve media endpoints in a number of geographically-distributed locations. One user (directory number) may have multiple endpoints (terminals/clients) registered.

# Definitions

# 6

This chapter contains the following sections:

- [Glossary](#)
- [Acronyms](#)

## 6.1 Glossary

<b>SIP</b>	Session Initiation Protocol, IETF standards for packet-based multimedia communication systems. See RFC 3261.
<b>SIP Endpoint</b>	SIP-EP, a term used in this document to refer to both SIP terminals/clients and SIP-PBXs (the media gateway).
<b>SIP Extension</b>	The SIP extension feature allows terminals that are compliant with SIP standards, IETF RFCs, to register in and connect to the MX-ONE system. These standards give recommendations for multimedia communications over IP networks. The term "IP extension" includes both H.323 and SIP extensions. The SIP extension is implemented as a generic extension. The SIP extension can be either single line access (have only one active call), or multi-line access (have several active calls). Which access type is valid depends on the terminal brand/model, and the configuration.
<b>SIP-PBX</b>	The Enterprise's point of SIP signaling inter-connection with the SIP extension/client. Here the MX-ONE System.

## 6.2 Acronyms

<b>CSP</b>	Common Service Profile (in MX-ONE)
<b>EP</b>	End Point (a terminal or client)

<b>HTTP</b>	HyperText Transfer Protocol
<b>IWD</b>	InterWorking Description (Interface Description)
<b>MCS</b>	MiCollab Server
<b>ODN</b>	Own Directory Number (main line)
<b>PBX</b>	Private Branch Exchange (Enterprise system)
<b>RTP</b>	Real-Time Protocol
<b>SIP</b>	Session Initiation Protocol
<b>SRTP</b>	Secure Real-Time Protocol

# Call History Log

# 7

This chapter contains the following sections:

- [Local Call History Log](#)
- [Central Call History Log](#)

Call History Log is an end-user service, also known as "Call Log/Call List" and "Name and number log", which provides a list of calls received by or placed from the user's number. It might be valid for several EPs, if the user has multiple terminals/clients. The Call History Log is stored centrally, in the PBX, and sent to the endpoints for presentation.

The service gives the ability for an extension/client to log calls; for example received, placed and missed calls. What is actually shown to the end-user is up to the EP.

## 7.1 Local Call History Log

If the log is local in the terminal/client, it is stored in the terminal/client based on the ordinary number and name information used in display functions. Local log is the default configuration, but must be turned off if central Call History is active. Any discussion about the Local Call History Log is outside the scope of this document. See the client/terminal documentation for details.

## 7.2 Central Call History Log

MX-ONE has the following central Call History Log limits with regard to the support for SIP extensions (currently support for Mitel 6800/6900 SIP phones, using a proprietary XML protocol and SIP signaling):

- Maximum 60 incoming calls (calls made to the EP, where missed calls are a subset of all incoming calls).
- Maximum 30 outgoing calls (calls placed from the EP, where failed/rejected outgoing calls are a subset of all outgoing calls).

The same capacity is valid for this service (JSON-based Call History Log).

### 7.2.1 Enabling/Disabling Central Call Log at Logon

When SIP REGISTER is received for an extension EP, the PBX will provision the terminal/client to use the Call History API service if the request comes from a supported EP (for example MiCollab), and is requested for a standard main line ("ODN"). The user registering must have a CSP profile with Central Call History Log enabled.

The PBX will expect an HTTPS:POST (or HTTP:POST) to activate/enable the Call History Log service for the requesting user.

## 7.2.2 Creating Central Call Log (at Logon)

If the service is active and there are stored logs for the user, the stored logs are pushed via an HTTP: GET response, if a GET request is received by the PBX.

## 7.2.3 Updating Central Call History Log

When an extension that supports Central Call History Log receives or makes a call, the log must be updated; that is, the endpoint be informed of the new call.

When SIP INVITE is received/sent for an extension EP, the PBX will inform the extensions that support Central Call History Log receives or makes a call, the log is updated; that is, the endpoint be informed of the new call in the log.

## 7.2.4 Deleting A Central Call History Log Entry

When an extension that supports centralized Call History Log requests to erase one specific log entry, or erase all entries, manually or in certain use cases, that shall be possible. An example for all entries must be erased is at logoff or checkout for Hospitality users. The Delete method can be used both ways; sent from the PBX or sent from the MCS.

