



# MaestroConference API for Conference Management

## 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 former: Conference Management interactions.

The API calls are all similar in format and include account level actions such as conference creation, conference modification including adding pins or schedule sessions, and call reporting/ gathering information about an existing call. Informational requests receive parameters using get query parameters and send data back as JSON or XML, modification requests are posts and can also receive data back as XML or JSON. The URL for both post and get requests is [http://myaccount.maestroconference.com/\\_access/<function name>](http://myaccount.maestroconference.com/_access/<function name>)

The data returned in a call contains code ( 0 = success, non zero represents an error message) and if successful a value (the result of the function call), and if error condition usually a message (Error text).

JSON data objects contain:

```
{"message":"","value":{"... function result ..},"code":"0"}
```

For XML formatted results it appears as:

```
<document>
  <value>
    < .. function result ..>
  </value>
  <code>0</code>
  <message></message>
</document>
```

**Below** is information about the format, actual API calls, and some sample PHP code for the "AddPerson" call, used to register someone with name & email for a conference.

## **Error Codes.**

Error codes can be anything other than 0, an error code of 0 indicates success or no error. General system errors are identified by number less than 0, function specific error codes are positive numbers. For function specific error see the function details, below are a list of general system error and what causes them.

Code	Reason
-1	General Error not categorized else where, read message for more details.
-2	Parameter error, one of the supplied parameters is invalid. The message will provide more detail as to the actual failure
-3	Servlet error, usually due to communications error. Read message for more details, call can probably be retried.
-4	Unhandled Exception. Somethihng the processing of this call failed in an unplanned process. If this persist please contact customer support.

## **About Email Notifications**

When someone registers with the self-registration URL, the system sends an immediate email to the email address with the registration information. When registering someone with the API, the system does not send such a notification. This allows the customer to customize the email as needed. Of course, the server accessing the API code can send such an email (via it's own SMTP server, or SendGrid, Amazon SES etc.)

There's also a conference setting on whether reminders are sent before the conference (24 and 2 hours currently). That reminder can be turned on and off, and will be sent the registered email addresses whether or not they came in via the API or the self-registration URL.

## **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).

## **Call.**

This contains information about a call

<b>Field</b>	<b>Type</b>	<b>Description</b>
--------------	-------------	--------------------

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
custom	CustomValue[]	List of custom values (key,value) pairs

## 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
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	Custom field from the person record.
custom2	String	Second custom field for the caller
personUID	String	UID of the person object associated with this caller

## CallTag.

This is a tag related to a 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

## 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 active call if the conference is ACTIVE
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
reservations	UID[]	List of conference reservation UID
reserved	Boolean	Is this a reserved conference
backgroundMusic	Boolean	Is background music to be played at the start of the call
custom	CustomValue[]	List of custom values (key,value)
securityToken	String	Security token associated with this conference
calls	String[]	List of calls associated with this conference, if conference is active only active call UIDs are listed. For conferences that are not active all call UIDs are listed.

### **ConferenceTag.**

This represents the tags associated with the conference scope.

<b>Field</b>	<b>Type</b>	<b>Description</b>
ID	Integer	Id for the tag
value	String	Tag of value

### **Customer.**

This object contains customer data.

<b>Field</b>	<b>Type</b>	<b>Description</b>
ID	Integer	Unique ID
name	String	Name
status	String	Status of the customer (ACTIVE, SUSPEND, CLOSED, HOLD)
DID	String	Inbound number assigned to this customer
recording	Boolean	Default state of recording flag when conference is created
greenroom	Boolean	Default state of the greenroom flag when conference is created.

contactEmails	String	Comma separated list of email used as the default for the conference when it is created.
backgroundMusic	Boolean	Is background music turned on by default for new conferences
placeCall	Boolean	Is this customer allowed to place outbound calls
inboundList	InboundAccess[]	A list of inbound access objects that are supported for this customer
custom	CustomValue[]	List of custom values associated with the customer

## 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	CustomValue[]	List of custom values (key, value) pairs
inboundAccess	Int	ID of the inbound access object assigned to this person
swAutoLoginUrl	String	A url to automatically login this user in Social Webinar when he has already dialed in to the call.

