MEETS SaaS API Usage Manual

Table of Contents

WIP Note

This document is currently a work in progress and may be missing information which will be made available in future versions of this document.

Overview

This API for CirQlive MEETS SaaS allows for the creation of applications to gather information to generate custom reports as well as automate various needed activities. Users with the appropriate permissions on a MEETS SaaS platform can generate access credentials to use the APIs within custom applications, whether those custom applications are developed in-house or provided by third parties. This document will cover functions available via these APIs. Please refer to the MEETS SaaS Authentication Manual for information on which authentication methods exists and how to utilize them.

Assistance

If you need API assistance and guidance please contact CirQlive API assistance. Please note that assistance is only supplied to engineers of record from organizations that have a MEETS SaaS contract which includes an API Assistance package. If your organization wishes to purchase an API Assistance package, please contact your MEETS SaaS account manager or CirQlive Sales. If you are an individual user or third party developer looking to develop your own MEETS-based application or services, please contact CirQlive Sales for your own Developer package.

Usage

MEETS SaaS APIs use HTTP-based endpoints which allow for a variety of different authentication methods, please refer to the MEETS SaaS API Authentication Manual for information regarding authentication. The functions themselves are designed with consistency and reusability in mind, and are therefore grouped into several categories indicative of the function's purpose, use, and behavior. Functions bearing the same category as one another will also utilize the same HTTP method and output format, making it possible to create per-category connection and output handling code which is usable for all functions within that category. Likewise, the output format to report that an error has occurred during the execution of a function will be uniform across all categories.

MEETS APIs take their input parameters from either GET or POST queries. Values to parameters are required to use URL percent encoding according to RFC 3986. When passing input parameters via POST, the parameters are required to be in a query string, and be passed along with a *Content-Type* of *application/x-www-form-urlencoded*. Some categories (such as "upload" and "download") may also allow or require certain additional HTTP headers to be set by the client application. All but a handful of responses are formatted as JSON, and those that aren’t are easily detectable and fully documented below.

Input limitations

MEETS SaaS sets limits upon the size values can be when used as parameters for input. If a limit is violated, the parameter will be ignored.

- GET — 512 bytes.
- POST — 4096 bytes.

In typical cases, it is unlikely to run into a limit. However, some functions allow multiple comma separated values to be sent as a single parameter value. Care must be taken to not request too much at a time.

UNIX Timestamps

UNIX Timestamps are used throughout these APIs wherever a specific point in time needs to be represented. UNIX Timestamps are integers containing the amount of seconds since midnight on 1970-01-01 UTC (known as the UNIX epoch). Common mistakes regarding time include specifying a value which is not an integer, is using a quantity other than seconds, is using a time zone other than UTC, or is offset from a different epoch.

Most libraries include functionality to convert between integer-based timestamps and human readable text. If your library uses a different epoch or timezone other than UTC, you will need to offset the values appropriately. If your library uses a quantity other than seconds (such as milliseconds or microseconds) you will need to multiply or divide in order to turn one quantity into the other. If you are using a language where dividing two integers results in a decimal, you will need to truncate the decimal data and convert back to an integer.

There are some useful websites to aid working with and understanding UNIX Timestamps.

Understanding MEETS SaaS Organization and Architecture

MEETS SaaS is provided to each client on a unique domain, typically at *<client>.meets.cirqlive.com*. MEETS SaaS integrates with multiple third party platforms, and allows platforms to federate their users into MEETS. MEETS at the present supports federation via LTI — Learning Tools Interoperability, LDAP — Lightweight

Directory Access Protocol, and SAML — Security Assertion Markup Language. Specifics regarding how to federate MEETS via LTI into a platform can be found in the MEETS SaaS LTI Launch Manual. Specifics for LDAP and SAML can be found via the Federated Authentication Overview Manual.

Since clients often have multiple third party platforms they'd like to federate into MEETS, MEETS provides the capability to create separate platform connections for each one. Each platform connection isolates its data from the others, in order to ensure there is no overlap between external identifiers used for federation, and to keep data well organized.

MEETS SaaS allows API tokens to be created to pull data from an entire MEETS SaaS domain. If you are attempting to correlate data with a specific platform that federates with MEETS, please be sure to filter that data by the platform connection it uses. In the MEETS SaaS Administration Panel, each platform connection has a "key" which is the numeric ID the API uses to refer to a connection as "meetsIdConnection". The data returned by MEETS specifies relevant identifiers from platforms that federate to it. If you need to search for data via these external identifiers, you must first use a "from" or "map" function to convert external identifiers to the identifiers MEETS uses internally.

Please be advised that due to tradeoffs inherent within the CAP theorem, various components within MEETS SaaS are built upon the eventual consistency model. Further, many of the conferencing services and other third party platforms that MEETS integrates with are the same and provide APIs that are designed specifically for an eventually-consistent model. As a result of both of these considerations, the details offered in MEETS SaaS and via its APIs are also eventually-consistent.

