Confex Meeting API

as of 2020-04-06

Author	Proofed by	Date	Notes:
Joseph Boissonneault	Kevin O'Neill	1/11/2018	Initial Document
Joseph Boissonneault	Jay R.	8/27/2018	Permissions
Kevin O'Neill	Joseph Boissonneault	4/6/2020	Suggested removal of apha and acr-specific URLs and hyperlinks and minor grammatical changes

General Description:

The general purpose of the Confex Meeting API is to provide extremely fast access to conference program data -- the essential “who”, “what”, “when”, and “where” of conference activities -- for applications hosted by Confex, our clients, and 3rd-party vendors engaged by our clients. It includes multiple objects, making it easy to fetch and consume JSON data organized in the unique style appropriate for each client.

Technology Used:

- MySQL, filesystem – main data store, used for the most volatile data
- ElasticSearch - searchable data store, where we compose data - https://www.elastic.co/
- RESTful API returning JSON from main data store or ElasticSearch

Object Tree:

Each Confex client organizes their data in a unique way. However, most have a common core with several additional methods of indexing their data. The following is the basic tree of objects for any Confex client.

Basic types of objects:

- Entries - An entry contains information to display to end-user, child entries, and child person/role data.
- SlotData - Contains the data regarding time/location of a child session or paper and also acts as a summary of child entry for ease of use.
- Role - Links an entry and a person record, they contain additional information, such as flags to determine who is the presenter.
- Person - Person data.
- Others - There are many other types of objects, such as FileMaps, CreditEvaluation, Favorites, Notes, Events, etc. You can see the complete list of Objects in Appendix A.

Parent-child relationships for entries:

Meeting → Program → Symposium → Session → Paper

Parent-child relationship for Slots/Entries:

[Program, Symposium] → SlotData → [Session, Paper]

Parent-child relationships for Entries/People:

[Program, Symposium, Session, Paper] -> Role -> Person

Meeting API

Description

The Meeting API is a RESTful API enabling CRUD (create, read, update, and delete) methods. This document covers only the read method. Third party vendors may obtain additional write method documentation on request of client.

(GET) a single object

URL: https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/[object]/[id]

The above URL will return a JSON blob from the current client/meeting pair specified in the URL.

Shared anatomy of an object:

- **_name**: Object Name.
- **id**: Object id.
- **_url**: Typically made up of Object Name and id, there are a few exceptions.
- **_expiration**: If Confex is warming data, this is the GMT DateTime the object is set to expire. Typically, data in ElasticSearch data store is actively warmed 3 months prior and 3 months after a meeting. We then rely on event/trigger which covers 98% of the changes made to objects in main data store. Occasionally, when support staff change configuration we will need to manually invalidate an object.
- **_originated:** GMT DateTime when object was created in ES.
- **_lastchanged**: GMT DateTime when change to main data store occured.
- **_meetingappchanged**: Field specific to Confex MeetingApp, checks only fields used in templates of Confex Meeting Application.
- **_meetingcode:** Code of the current meeting.
- **ParentList_***: An arrayref of objects, which are a parent of object.
- **ChildList_***: An arrayref of objects, which are a child of object.
- **label**: Label of an object, for an end-user to see.
  NOTE: It may differ for objects with the same name, I.E. Paper objects may use labels “Proposal”, “Abstract”, “Paper”, etc.
- **_maxRefresh**: The max time before an object is refreshed.
- ***:** All other fields are specific to object type. See Appendix A for shared mappings of common objects.

