



API for Conductor Interactions

Overview

MaestroConference has two sets of APIs: the first is "Conference Management" allowing customers to create, query, and modify conferences (with pins, scheduled conference times, etc.). The second is the "Conductor API" which moves callers into breakout groups, puts hands up and down, adjusts microphones and other "Conductor actions" during a live call.

This document describes the latter: the Conductor API interactions.

The client interface is broken up into essentially 3 type of interactions and a group of objects. The client submits commands, receives events and queries for information and can maintain its own representation of the object states. Command requests are submitted via POSTs using name equal value pairs to pass parameters. They will respond with at status (TRUE/FALSE) comma(,) and additional information, used only to pass back messages for FALSE results. The key idea to understand is that the command submitted will respond with a status that only indicates that the command was accepted not an outcome, a FALSE status will be followed by some message to identify the failure. The outcome if any is sent to the clients via events since all clients need to know about the state change. The events are received by the clients and require no feedback to the servers. The information queries are the more traditional RPC type calls that take a request and return data. Informational requests receive parameters using get query parameters and send data back as JSON or XML.

Example.

If GUI will want to mute a caller it follows the steps:

1. Gather information to assemble the mute request
2. Submit the request and handle failed submits, these indicate the command was not received, or other submitting related issue.
3. Server process the request and raises and event.
4. Event is sent to all UI Client.
5. UI reacts to mute state change and displays updated info.

All requests take a required parameter of conferenceUID to identify the conference or callUID to identify the call to act against, the security token of the given conference for authentication and optional type parameter of XML or JSON to define the format for the returned data. The URL to access these calls is http://myaccount.maestroconference.com/_conference/<function name>. Depending on the request made additional parameters may be required.

Objects.

Objects in general all have a UID property that identifies them. These objects are used to store information and state and can be retrieved using the get functions that take a UID as a parameter (see informational function section).

Audio

This contains information about the available audio clips for this conference

| Field | Type | Description |
|--------------|-------------|--------------------|
| UID | String | Unique ID |
| name | String | Clip name |
| value | String | Path |

Call.

This contains information about the call

| Field | Type | Description |
|-----------------|-------------|---|
| UID | String | Unique ID |
| name | String | Call name |
| actualStartTime | DateTime | Start time of call |
| actualEndTime | DateTime | End time of call |
| status | STATUS | ACTIVE/CLOSED |
| notes | Text | Notes on this call |
| conferenceUID | String | UID of the conference this call belongs to |
| socketXML | Int | Port number of XML based events for this conference |
| callers | UID[] | List of caller UIDs |
| breakouts | UID[] | List of breakout UIDs |
| data | String | Pass through data block |
| server | String | Conference Server |

| | | |
|---------------------|-----------------|--|
| serverConferenceUID | String | UID of the conference the servers uses |
| currentTime | DateTime | Current time on the server, used to calculate timer offset |
| talking | Boolean | Is the conference sending talking messages |
| greenroom | Boolean | Is the green room enable for this call |
| tags | CallTag[] | List of call tags |
| settings | RoleFlag[] | Active list of RoleFlags active during this call only, loads from conference when call is created. |
| custom | Text | Custom data not relevant to conference system |
| backgroundMusic | Boolean | Is background music enabled for this call |
| link | String | Last link pushed. |
| message | String | Last message pushed. |
| feedback | String | List of feedback messages |
| enableFeedback | Boolean | Display feedback window |
| enableBroadcast | Boolean | Display hands up |
| custom | Key/Value Pairs | List of key/value pairs |

CallTag

This is a tag related to the call.

| Field | Type | Description |
|-----------------|---------|---|
| ID | Integer | Unique ID |
| value | String | Tag value |
| conferenceTagId | Integer | Conference tag id if this tag comes from the conference scope |

Person.

This is a person data object.

| Field | Type | Description |
|-------|------|-------------|
|-------|------|-------------|

| | | |
|---------|-----------------|---|
| UID | String | Unique Identifier for object |
| name | String | Name of meeting |
| ANI | String | Caller ID |
| PIN | String | PIN number |
| role | ROLE | Role this person receives |
| email | String | Email the person used to register |
| notes | String | Notes associated with the person for the conference |
| custom1 | String | Custom field 1 |
| custom2 | String | Custom field 2 |
| tags | ConferenceTag[] | List of conference tags assigned to this person |
| custom | Key/Value pair | List of key/value pairs |

ConferenceTag.

This represents the tags associated with the conference scope.

| Field | Type | Description |
|--------------|-------------|--------------------|
| ID | Integer | Id for the tag |
| value | String | Tag of value |

Conference.