Supported Services

MEETS SaaS can be used to integrate with a number of different kinds of services and platforms. These include user and learning management systems, conferencing services, chat and presence services, and storage services. Each service is abstracted into one set of functions that work across all services of its particular type. Using these APIs to work with a particular service should generally be compatible using the same functionality with another service of its kind. However there are still a few details to be aware of:

- Not all services necessarily support certain functionality that other services of the same type support. While the base functionality is the same across services of a particular type, attempting to get unsupported information will return an empty set, or trying to perform an unsupported operation will result in failure.
- Results which supply or link to external data could behave differently.
- Some services may have usage limitations that the others do not.

For these reasons, the MEETS SaaS APIs which provide lists of data indicate which service is used, both for informational purposes, and so that operations based on such data can be adjusted accordingly.

User / Learning / Training Platforms

MEETS SaaS supports federation into it by any platform that supports LTI — Learning Tools Interoperability. Specifics regarding how to federate MEETS via LTI into a platform can be found in the MEETS SaaS LTI Launch Manual. For non-LTI-capable platforms MEETS SaaS supports federation via LDAP — Lightweight Directory Access Protocol, and SAML — Security Assertion Markup Language. Specifics for LDAP and SAML can be found via the Federated Authentication Overview Manual

API functions which indicate the platform type used for a *connection* to MEETS will be one of the following:

- BlackboardLearn — Blackboard Learn
- Canvas — Instructure Canvas
- Moodle — Moodle
- Totara — Totara
- Desire2Learn — Brightspace by D2L, formerly Desire 2 Learn
- Other — Any other service

These other services while not specifically identified above, support LTI natively, and CirQlive is aware of clients that use these platforms to access MEETS:

- Agilix Buzz, formerly BrainHoney
- EDUonGo
- Jenzabar eLearning
- Pearson eCollege
- PowerSchool Learning, formerly Haiku Learning
- Sakai
- Schoology
- Thought Industries Learning Business Platform (requires MEETS version 7+)

For more information regarding *connections*, please see above in the section *MEETS SaaS Organization*.

If you use the MEETS SaaS LTI Launch Manual to add federation into MEETS from your own software, the platform type of the LTI connection will be *Other*. Using the API functions described here will allow you to create a tighter integration with MEETS than what is typically supported by the above identified platforms.

MEETS SaaS is often integrated into open source PHP-based Content Management Systems by making use of these APIs, including:

- Drupal
- Joomla
- WordPress

These platforms do not support LTI natively, but are extensible enough that they can be customized with the appropriate developers.

Conferencing Services

API functions will indicate a specific conferencing services with one of the following strings:

- bluejeans — BlueJeans
- collaborate — Blackboard Collaborate
- gotomeeting — GotoMeeting by LogMeIn, formerly Citrix GotoMeeting
- gototraining — GotoTraining by LogMeIn, formerly Citrix GotoTraining
- gotowebinar — GotoWebinar by LogMeIn, formerly Citrix GotoWebinar
- webexeventcenter — Cisco Webex Events, formerly WebEx Event Center

- webexmeetingcenter — Cisco Webex Meetings, formerly WebEx Meeting Center
- webextrainingcenter — Cisco Webex Trainings, formerly WebEx Training Center
- zoom — Zoom Meetings
- zoomwebinars — Zoom Webinars

Please be aware these strings are based on names used by a particular conferencing service when support was added to MEETS. Their names may have changed over time. Historical names are used for the sake of backwards compatibility.

This list of services that MEETS may indicate will obviously be further limited by which conferencing services are supported by your subscription or installation of MEETS.

Output Format

JSON

Almost all responses from MEETS SaaS API will be represented by a single JSON object. This object minimally contains a member named "status", whose value will be an integer matching that of the HTTP status code of the response. This status indicates which additional members are available, according to the range of codes the HTTP status falls within.

Successful Responses

Successful responses will return an HTTP 2xx status. The body of the response is category-specific and described below.

Redirection Responses

Redirection responses responses will return an HTTP 3xx status. An additional JSON value will be present, named "location" specifying the URL to which to redirect, identical to that which will be set in the response's HTTP "Location" header. (If the status was specifically 300, which means "Multiple Choices", the "location" member will instead be an array of possible URLs.) Note that such redirection must be implemented according to the HTTP 1.1 standard in terms of whether to reissue the same request at the new URL, switch method, etc. In practice, only 300 and 303 will be used, and as such, any redirection should be issuing a new GET request to the provided URL (or one of the provided URLs) and should NOT re-authenticate the new request.