# Call History

## 8

This chapter contains the following sections:

- [GET Call History for a Directory Number](#)
- [POST Call History Reporting](#)
- [DELETE Call History](#)
- [GET API Specification](#)

The **Call History API** provides a comprehensive solution for managing and retrieving call records from a Private Branch Exchange (PBX) system. This API allows users to perform key operations such as retrieving detailed call logs, subscribing to call history notifications, and deleting call records for a specified directory number.

The call history API information are as follows:

**Table 1: Call History API Information**

Version	1.0.4
Base URLs	<code>http://{serviceName}:22227/v1.0</code> <code>https://{serviceName}:22228/v1.0</code>
Authentication	The API uses <b>Digest Authentication</b> to authenticate requests using a phone number and password.

## Components

The **CallItem** object contains details about individual call history entries.

**Table 2: CallItem Object**

Parameter	Supported Value	Description
directoryNumber	String	Indicates the unique identity of a callee. Supports maximum of 24 digits as an input.
name	String	Indicates the user name of the directory number that is UTF-8 encoded. Supports maximum of 40 characters as an input.

Parameter	Supported Value	Description
callIdentity	String	Indicates the unique identity of the call. Supports maximum of 24 digits as an input.
dateTime	String	Indicates the date and time of the call in ISO format (UTC).
timeZone	String	Indicates the ISO time zone. For example, GMT.
duration	String	Indicates the length of the call in Hour:Minute format.
typeOfCall	String	Indicates the call type. Allowed values: <i>incoming-answered</i> , <i>incoming-missed</i> , and <i>outgoing</i> .
transferredCall	Boolean	Indicates if the call was transferred.
divertedCall	Boolean	Indicates if the call was diverted.
firstDialledNumber	String	Indicates the number initially dialed (for example, in case of diversion).
remoteNumber	String	Indicates the public subscriber number (maximum 20 digits).
directoryNumber2	String	Indicates the extra number related to the call (for example, redirected-to number, maximum 24 digits).
name2	String	Indicates the user name associated with the second directory number.



Parameter	Supported Value	Description
infoText2	String	Indicates the information text or reason related to the second directory number.

## 8.1 GET Call History for a Directory Number

Retrieve the call history for a specific directory number, with optional query parameters for filtering by time range or call type.

### Sample Endpoint

```
GET /callHistory/{directoryNumber}
```

**Table 3: Endpoints**

Parameter Type	Parameter Name	Sample Values/ Format	Mandatory	Description
Path	directoryNumber	String	Yes	Indicates the directory number for which to retrieve the call history.
Query	startTime	YYYY-MM-DD HH:MM:SS	No	Indicates the start time for the call history in ISO format.
Query	stopTime	YYYY-MM-DD HH:MM:SS	No	Indicates the end time for the call history in ISO format.
Query	type	incoming- answered, incoming-missed	No	Indicates the call type.

Table 4: Responses

Response Type	Value	Description
Successful Response	200 OK	<p>Indicates if the call history is successfully retrieved.</p> <p>Sample response body.</p> <pre>{   "subscriber": "1234567890",   "callItems": [     {       "directoryNumber": "9876543210",       "name": "John Doe",       "callIdentity": "12345",       "dateTime": "2023-10-03 12:34:56",       "timeZone": "GMT",       "duration": "00:05:30",       "typeOfCall": "incoming-answered",       "transferredCall": false,       "divertedCall": false,       "firstDialledNumber": "1234567890",       "remoteNumber": "9876543210",       "directoryNumber2": "1122334455",       "name2": "Jane Smith",       "infoText2": "Forwarded"     }   ] }</pre>
Error Response	400 Bad Request	Parameter missing or incorrect.
	401 Unauthorized	Authentication required.
	403 Forbidden	Authentication rejected.
	404 Not Found	Directory number not found.
	500 Internal Server Error	Server error.
	503 Service Unavailable	Server is restarting or overloaded.

## 8.2 POST Call History Reporting

Enable or disable call history reporting for a directory number.

### Sample Endpoint

```
POST /callHistory/{directoryNumber}
```

**Table 5: Endpoints**

Parameter Type	Parameter Name	Sample Values/ Format	Mandatory	Description
Path	directoryNumber	String	Yes	Indicates the directory number to enable or disable call history reporting for.

**Table 6: Responses**

Response Type	Value	Description
Successful Response	200 OK	Reporting status updated successfully.
Error Response	400 Bad Request	Parameter missing or incorrect.
	401 Unauthorized	Authentication required.
	403 Forbidden	Authentication rejected.
	404 Not Found	Directory number not found.
	500 Internal Server Error	Server error.
	503 Service Unavailable	Server is restarting or overloaded.