This represents the conference room and all the items related to it.

| Field | Type | Description |
|--------------------|-------------|---|
| UID | String | Unique Identifier for object |
| name | String | Name of meeting |
| scheduledStartTime | DateTime | Scheduled starting time of meeting |
| scheduledEndTime | DateTime | Scheduled ending time of meeting |
| customer | UID | UID of the customer this conference belongs |
| status | String | Status of conference (SCHED,ACTIVE, IDLE, CLOSED, CANCEL) |

| | | |
|---------------|------------------|---|
| data | String | Data payload for GUI, not used by application |
| persons | UID[] | List of person UIDs |
| audio | Audio[] | List of available audio clips |
| DID | String | Inbound number for the call |
| recording | Boolean | Is the call being recorded |
| notes | String | Notes accumulated for the conference |
| activeCallUID | UID | UID of the main active call if the conference is ACTIVE, being deprecated, please use mainCallUID |
| mainCallUID | UID | UID of the main call associated with this conference |
| calls | UID[] | List of the UID of all the active calls associated with this conference. |
| tags | ConferenceTags[] | List of tags in this conference |
| setting | RoleFlag[] | List of settings for each role |
| custom | Text | Custom data not relevant to conference system |
| largeCall | Boolean | Is this conference setup to support large (multiroom) calls. |
| placeCall | Boolean | Is this conference setup to support placing outbound calls. |
| custom | Key/Value pairs | List of key/value pairs |

Breakout.

This represents a group of callers that are grouped for sound

DateTime

| Field | Type | Description |
|---------|--------|-------------------------------------|
| UID | String | Unique ID |
| name | String | Name |
| callers | UID[] | List of caller UID in this breakout |

| | | |
|------------------|----------|---|
| attention | Boolean | breakout wants attention either because somebody in group wants attention or the group wants attention. |
| attentionTime | DateTime | Time attention was set or oldest time from callers |
| lastAssistedTime | DateTime | Time that an assistant or presenter left group. If assisted group set to 0. |
| assistedGroup | Boolean | If a presenter or assistant is present in the group |
| data | String | Data payload, not used by application |
| private | Boolean | Is this group private (not hearing mike) |

Caller.

This represents an actual call into the meeting

| Field | Type | Description |
|--------------|----------|--|
| UID | String | Unique ID |
| ANI | String | Caller phone number |
| PIN | String | 6 digit pin number used to enter conference |
| DID | String | Phone number call to enter conference |
| startTime | DateTime | Time call was received |
| endTime | DateTime | Time call as terminated |
| role | Role | Roll the call has in meeting |
| breakout | UID | Optional breakout UID if call is present in a breakout |
| breakoutTime | DateTime | Time this caller joined/left breakout |
| mic | Boolean | Has microphone |
| mute | Boolean | is muted |
| deaf | Boolean | is deaf |
| active | Boolean | Is call currently active / live |
| tag | String | Tag assigned to caller, defaults to ANI if blank |
| attention | int | Caller wants attention (0-5) |

| | | |
|---------------|----------------|--|
| attentionTime | DateTime | Time caller requested attention |
| data | String | Data payload, not used by application |
| notes | String | Notes associated with caller, pre-populated with the person notes. |
| custom1 | String | Custom1 field from the person record. |
| custom2 | String | Custom2 Field from the person Record. |
| volume_in | Number | Volume adjustment |
| volume_out | Number | Volume adjustment |
| energy | Number | Energy level |
| custom | Key/Value pair | List of key/value pairs |

Role.

Enum that represents the defined roles.

| Value | Description |
|-------------|-----------------------|
| PARTICIPANT | Call that is passive. |
| ASSISTANT | Assistant |
| PRESENTER | Presenter |
| OPERATOR | Operator |

RoleFlag.

List of flags for caller when they enter the call.

| Value | Description |
|-------|-----------------------------------|
| role | PARTICIPANT, PRESENTER, ASSISTANT |
| mike | Mike status |
| mute | Mute status |
| deaf | Deaf status |

Customer.

This represents an actual customer hosting the meeting

| Field | Type | Description |
|-----------------|----------------|---|
| UID | String | Unique ID |
| name | String | Name of the customer account. |
| status | String | Account status. |
| DID | String | DID assigned to account |
| recording | String | Flag indicating if the customer has recording enabled by default |
| greenroom | String | Flag indicating if the customer has greenroom enabled by default |
| contactEmails | String | Comma separated list of emails associated with this customer |
| backgroundMusic | String | Flag indicating if the customer has background music enabled by default |
| placeCall | String | Flag indicating if this customer has outbound calling enabled |
| custom | Key/Value pair | List of key/value pairs |

Information Requests.