It is important to note is that the resulting URL may be to a third-party server which may employ further redirection, or even require JavaScript operations to process their responses, before reaching its final destination. Please consult the documentation of the relevant third-party service for any considerations post-redirect.
Example:

```
{"status":303,"location":"http://example.com/go_here/your_desired_url"}
```

Error Responses

Error responses will return an HTTP 4xx status for client errors and an HTTP 5xx status for server errors. An additional JSON value will be present, named "error", which will be a string describing what went wrong.
Example:

```
{"status":404,"error":"Conferencing event with ID \"322\" not found"}
```

Non-JSON

Successful Responses

Successful non-redirect responses from the "download" category, which are designed to provide direct access to file data, are not JSON by necessity (though errors which occur with this category will still use the usual JSON error format).

Error Responses

The only other cases which will not return JSON are some rare error conditions (such as a connection timing out). These cases will always have the HTTP status of 500. The response in such cases will be unspecified, and are recommended to be treated as a temporary, unknown error.

Function Categories

Functions in the MEETS SaaS API are grouped by category. Each category consists of a verb which best describes its functions' purpose and behavior. The actual names of a function in a category will begin with that category's verb, followed by the subject on which the operation described by the verb will be performed. For example "list_admins" is a function from the "list" category and "download_recording" is a function from the "download" category. Functions bearing the same category as one another will also utilize the same HTTP method and output format.

list

List is the most commonly used type of function for accessing data, and are accessible via either GET or POST. Data within MEETS is organized into several entity types: admins, conference recordings, conferencing accounts, conferencing events, connections (to LMSs), courses, and users (each is elaborated upon below). The response from "list" functions will consist of members named for each entity type, each containing an object which maps that entity's numeric identifier (as the object's keys) to data objects representing the entity with that ID (as the object's values). Additionally, the top-level response object will contain a member called "nextPageStartAfterId" which is used for pagination (described below).

Expected responses:

- 200 "Success" — Indicates that the query was successful and the desired data is attached. Please note that while the query may have been successful, the requested objects may still be empty if there was no data matching the query. This is by design.

Entity Types

Each list function requests a certain type of entity and allows limiting the results by certain criteria. The member representing the map for entity type being requested will include any which match those criteria (within the pagination limit), and the other entity maps will contain any other types which are related in some way to the requested type (or related to others which are related to them, etc.) and which might be necessary for cross-referencing. For example, the function to list conferencing events will also contain a list of any users who are or were the hosts of the matching events. The event objects themselves will contain a user ID which can be used to look up the complete user information in the users list. Users likewise belong to a particular connection, and will themselves have an ID which can be used to look up the connection information. For the purpose of the examples in this documentation, `response` will be assumed to mean the top-level response object and `thisEntity` will be assumed to mean an entity object of the type being documented. The entity types which exist and the members they will minimally contain are as follows:

admin

These objects represent people who have access to the MEETS administration section. Each admin object will contain the following members:

- meetsIdAdmin — The numeric ID which represents this admin
- email — A string containing the admin's email address
- nameFirst — A string containing the admin's first name
- nameLast — A string containing the admin's last name
- permissions — An object mapping the names of various access permissions to boolean values for whether this admin has that permission:
  - manageAdmins — Whether this admin can create and modify accounts for other admins
  - manageConnections — Whether this admin can create, modify, and get the consumer access information for platform connections
  - viewData — Whether this admin can view information and statistical data from MEETS usage
  - manageConferencingAccounts — Whether this admin can configure or manage the MEETS conferencing accounts
  - manageContent — Whether this admin can access and manage MEETS content (files, recordings, etc.)

conferenceRecording

These objects represent recordings which were created within a conferencing event. Each conferenceRecording object will contain the following members:

- meetsIdConferenceRecording — The numeric ID which represents this recording
- meetsIdConferencingEvent — The numeric ID of the event in which this recording was created, which can be cross-referenced with the full event object by accessing `response.conferencingEvents[thisEntity.meetsIdEvent]`
- timeAdded — UNIX timestamp (integer) representing the time when this recording was first added to the MEETS database. Please note that this is not the time when the recording was originally created, but merely the time when MEETS first became aware of its existence.
- type — String containing either "original" or "converted". Original recordings are in whatever format is native to the conferencing service which created it. Converted recordings are those which have undergone CirQlive's auto-conversion process to convert those proprietary formats to a standard mp4.

conferencingAccount

These objects represent conferencing accounts of which MEETS is making use. Each conferencingAccount object will contain the following members:

- meetsIdConferencingAccount — The numeric ID which represents this account
- conferencingService — A string representing which conferencing service this account belongs to. Please see above regarding supported conferencing services for a list of possible values.
- conferencingUsername — A string representing the username of this account
- meetsIdUsers — An array with each element an integer representing the numeric ID of a user who is set up to be able to use this account. These can be cross-referenced with the full user objects by accessing `response.users[thisEntity.meetsIdUsers[0]]`, `response.users[thisEntity.meetsIdUsers[1]]`, etc.

conferencingEvent

These objects represent conferencing accounts which were or are to be held via MEETS. Each conferencingEvent object will contain the following members:

- meetsIdConferencingEvent — The numeric ID which represents this event
- name — A string containing the name of the event
- timeBegin — UNIX timestamp (integer) representing the starting time of this event
- timeEnd — UNIX timestamp (integer) representing the concluding time of this event
- meetsIdUser_host — The numeric ID of the user who is or was this event's host. These can be cross-referenced with the full user object by accessing `response.users[thisEntity.meetsIdUser_host]`.
- conferencingService — A string representing which conferencing service this event was or will be hosted with. Please see above regarding supported conferencing services for a list of possible values.
- meetsIdConferencingAccount — The conferencing account which was or will be used to host this event, or null if the event has not yet been registered with the conferencing service
- meetsIdCourses — An array with each element an integer representing the numeric ID of a course in which this event appears. These can be cross-referenced with the full course objects by accessing response.courses[thisEntity.meetsIdCourses[0]], response.courses[thisEntity.meetsIdCourses[1]], etc.
- meetsIdUser — An array with each element an integer representing the numeric ID of a user on whose calendar this event appears (personal meetings). These can be cross-referenced with the full user objects by accessing response.users[thisEntity.meetsIdUsers[0]], response.users[thisEntity.meetsIdUsers[1]], etc.

connection

These objects represent platform connections into MEETS (as seen in the MEETS administration section). Each connection object will contain the following members:

- meetsIdConnection — The numeric ID which represents this connection
- name — A string containing the name given to this connection as it has been configured in the MEETS administration section
- platformUrl — The base URL of the external platform making use of this connection as it has been configured in the MEETS administration section
- platformType — A string representing the type of platform making use of this connection as it has been configured in the MEETS administration section. Please see above regarding supported user/learning/training platforms for a list of possible values.

- settings — An object mapping the names of various settings to boolean values for whether the setting is on or off for this connection:
    - tlsRequired — Whether the host platform is required to be using TLS to serve its platform launcher into MEETS
    - teacherScheduleOthers — Whether teachers in this connection's courses may schedule events on behalf of other teachers within the same course
    - teacherSubstitute — Whether teachers in this connection may substitute for others by hosting another teacher's conferencing event
    - studentsHost — Whether students in this connection may have conferencing accounts and host events within courses (if not, hosting is limited to teachers)
    - featureAppointments — Whether appointment booking is available to users within this connection from within MEETS
    - featureRecordings — Whether conference recordings are available to users within this connection from within MEETS
    - featureRepository — Whether the file repository feature is enabled within this connection

course

These objects represent courses which exist within MEETS. Each course object will contain the following members:

- meetsIdCourse — The numeric ID which represents this course
- meetsIdConnection- The numeric ID of the connection in which this course exists. These can be cross-referenced with the full connection objects by accessing `response.connections[thisEntity.meetsIdConnection]`
- name — A string containing the name of the course

When the connection uses LTI, the following additional members exist:

- ltiIdContext — A string containing the course's ID as sent to MEETS via LTI. Note that depending on what LMS and LTI configuration is in use, these IDs may represent the course IDs as they exist within the LMS, or may be unique to the LMS's LTI module without a correlation to the rest of the system.
- auxiliary — An object mapping any auxiliary course data configured to be sent via LTI using MEETS's auxiliary data system, with the keys and values as those defined when the LTI link was created

user

These objects represent people who have accessed the main section of MEETS. Each user object will contain the following members:

- meetsIdUser — The numeric ID which represents this user
- meetsIdConnection — The numeric ID of the connection in which this user exists. These can be cross-referenced with the full connection objects by accessing response.connections[thisEntity.meetsIdConnection]
- nameFirst — A string containing the user's first name
- nameLast — A string containing the user's last name
- email — A string containing the user's email address

When the connection uses LTI, the following additional members exist:

- ltiIdUser -A string containing the user's ID as sent to MEETS via LTI. Note that depending on what LMS and LTI configuration is in use, these IDs may represent the user IDs as they exist within the LMS, or may be unique to the LMS's LTI module without a correlation to the rest of the system.

- auxiliary — An object mapping any auxiliary course data configured to be sent via LTI using MEETS's auxiliary data system, with the keys and values as those defined when the LTI link was created

Pagination

All "list" functions Return up to a maximum of 1000 matching results for the primary entity type. (Associated entities may be more or less depending on what interlinks exists.) The number of results per page may be further limited by passing a query parameter named "limit" via either GET or POST, but be aware that attempts to set the limit above 1000 will be refused. All "list" function responses will have their top-level object contain a member called "nextPageStartAfterId". If the limit was not reached while retrieving data, this member will be "null". If, however, the limit WAS reached, and more data follows the data being presented, this will contain a numeric ID which can be passed to the query parameter named "startAfterId" to get the next page of results (assuming that the rest of the query remains the same). To make such pagination more uniform, a value of 0 for startAfterId is also accepted, and will mean "start at the beginning". In practice, this means that to get a complete list of results you can make your query within a loop which updates the startAfterId to the value of the previous iteration's nextPageStartAfterId until nextPageStartAfterId comes back as null.