## 8.3 DELETE Call History

Delete call history for a specified directory number. This can target specific call records by callIdentity or use special values to delete multiple records (for example, all history, all outgoing calls, and so on).

### Sample Endpoint

```
DELETE /callHistory
```

Table 7: Endpoints

Parameter Type	Parameter Name	Sample Values/ Format	Mandatory	Description
Query	directoryNumber	String	Yes	Indicates the directory number of the subscribed entity whose call history is to be deleted.
Query	callIdentity	String	Yes	<p>Indicates the unique call identity of the specific call record to delete. Alternatively, the following special values can be used:</p> <ul style="list-style-type: none"> <li>• <b>all</b>. Deletes all call history.</li> <li>• <b>all-outgoing</b>. Deletes all outgoing calls.</li> <li>• <b>all-incoming</b>. Deletes all incoming calls except missed ones.</li> <li>• <b>all-incoming-missed</b>. Deletes all incoming missed calls.</li> </ul>

**Table 8: Responses**

Response Type	Value	Description
Successful Response	200 OK	Call history successfully deleted.
Error Response	400 Bad Request	Parameter missing or incorrect.
	401 Unauthorized	Authentication required.
	403 Forbidden	Authentication rejected.
	404 Not Found	Directory number not found.
	500 Internal Server Error	Server error.
	503 Service Unavailable	Server is restarting or overloaded.

## 8.4 GET API Specification

Retrieve the specification of the Call History API.

**Sample Endpoint**

```
GET /callHistorySpec
```

**Table 9: Responses**

Response Type	Value	Description
Successful Response	200 OK	Indicates the specification retrieved successfully.

# Some Use Case Examples

## 9

Get all answered call for user `directoryNumber`, with password `VeryGoodPasswd` in service node `192.168.7.10`

```
curl --silent --request GET
--digest -u directoryNumber:VeryGoodPasswd http://192.168.7.10:22227/v1.0/
callHistory/directoryNumber?type=incoming-answered
```

Delete all missed call for user `directoryNumber`, with password `VeryGoodPasswd` in service node `192.168.7.10`

```
curl --silent --request DELETE
--digest -u directoryNumber:VeryGoodPasswd
http://192.168.7.10:22227/v1.0/callHistory/ directoryNumber?callIdentity=all-
incoming-missed
```

Request Service node to push updates in the Call History Log to FTP server with User and Password for user `directoryNumber`, with password `VeryGoodPasswd` in service node `192.168.7.10`

```
curl --request "POST" --digest -u
directoryNumber:VeryGoodPasswd 'http://192.168.7.10:22227/v1.0/callHistory/
directoryNumber'
--header 'Content-Type: application/json' -d '{
  "action": "startReporting", "callbackUrl":
  "https://4bc894cb-3be0-48df-912c-5194ac579a63.mock.pstmn.io/callHistory",
  "callbackUrl": "https://4bc894cb-3be0-48df-912c-5194ac579a63.mock.pstmn.io/
callHistory", "authenticationType":
  "digest", "user": "User", "password":
  "password" }'
```

# SIP Ports Used

10

Table 10: SIP Ports Used

Protocol	Port number	Comments
JSON for CHLog	22228	HTTPS, for Central Call History, secure
JSON for CHLog	22227	HTTP, for Central Call History, non-secure

# References

This chapter contains the following sections:

- [Internal and CPI Documents](#)
- [Standards, RFCs](#)

## 11.1 Internal and CPI Documents

- FS 'Name and Number Log', 449/15517-ANF 901 14 Uen (Centralized part, for SIP ext)
- IWD 'SIP extension interface', 64/15519-ANF 901 14 Uen
- MXO-4600, Jira ticket on Central Call History log for MiCollab in MX-ONE 7.4 SP1
- Name and Number Log, centralized Operational Directions, 38/154 31-ANF 901 14 Uen
- The extension\_profile command, parameter -ext-cnnlog, 201/19082-ANF 901 14 Uen

## 11.2 Standards, RFCs

- RFC 3261, Session Initiation Protocol
- RFC 7230-7232 & 7234, Hypertext Transfer Protocol - HTTP/1.1 (replaces RFC 2616)
- RFC 7235, Hypertext Transfer Protocol (HTTP/1.1): Authentication (replaces RFC 2616)
- RFC 7616, HTTP Digest Access Authentication (SHA-256)
- RFC 7617, The 'Basic' HTTP Authentication Scheme (replaces RFC 2617, MD5)