This class of functions are used to retrieve information form the server.

getBreakout

This call is used to retrieve data about a breakout.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call to reference. |
| securityToken | Security token of the conference this call is associated with |
| UID | UID of the breakout. |
| type | (Optional, defaults to XML) XML or JSON, defines output format used for response. |

getCaller

This call is used to retrieve data about the caller.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call to reference. |
| securityToken | Security token of the conference this call is associated with |
| UID | UID of the caller. |
| type | (Optional, defaults to XML) XML or JSON, defines output format used for response. |

getConference

This call is used to retrieve data about the conference and related UIDs only. For full conference details including child objects use getConferenceData.

Parameter Value

| | |
|---------------|--|
| conferenceUID | UID of conference to reference. |
| securityToken | Security token of the conference this call is associated with |
| type | (Optional, defaults to XML) XML or JSON, defines output format used for response |

getConferenceData

This call is designed to retrieve all the data associated with the conference and all the child objects such as person.

Parameter Value

| | |
|---------------|--|
| conferenceUID | UID of conference to reference. |
| securityToken | Security token of the conference this call is associated with |
| type | (Optional, defaults to XML) XML or JSON, defines output format used for response |

getCall.

This call is designed to retrieve all the data associated with the call and all the child UIDs.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call to reference. |
| securityToken | Security token of the conference this call is associated with |

type (Optional, defaults to XML) XML or JSON, defines output format used for response

getCallData

This call is designed to retrieve all the data associated with the conference and all the child objects such as callers and breakouts.

Parameter Value

callUID UID of call to reference.

securityToken Security token of the conference this call is associated with

type (Optional, defaults to XML) XML or JSON, defines output format used for response

getCustomer

This call is used to retrieve data about the hosting customer only.

Parameter Value

conferenceUID UID of conference to reference.

securityToken Security token of the conference this call is associated with

type (Optional, defaults to XML) XML or JSON, defines output format used for response

Command Requests.

This group of requests initiate actions on the conference. They return a string TRUE or FALSE, <message> with <message> being a text string with some description of what failed.

createBreakout

This call is designed to create a breakout associated with the conference with the passed UID.

Parameter Value

callers Comma separated list of caller UIDs to group together

callUID UID of call to change

securityToken Security token of the conference this call is associated with
breakoutUID (Optional) Specify the UID of the breakout group.
name (Optional) Name to be assigned to breakout
private (Optional) Boolean value that sets the private flag on a group upon creation

createBreakout

This call is designed to create multiple breakouts associated with the conference with the passed UID. Each breakout is defined by caller<#> with an optional breakoutUID<#> parameter. For example to create 2 groups specify the following parameters
conferenceUID=<XXX>, count=2, callers1=c1,c2,c3,
callers2=f1,f2,f3,breakoutUID2=SALES

| Parameter | Value |
|------------------|--|
| callUID | UID of call to change |
| securityToken | Security token of the conference this call is associated with |
| count | Number of breakouts defined |
| callers<#> | List of callers for breakout #, comma separated |
| breakoutUID<#> | (Optional) Specify the UID of the breakout group for #. |
| name<#> | (Optional) Name to be assigned to breakout for # |
| private<#> | (Optional) Boolean value that sets the private flag on a group upon creation |

closeCall

This call will close the active call, kicking out all callers and updating call status and conference status.

| Parameter | Value |
|------------------|---|
| callUID | UID of call to close |
| securityToken | Security token of the conference this call is associated with |

updateCall

This call will update call information with supplied data.

| Parameter | Value |
|------------------|---|
| callUID | UID of call to update |
| securityToken | Security token of the conference this call is associated with |

| | |
|-------|---|
| field | Name of field to update (name,notes, note (appends text to end of notes field), talking, data, greenroom,entryway, custom, backgroundMusic,link,message). |
| value | Value to update field. |

updateCaller

This call will update caller information with supplied data. You can also operate on multiple callers at the same time by supplying a comma separated list of UIDs in the UID parameter. The action will be performed on all the UIDs and any failure will result in the call returning a failure. It will attempt to perform the update on all the UIDs supplied. This is not transactional so you need to handle partial failure manually.

Parameter Value

| | |
|---------------|--|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| UID | UID of the caller to act upon or a comma separated list of UIDs |
| field | Name of field to update (mike,deaf,mute,tag,attention,breakout,data,role, notes, note (appends text to end of notes field),active (if set any value will drop caller), custom1, custom2,volume_in, volume_out,energy). |
| value | Value to update field. |

updateActiveCaller

This call will update all active callers information with supplied data. The action will be performed on all active callers and any failure will result in the call returning a failure. This is not transactional so you need to handle partial failure manually. If the role parameters is supplied then it will validate the role and only update callers that match this role.