Working Example

This is a complete working example showing a possible request and response from the list_conference_recordings function, along with helpful comments describing certain parts of the response object. Please note that the formatting and comments are present in this documentation for the sake of clarity. Actual API responses will be unformatted, without comments or whitespace. Note also that the order in which maps, objects, and object members appear is not guaranteed to be consistent and is subject to change at any time, and should therefore not be relied upon. The order presented here is simply for clarity. Finally, please note that in practice, limiting results to a small number such as 5 is not recommended, as it will require a greater number of requests to be made to get a complete set. This number is used in this example merely for the sake of keeping the sample results short.

Request: `<api and authentication base>/list_conference_recordings?timeAddedAfter=1466158312&limit=5&startAfterId=524`

Response:

```
{
  "status" : 200, // Successful Response
  "conferenceRecordings" : // List of matching recordings
  {
    "553" :
    {
      "meetsIdConferenceRecording" : 553,
      "meetsIdConferencingEvent" : 114, // Anything beginning with "meetsId" can be cross-referenced. In this case, the event with ID
114 is present in the events map below
      "timeAdded" : 1466172120,
      "type" : "original"
    },
    "556" :
    {
      "meetsIdConferenceRecording" : 556,
      "meetsIdConferencingEvent" : 117,
      "timeAdded" : 1466172120,
      "type" : "original"
    },
    "578" :
    {
      "meetsIdConferenceRecording" : 578,
      "meetsIdConferencingEvent" : 114,
      "timeAdded" : 1466172120,
      "type" : "original"
    },
    "580" :
    {
      "meetsIdConferenceRecording" : 580,
      "meetsIdConferencingEvent" : 117,
      "timeAdded" : 1466172120,
      "type" : "original"
    },
    "587" :
    {
      "meetsIdConferenceRecording" : 587,
      "meetsIdConferencingEvent" : 117,
      "timeAdded" : 1466172120,
      "type" : "converted"
    }
  },
  "conferencingEvents" :
  {
    "114" : // This matches an event id which was referenced by the recordings above
    {
      "meetsIdConferencingEvent" : 114,
      "name" : "API Examples",
      "timeBegin" : 1442860200,
      "timeEnd" : 1442861100,
      "meetsIdUser_host" : 2, // The information for the user with ID "2" will be in the users map below
      "conferencingService" : "webexmeetingcenter",
      "meetsIdConferencingAccount" : 5,
      "meetsIdCourses" : [1] // Any IDs in this array will have their full object present in the courses map below
    },
    "117" : // This does too
    {
      "meetsIdConferencingEvent" : 117,
      "name" : "More API Examples",
      "timeBegin" : 1442865000,
      "timeEnd" : 1442865900,
      "meetsIdUser_host" : 2,
      "conferencingService" : "webexmeetingcenter",
      "meetsIdConferencingAccount" : 5,
      "meetsIdCourses" : [1]
    }
  },
  "conferencingAccounts" :
  {
    "5" :
    {
      "meetsIdConferencingAccount" : 5,
      "conferencingService" : "webexmeetingcenter",
      "conferencingUsername" : "teacheraccount",
      "meetsIdUsers" : [2] // Any IDs in this array will have their full object present in the users map below
    }
  },
  "users" :
  {
    "2" :
    {
      "meetsIdUser" : 2,
      "meetsIdConnection" : 1,
      "ltiIdUser" : "XjQ35e42",
      "nameFirst" : "Tea",
      "nameLast" : "Cher",
      "email" : "teacher@example.com",
      "auxiliary" : {}
    }
  },
  "courses" :
  {
    "1" :
    {
      "meetsIdCourse" : 1,
      "meetsIdConnection" : 1,
      "ltiIdContext" : "tLr83M1d",
      "name" : "Using MEETS SaaS APIs",
      "auxiliary" : {}
    }
  },
  "connections" :
  {
    "1" :
    {
      "meetsIdConnection" : 1,
      "name" : "Main LMS",
      "platformUrl" : "http : //lms.example.com/",
      "platformType" : "Moodle",
      "settings" :
      {
        "tlsRequired" : true,
        "teacherScheduleOthers" : true,
        "teacherSubstitute" : true,
        "studentsHost" : false,
        "featureRecordings" : true,
        "featureRepository" : true
      }
    }
  },
  //Note that there is no "admins" map present. This is because none of the above objects in the reference chain have references to
admins.
  //Only those maps which are present in the reference chain for the requested entity will be present. (These may be still be empty if
no entity of that type was actually referenced.)

  "nextPageStartAfterId" : 587 // Note that this indicates that there are more results available. If this were null, it would indicate
that there are no further results.
}
```

To get the next set of results in the same results set, based on the above example, a new request should be issued.