## InboundAccess.

This is a object that contains information about the difference inbound access methods.

Field	Type	Description
ID	Int	Unique Identifier for object
name	String	Name of access method
carrierType	String	Type of carrier
carrierID	String	The id assigned to this method

by the carrier

## **Reservation.**

This is a reservation record for a conference

<b>Field</b>	<b>Type</b>	<b>Description</b>
UID	String	Unique ID
conferenceUID	String	UID of the conference this reservation is associated with
estimateCaller	Integer	Estimated number of callers for this reservation
startTime	Date	Start time and date of the reservation
endTime	Date	End time of the reservation
duration	Integer	Number of minutes the reservation is for

## **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

## ***General Request***

All requests take a required parameters of customer and key to identify the customer and the key for authentication. Additional parameters are defined by the function being called. An optional type parameter of XML or JSON to define the format for the returned data, the

default is XML.

Standard parameters to all requests

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
customer	String	UID of the customer
key	String	API key defined for this customer

## ***Information Requests.***

This class of functions are used to retrieve information from the server and are access via http get requests.

### ***getCustomer***

This call is used to retrieve data about the customer. Returns customer object.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getActiveConference***

This call is used to retrieve a list of conferences that are currently active. Returns list of conference objects.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getUpcomingConference***

This call is used to retrieve a list of conferences that are upcoming. Returns list of conference objects.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getPossibleConference***

This call is used to retrieve a list of conferences that are currently possible to start. Returns list of conference objects.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getExpiredConference***

This call is used to retrieve a list of conferences that have expired. Returns list of conference objects.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getCanceledConference***

This call is used to retrieve a list of conferences that were canceled. Returns list of conference objects.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
<None>		

### ***getConference***

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

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
conferenceUID	String	Unique Identifier of the conference

### ***getConferenceData***

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

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
conferenceUID	String	Unique Identifier of the conference

### ***getConferenceReservation***

This call is designed to retrieve the data associated with the conference reservation.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
reservationUID	String	Unique Identifier of the conference reservation

### ***getCall***

This call is designed to retrieve information associated with a specific call. Will not retrieve data for active call. Use the conductor API for live calls.

<b>Parameter</b>	<b>Type</b>	<b>Description</b>
------------------	-------------	--------------------

conferenceUID	String	Unique Identifier of the conference
callUID	String	Unique Identifier of the call

## getCallData

This call is designed to retrieve information associated with a specific call. Will not retrieve data for active call. Use the conductor API for live calls. This will return a list of callers and details about the call.

Parameter	Type	Description
conferenceUID	String	Unique Identifier of the conference
callUID	String	Unique Identifier of the call

## Command Requests.

This group of requests initiate actions on the system. They are POST urls that require the parameters to be sent via http post.

### updateCustomer

This call is designed to update customer editable fields. Returns an updated customer object.

Parameter	Type	Description
field	String	Field to update (contactEmail, greenroom, recording, affiliateCode, reminder, backgroundMusic, preCall, postCall, banner, callerInterface, regComment)
value	String	Value to be assigned to field

### createConferenceReservationless

This call creates a reservationless conference with the supplied parameters. It returns a conference object.

Parameter	Type	Description
name	String	Name of conference (Optional)If supplied it overrides the default from the customer record. If missing then the customer contactEmail list is used.
contactEmail	String	(Optional)If supplied it overrides the customer default
greenroom	Boolean	(Optional)If supplied it overrides the customer default
recording	Boolean	(Optional)If supplied it overrides the customer default

estimatedCallers	Numeric	Estimated number of callers for this conference
------------------	---------	---

## createConferenceReserved

This call creates a reservation based conference with the supplied parameters. It returns a conference object. The parameters allow for variable number of reservations to be added during creation by specifying separate estimatedCallers, startDate and duration for each reservation. The conference will be created if at least one of the supplied reservation items are valid. If none of the reservation items are valid the call will error with code 1.