Parameter Value

| | |
|---------------|--|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| field | Name of field to update (mike,deaf,mute,tag,attention,breakout,data,role, notes, note (appends text to end of notes field),active (if set any value will drop caller), custom1, custom2,volume_in, volume_out,energy). |
| value | Value to update field. |
| role | (Optional)If role is supplied then only callers that match the role will be updated. |

dropCaller

This call will terminate caller and hang up on them.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| UID | UID of the caller to drop |

playTone

This call will terminate caller and hang up on them.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| tone | UID of the audio record to play |

updateBreakout

This call will update caller information with supplied data.

Parameter Value

| | |
|---------------|--|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| UID | UID of the breakout to act upon |
| field | Name of field to update (name,attention,callers(comma separated list of callers to action, remove them if present, add them otherwise), data,private). |
| value | Value to update field. |

updateConferenceSetting

This call will update the callers conference settings for a role when the join a call

Parameter Value

| | |
|---------------|--|
| conferenceUID | UID of conference |
| securityToken | Security token of the conference |
| role | Role of setting to change (PARTICIPANT, ASSISTANT, PRESENTER). |
| setting | Name of setting to change (mike,mute,deaf). |

value Value to update field (Boolean)

updateCallSetting

This call will update the callers settings for a role when the join a call. Does not alter the default at the conference level.

Parameter Value

callUID UID of call
securityToken Security token of the conference this call is associated with
role Role of setting to change (PARTICIPANT, ASSISTANT, PRESENTER).
setting Name of setting to change (mike,mute,deaf).
value Value to update field (Boolean)

updateCallerTag.

This call will relate a call tag with a call if the action is add, or remove the relation of the action is remove.

ParameterValue

callUID UID of call
callerUID UID of the caller, support comma separated list of UID for multiple updates.
callTagID Id of the call tag
action **Add**, create relation. **Remove**, remove relation.

createCallTag.

This call will relate a call tag with the supplied conference tag id or a new call scoped tag with the value specified. Only conferenceTagID or value should be supplied.

Parameter Value

callUID UID of call
securityToken Security token of the conference this call is associated with
conferenceTagID ID of the conference tag you want added to the call
value Value new call tag should contain

updateCallTag.

This call updates the value of the call tag id.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| callTagID | ID of the call tag to modify |
| value | New value to assign to call tag |

placeCall.

This attempts to place an outbound call and connect it to this call.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| name | (Optional) Name of caller to display |
| role | Role this caller is to assume when placed into the call |
| number | Phone number to call (US numbers must be prefixed with 1) |

updateCustomValue.

This create/updates a key/value pair of data associated with the current call. It will only update the parent conference, the current call, person records associated with this call and caller record of this call. Use updateCustomValue from the conference management API to update any other records.

Parameter Value

| | |
|---------------|---|
| callUID | UID of call |
| securityToken | Security token of the conference this call is associated with |
| objectType | (customer/conference/call/person/caller) Object type |
| objectUID | UID of the object to update |
| customKey | Key of the value |
| customValue | Value to associate with the key |

Event.

Event messages are data items that have caused data to change in some way. They are to notify you that the system has changed something about the supplied object in some way. These changes can originate from the GUI, the system or the caller themselves. The format for the messages is defined below. The second table below identifies the source/name combination in more detail as to when or why the event gets triggered.

Field Type Description

sourceString What object triggered the change
name String Name of event (see list below for full description)
target Object What the current state of the changed object is

| Source | Name | Description |
|---------------|--------------|--|
| CALL | NEWCALLER | Message is sent when the system received a new call into the conference, the target is the details associated with the new caller. |
| CALL | UPDATE | Call data was updated |
| CALL | NEWBREEAKOUT | Call created a new breakout |
| CALLER | UPDATE | Caller data was updated |
| CALLER | ATTENTION | Caller wants attention |
| CALLER | TALKING | Update caller talking status |
| CALLER | BREAKOUT | Caller changed braekout |
| CALLER | DEAF | Caller changed deaf state |
| CALLER | MIKE | Caller changed mike state |
| CALLER | MUTE | Caller changed mute state |
| CONFERENCE | NEWBREAKOUT | Message is sent once a breakout group is established |
| BREAKOUT | UPDATE | Breakout data has changed |
| TAG | UPDATE | New tag or updated tag for the call |
| TAG | MEMBERSHIP | Contains callerUID, tag id and membership status. |

Rev: 10-03-14