Request: `<api and authentication base>/list_conference_recordings?timeAddedAfter=1466158312&limit=5&startAfterId=`**`587`**.

get

Get is used to retrieve data reports for a specific entity.

Expected responses:

- 200 "Success" — Indicates that the query was successful and the desired data is attached. Please note that while the query may have been successful, the requested objects may still be empty if there was no data matching the query. This is by design.

from

From is used to convert an identifier from a platform integrated with MEETS SaaS to the identifiers MEETS uses internally. As most API functions offer filtering by MEETS' internal identifiers, to filter by an external identifier requires first using a from function to look up the correct identifier to use.

Expected responses:

- 200 "Success" — Indicates that the query was successful and the desired MEETS internal identifier is attached. Please note that MEETS does not change its internal identifiers that are linked to an external one. Therefore the identifier returned for an external one can be cached permanently.

map

generate

create

modify

delete

upload

download

This category of function provides direct access to a file's data. It may be accessed via either GET or POST, and is the only category whose successful responses will not necessarily contain JSON.

Expected responses:

- 303 "See Other" — Indicates that the file is being stored on a third-party's server (such as a conference recording being stored on the conferencing service's servers) and is redirecting the requester there via the Location header (the response data in this case will still be JSON). Note that the subsequent request to the redirect URL should NOT be re-authenticated using MEETS SaaS Authentication. Also important to note is that the resulting third-party URL may employ further redirection, or even JavaScript operations, before reaching its final destination. Please consult the documentation of the relevant third-party service for any considerations post-redirect. Finally, please note that the URL being redirected to may be temporary and is intended only for immediate redirection. If a long-term URL to the file is desired, the "generate" family of functions should be used instead.
- 200 "Success" — Indicates that the file was found locally and its contents are attached.

Functions

list_admins

Get a list of admin users who can administrate the MEETS SaaS platform.

Main entity: admins

list_connections

Get a list of platform connections which were created in a MEETS platform.

Main entity: connections

list_course_users

Get a list of users which exist in a particular course (also known as a "context"), along with their roles in that course and associated cross-reference entities.

Main entity: users

Required Parameters:

- meetsIdCourse — MEETS-internal course ID.

Optional Parameters:

- roles — Comma-separated list of "administrator", "teacher", "student". Results will be limited to those users who have at least one of the specified roles within the specified course.
- rolesExclude — Comma-separated list of "administrator", "teacher", "student". Results will be limited to those users who do not have any one of the specified roles within the specified course.

Successful response data will add the following to the users entity beyond the minimum:

- roles; An object with the following keys:
  - administrator — A boolean, as to whether the user has administrator privileges within the specified course.
  - teacher — A boolean, as to whether the user has teacher privileges within the specified course.
  - student — A boolean, as to whether the user is expected to participate as a student within the specified course.

list_courses

Get a list of courses (also known as "contexts") which exist in the MEETS platform, along with associated cross-reference entities.

Main entity: courses

Optional Parameters:

- meetsIdConnections — Comma-separated list of MEETS-internal connection IDs. Results will be limited to those courses present in at least one of the connections listed.
- meetsIdUsers_canAccess — Comma-separated list of MEETS-internal user IDs. Results will be limited to those courses that at least one of the specified users can access.
- meetsIdUsers_isAdministrator — Comma-separated list of MEETS-internal user IDs. Results will be limited to those courses that at least one of the specified users can access as an administrator.
- meetsIdUsers_isTeacher — Comma-separated list of MEETS-internal user IDs. Results will be limited to those courses that at least one of the specified users can access as a teacher.
- meetsIdUsers_isStudent — Comma-separated list of MEETS-internal user IDs. Results will be limited to those courses that at least one of the specified users are present as students.

If multiple filtering parameters are used, results will be limited to those which match ALL provided filters

list_conferencing_events

Get a list of web conferencing events, along with associated cross-reference entities.

Main entity: conferencingEvents

Optional Parameters:

- meetsIdConnections — Comma-separated list of MEETS-internal connection IDs. Results will be limited to those events available from courses in at least one of the connections listed.
- meetsIdCourses — Comma-separated list of MEETS-internal course IDs. Results will be limited to only events available to at least one of the courses listed.
- meetsIdUsers_host — Comma-separated list of MEETS-internal user IDs. Results will be limited to only events scheduled to be hosted by at least one of the specified users.
- meetsIdConferencingAccounts — Comma-separated list of MEETS-internal conferencing account IDs. Results will be limited to only events hosted using one of the listed accounts.
- conferencingServices — Comma separated list of strings representing conferencing services. Please see above regarding supported conferencing services for a list of possible values.
- timeBeginBefore — UNIX timestamp. Only events scheduled to begin before the given time will be included in the results.
- timeBeginAfter — UNIX timestamp. Only events scheduled to begin after the given time will be included in the results.
- timeEndBefore — UNIX timestamp. Only events scheduled to end before the given time will be included in the results.
- timeEndAfter — UNIX timestamp. Only events scheduled to end after the given time will be included in the results.