Parameter name	Type	Description
name	String	Name of conference (Optional)If supplied it overrides the default
contactEmail	String	from the customer record. If missing then the customer contactEmail list is used. (Optional)If supplied it overrides the customer default
greenroom	Boolean	(Optional)If supplied it overrides the customer default
recording	Boolean	(Optional)If supplied it overrides the customer default
backgroundMusic	Boolean	(Optional)If supplied it overrides the customer default
reservationCount	Numeric	Number of reservation for this conference. Must be at least 1
estimatedCallers<#>	Numeric	Estimated number of callers for this reservation
startDate<#>	DateTime	Starting date and time for this reservation(yyyy.MM.dd HH:mm:ss)
duration<#>	Numeric	Number of minutes this reservation is for, it will be rounded up to next 15 minute increment.

### Additional Error Codes

Code	Reason
1	No valid reservation was found. None of the specified reservation items we able to be used to create the conference.

## addConferenceReservation

This call creates a reservation for an existing conference with the supplied parameters. It returns a conference schedule object.

Parameter	Type	Description
conference	String	UID of existing conference
estimatedCallers	Numeric	Estimated number of callers for this reservation
startDate	DateTime	Starting date and time for this reservation (yyyy.MM.dd HH:mm:ss)
duration	Numeric	Number of minutes this reservation is for, it

will be rounded up to next 15 minute increment.

### Additional Error Codes

Code	Reason
1	Reservation start time over lapse with another.
2	Reservation duration over lapse with another.
3	Estimated caller size in invalid
4	Cannot add conference reservation to reservationless conference

### addPerson

This function creates a new person in the conference with the given data, only role is required other fields are optional. It returns a person object.

Parameter	Type	Description
conference	String	UID of existing conference
role	String	Role to assign person, must be one of valid ROLE types
name	String	(Optional) Name to assign this person.
email	String	(Optional) Email of the person.
notes	String	(Optional) Notes for this person.
custom1	String	(Optional) Custom1 field data.
custom2	String	(Optional) Custom2 field data.

### updatePerson

This function updates a person with the given data based on the UID. It returns a person object.

Parameter	Type	Description
person	String	UID of the person to apply updates against.
field	String	Name of the field to update (name, email, notes, custom1, custom2, role)
value	String	New value to assign to field

### removePerson

This function removes a person in the conference with the given UID.

Parameter	Type	Description
conference	String	UID of existing conference
person	String	UID of the person to remove.

## updateConferenceReservation

This function edits a reservation with the passed in parameters. It returns an updated reservation object.

Parameter	Type	Description
reservationUID	String	UID of existing conference reservation (Optional) New start time for this
startDate	DateTime	reservation. Invalid for reservationless conference.
estimatedCallers	Integer	(Optional) Estimated number of callers.
duration	Integer	(Optional) Reservation length. Invalid for reservationless conference.

### Additional Error Codes

Code	Reason
1	Reservation start time over lapse with another.
2	Reservation duration over lapse with another.
3	Estimated caller size is invalid
4	Cannot edit start date or duration of reservationless conference

## removeConferenceReservation

This call is designed to remove a conference reservation. Note you cannot remove a past reservation or the last reservation on a conference.

Parameter	Type	Description
reservationUID	String	Unique Identifier of the conference reservation

### Additional Error Codes

Code	Reason
1	Conference is reservationless
2	Last reservation of conference, cannot remove.
3	Reservation is already past.
4	Reservation is currently active.

## cancelConference

This call is designed to cancel a conference. It will return a conference object for the canceled conference.

Parameter	Type	Description
-----------	------	-------------

conference	String	Unique Identifier of existing conference to cancel
------------	--------	--

### Additional Error Codes

Code	Reason
1	Conference does not exist or is invalid

### updateConference

This call is designed to update conference editable fields. Returns an updated conference object.