If multiple filtering parameters are used, results will be limited to those which match ALL provided filters

list_conference_recordings

Get a list of recordings which were made during web conferencing events, along with associated cross-reference entities.

Main entity: conferenceRecordings

Optional Parameters:

- meetsIdConnections — Comma-separated list of MEETS-internal connection IDs. Results will be limited to those recordings from events available from courses in at least one of the connections listed.
- meetsIdCourses — Comma-separated list of MEETS-internal course IDs. Results will be limited to those recordings from events available to at least one of the courses listed.
- meetsIdConferencingEvents — Comma-separated list of MEETS-internal event IDs. Results will be limited to only recordings from the listed events.

- meetsIdUsers_host — Comma-separated list of MEETS-internal user IDs. Results will be limited to only recordings from events hosted by one of the listed users.
- meetsIdConferencingAccounts — Comma-separated list of MEETS-internal conferencing account IDs. Results will be limited to only recordings from events hosted using one of the listed accounts.
- type — Either "original" or "converted". Results will be limited to the given type of recording.
- conferencingServices — Comma separated list of strings representing conferencing services. Please see above regarding supported conferencing services for a list of possible values. Please note that GotoMeeting and GoToWebinar do not support this feature, and therefore their values cannot be present.
- timeAddedBefore — UNIX timestamp. Only recordings which were added to MEETS's database before the given time will be included in the results.
- timeAddedAfter — UNIX timestamp. Only recordings which were added to MEETS's database after the given time will be included in the results.
- timeEventBeginBefore — UNIX timestamp. Only recordings which were made from events which began before the given time will be included in the results.
- timeEventBeginAfter — UNIX timestamp. Only recordings which were made from events which began after the given time will be included in the results.
- timeEventEndBefore — UNIX timestamp. Only recordings which were made from events which ended before the given time will be included in the results.
- timeEventEndAfter — UNIX timestamp. Only recordings which were made from events which ended after the given time will be included in the results.

If multiple filtering parameters are used, results will be limited to those which match ALL provided filters

Notes:

- It is important to keep in mind that in terms of MEETS working with third party service data that MEETS is eventually-consistent. Therefore all usage of this API function should be aware of this and only be used within an eventually-consistent programming model.
- MEETS generally has no way of knowing about a conference recording at the moment it was created. MEETS periodically reviews recent events that occurred to find recordings via a pseudo-cron job. Only recordings that MEETS currently knows about will be returned with this API function.
- Likewise if a recording was directly deleted from the conferencing service's side of things, MEETS does not know about it. MEETS does not scan old events to check if recordings were deleted.
- MEETS learns about recordings that were deleted when one tries to access them.

list_users

Get a list of users which exist in the MEETS platform, along with associated cross-reference entities.

Main entity: users

Optional Parameters:

- meetsIdConnections — Comma-separated list of MEETS-internal connection IDs. Results will be limited to those users present in at least one of the connections listed.
- meetsIdCourses — Comma-separated list of MEETS-internal course IDs. Results will be limited to only users available to at least one of the courses listed.

If multiple filtering parameters are used, results will be limited to those which match ALL provided filters

get_conferencing_event_attendance

Retrieve attendance for a specific web conferencing event. Required Parameters:

- meetsIdConferencingEvent — MEETS-internal event ID.

Successful response data:

- status — Status of the request
- attendance; An array of the following object, where each entity represents when a user attended:
  - wasHost — A boolean, as to whether the attendee was the host of the session or an unprivileged participant.
  - attendeeName — Full name of the attendee.
  - emailAddress — Electronic mail address of the attendee.
  - ip — Internet Protocol address (IPv4 or IPv6) of the attendee.
  - browser — Details regarding the browser used by the attendee.
  - timeJoin — UNIX timestamp of when the attendee joined the session.
  - timeLeave — UNIX timestamp of when the attendee left the session.
- isFinal — A boolean, as to whether the attendance data appears to be finalized, or should be rechecked later for possibly more accurate results (due to eventual consistency).

Notes:

1. At this time, only the following services supply attendance data:
   - BlueJeans
   - Blackboard Collaborate
   - GotoMeeting by LogMeIn
   - GotoTraining by LogMeIn
   - GotoWebinar by LogMeIn
   - Cisco WebEx Events
   - Cisco WebEx Meetings
   - Cisco WebEx Trainings
   - Zoom Meetings (up to 2000 times per day)
   - Zoom Webinars (up to 2000 times per day)
2. Not all services will supply values for all these fields.
3. If a participant joined and left multiple times in a session there will be multiple entries for that participant.
4. Some conferencing services may not supply attendance data until a full 24 hours after the session.
5. Some conference services may not keep attendance data for sessions that occurred long ago.
6. Conferencing services will not supply attendance data if no one attended. Some won't supply it if only a single person attended.

7. If the event exists, but there is no data to show, either because the service does not support attendance (as per note 1), attendance data is not available (as per note 4 and 5), or because not enough people attended the session (as per note 6), the attendance entry will consist of an empty JSON array ("`[]`"). This does not indicate an error.
8. The attendance data is taken verbatim from the web conferencing services.

Example request: `<api and authentication base>/get_conferencing_event_attendance?meetsIdConferencingEvent=1538`

Example response:

```
{
  "status" : 200, // Successful Response
  "attendance" : // List of web conferencing event attendance
  {
    [
      {
        "wasHost" : false,
        "attendeeName" : "Jane Dough (6)",
        "emailAddress" : "jdough@example.com",
        "ip" : "127.0.0.1",
        "browser" : "WINDOWS, CHROME",
        "timeJoin" : 1429139036,
        "timeLeave" : 1429139103
      },
      {
        "wasHost" : false,
        "attendeeName" : "Jane Dough (6)",
        "emailAddress" : "jdough@example.com",
        "ip" : "192.168.1.1",
        "browser" : "WINDOWS, FIREFOX",
        "timeJoin" : 1429150536,
        "timeLeave" : 1429183365
      },
      {
        "wasHost" : true,
        "attendeeName" : "The Puppetmaster (13)",
        "emailAddress" : "pmaster@example.com",
        "ip" : "10.0.2.254",
        "browser" : "LINUX, GALEON",
        "timeJoin" : 1429139000,
        "timeLeave" : 1429185363
      },
      {
        "wasHost" : false,
        "attendeeName" : "John Dough (7)",
        "emailAddress" : "jdough2@example.com",
        "ip" : "192.168.1.2",
        "browser" : "WINDOWS, EDGE",
        "timeJoin" : 1429150338,
        "timeLeave" : 1429184365
      }
    ]
  },
  "isFinal" : true // This data set is finalized and will not have further improvements
}
```

get_conferencing_authentication_log

Retrieve a log of authentication MEETS performed with conferencing services.

Optional Parameters:

- limit - For pagination usage as is described for "list" functions.
- startAfterId - For pagination usage as is described for "list" functions.
- timeAfter - Limit logs to those that occurred after the specified UNIX timestamp.
- timeBefore - Limit logs to those that occurred before the specified UNIX timestamp.

Successful response data:

- status — Status of the request
- conferencingAuthenticationLog; An array of the following object, where each entity represents a log entry:
  - meetsIdLogConferencingAuthentication — The numeric ID representing this log entry.
  - conferencingService — A string representing a conferencing service. Please see above regarding supported conferencing services for a list of possible values.
  - meetsIdConferencingAccount — The numeric ID which represents the account used. Can be *null* if no conferencing account was used (which is typical when doing administrator level access).
  - conferencingUsername — A string representing the username of this account at the time the log entry was created.
  - authenticationMethod — An integer describing the authentication method used. *0* — user credentials with a regular user account, *1* — user credentials with the administrator account, *2* — singular authentication with a regular user account, *3* — singular authentication with the administrator account.
  - time — UNIX timestamp of when the log entry was created.
- nextPageStartAfterId — If the limit was not reached while retrieving data, this member will be "null". If, however, the limit WAS reached, and more data follows the data being presented, this will contain a numeric ID which can be passed to the query parameter named "startAfterId" to get the next page of results (assuming that the rest of the query remains the same).

Notes:

- This function is provided for those with conformance requirements. There is no actual expected utility this function provides.

- Log entries in no way indicate which user if any performed an operation. Only the account used is described, which can occur for one user on behalf of another, or even when MEETS is running background operations.
- The log only contains historical data, there is no expected link between entries and live data that other functions may return. Relationships between data may have changed since the log entry was made.

from_lti_id_context

Convert an LTI context ID to the ID used for the corresponding course in MEETS.

Required Parameters:

- meetsIdConnection — The numeric ID of the MEETS LTI connection in which this LTI context ID exists.
- ltiIdContext — A string containing the context ID as sent to MEETS via LTI.

Successful response data:

- status — Status of the request.
- meetsIdCourse — The numeric ID which MEETS uses to represent the course associated with the LTI context.

from_lti_id_user

Convert an LTI user ID to the ID used for the corresponding user in MEETS.

Required Parameters:

- meetsIdConnection — The numeric ID of the MEETS LTI connection in which this LTI context ID exists.
- ltiIdUser — A string containing the user ID as sent to MEETS via LTI.

Successful response data:

- status — Status of the request.
- meetsIdUser — The numeric ID which MEETS uses to represent the user associated with the LTI user.

download_conference_recording

Download a certain recording made during a web conference.

Required Parameters:

- meetsIdConferenceRecording — The MEETS-internal recording ID for the recording to download.

Notes:

- You will encounter a failure with a *status* of *500* if the externally hosted conference recording is no longer available on the remote host. When this occurs, the recording will also be removed from appearing in subsequent calls to the *list_conference_recordings* API function.