Parameter	Type	Description
field	String	Field to update (contactEmail,registerText,notes,recording,greenroom,reminder,backgroundMusic,callerInterface,preCall,postCall,banner,regComment)
value	String	Value to be assigned to field

### Additional Error Codes

Code	Reason
1	Conference does not exist or is invalid
2	Field is not valid

### updateCustomValue

This call is designed to update custom values associated with Customer, Conference, Call, Caller or person. To remove a custom value just assign it an empty value. You cannot update call or callers that are currently active.

Parameter	Type	Description
objectType	String	Type of the object (conference, person, call, caller, customer)
objectUID	String	UID of the object to associate the value
customKey	String	Key to associate the value with
customValue	String	Value to associate to the object's given key

### Maestroconference recognized custom values

Maestroconference's conference administration site supports the use of the following custom

values. The tables are broken up by the customer object that are assigned these values.

#### Conference

<b>Custom Key</b>	<b>Value</b>
MC_CONFIRMATION_PAGE	URL of the confirmation page to redirect the user to when they use the self registration URL, if a confirmation query parameter is provided to the self registration URL it will over ride this value. If this value is missing or set to blank it will use the default confirmation page.
RESELLER_INFO_TEXT	The text to display in the reseller information button on the conductor screen. Both RESELLER_INFO_TEXT and RESELLER_INFO_URL need to be set for the button to show up.
RESELLER_INFO_URL	The URL to display on the browser in a new window when the button is clicked. Both RESELLER_INFO_TEXT and RESELLER_INFO_URL need to be set for the button to show up.

#### Call

<b>Custom Key</b>	<b>Value</b>
RESELLER_INFO_TEXT	The text to display in the reseller information button on the conductor screen. Both RESELLER_INFO_TEXT and RESELLER_INFO_URL need to be set for the button to show up.
RESELLER_INFO_URL	The URL to display on the browser in a new window when the button is clicked. Both RESELLER_INFO_TEXT and RESELLER_INFO_URL need to be set for the button to show up.

## Sample API call - PHP

For: AddPerson - Used to register a participant for a conference and get the PIN.

```
<?php
function AddPerson( $ConferenceId, $Role, $Name, $Email, $Notes )
{
    $MCBaseUrl = "http://myaccount.maestroconference.com/_access/";

    //Retrieve Maestro Conference CustomerUID and Security Token
    $CustId     = ""; //Add your CustomerUID here
    $SecurityToken = ""; // Add the 32 character Security Toker here - you can find it on
your MyAccount Page under API Settings

    $ch = curl_init();
    $Url = $MCBaseUrl . "addPerson"; // Add Person Call
    echo $Url . "<br />";

    //Create POST Array for Post Variables
    $PostFields["customer"] = $CustId;
    $PostFields["key"]      = $SecurityToken;
    $PostFields["conference"] = $ConferenceId;
    $PostFields["role"]     = $Role;
    $PostFields["name"]     = $Name;
    $PostFields["email"]    = $Email;
    $PostFields["notes"]    = $Notes;
    $PostFields["type"]     = "json";

    echo "<pre>";
    print_r($PostFields);
    echo "</pre>";

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_URL,$Url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($PostFields));
    $Response = curl_exec($ch);
    curl_close ($ch);

    //Decode the Reponse to JSON Array
    $Response = json_decode($Response, true);
    echo "<pre>";
    print_r($Response);
    echo "</pre>";
    //If $Response['code']==0 means no error -
```

```
if( $Response['code'] == 0 ) {  
  
    //Return PIN code returned by Maestro Conference API Call  
    return $Response['value']['PIN'];  
}  
//In case of error just return a NULL  
return null;  
}  
  
// Fill in the required information in the AddPerson function.  
AddPerson( $ConferenceId, $Role, $Name, $Email, $Notes )
```

?>