Example: The following is a simple example of fetching the highest level object “Home/0”. It contains basic information on helpful indices to show, start/end dates of meeting, and searchable objects.

curl --insecure -v -H "Content-Type: application/json" -H -X GET https://client.confex.com/client/code/meetingapi.cgi/Home/0


```
{
  "SearchableModels": [
    "Session",
    "Paper",
    "Person"
  ],
  "ChildList_Favorites": [
    "Favorites/Entries",
    "Favorites/Schedule",
    "Favorites/People"
  ],
  "ChildList_FilterByIndice": [
    "ModuleSessionsByDay/0",
    "ModuleProgramBook/0",
    "Index/CESession~1",
    "Index/Handout~1",
    "Index/Recording~1"
  ],
  "Meeting_MeetingEnd": "2016-11-02",
  "_lastchanged": "2016-10-26T16:03:50",
  "Meeting_URL": "http://www.[client].org/events-and-meetings/annual",
  "ChildList_Module": [
    "ModuleSessionsByDay/0",
    "ModuleProgramBook/0",
    "Index/isPoster~1",
    "Index/PeopleIndex~A",
    "Index/Affiliation^A",
    "Index/CESession~1",
    "Index/Handout~1",
    "Index/Recording~1",
    "ModuleMeetingInfo/0"
  ],
  "Meeting_MeetingStart": "2016-10-29",
  "id": "0",
  "_url": "Home/0",
  "_expiration": "2017-05-12T16:43:10",
  "_meetingappchanged": "2016-10-26T16:03:50",
  "Meeting_ClientHomePageLink": "",
  "_meetingcode": "[meetingcode]",
  "_name": "Home",
  "MeetingTitle": "[client] 2016 Annual Meeting & Expo (Oct. 29 - Nov. 2, 2016)",
  "ChildList_CreditsEval": [
    "CreditEval/Meeting21"
  ],
  "label": "Home",
  "_originated": "2016-09-28T16:10:42",
  "_maxRefresh": "86400000"
}
```


(GET) multiple objects

https://[client].confex.com/[client]/[meetingcode]/[Object]

The above URL will return an array of JSON objects.

Additional paging and query options:

- size - size of each page. (**deprecated**)
- page - page number to retrieve. (**deprecated**)
- urls - an array of urls to fetch.
- since - checks a date field and returns objects changed since value specified.
- before - checks a date field and returns objects changed since value specified.
- datefield - default is _lastchanged. Allows you to change date searched on.
- Fields - used for minimizing data returned, ex: [‘Label’, ‘Title’, ‘_url’]

  NOTE: The date format uses GMT format YYYY-MM-DDTHH:MM:SS (ie 2007-03-19T03:21:00)
  NOTE: All the above params are supported using the header X-Confex-MeetingApi-Config. Excepting “Fields”, all are supported in the params of the URL as well.

Example: The following example will fetch the first 100 people who were changed since 2016-07-04. It will limit the return to FirstName, LastName, and EmailAddress.

curl -v -H "Content-Type: application/json" -H 'X-Confex-MeetingApi-Config: {"size": 100, "page": 1, "Fields": ["FirstName", "LastName", "EmailAddress"]}' -X GET https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Person

Relevant Header Response:
X-Confex-MeetingApi-Config:
{"Fields":["FirstName","LastName","EmailAddress"],"page":1,"pagetotal":168,"total":16766,"IncludeDefaults":0,"size":100}

Data Response Example:
[{"EmailAddress":"uzma.khan@aku.edu","FirstName":"Uzma","LastName":"Rahim Khan","_expiration":"2016-04-22T19:13:50","_lastchanged":"2016-04-21T19:13:50","_maxRefresh":"86400000","_meetingappchanged":"2016-04-21T19:13:50","_meetingcode":"143am","_name":"Person","_originated":"2016-03-31T19:16:16","_url":"Person/323164","id":"323164"},{"EmailAddress":"ammi@pitt.edu","FirstName":"Ann M.", "LastName":"Mitchell","_expiration":"2016-04-22T19:13:50","_lastchanged":"2016-04-21T19:13:50","_maxRefresh":"86400000","_meetingappchanged":"2016-04-21T19:13:50","_meetingcode":"143am","_name":"Person","_originated":"2016-03-31T18:56:23","_url":"Person/263428","id":"263428"},{"EmailAddress":"mrosenma@iu.edu","FirstName":"Marc","LastName":"Rosenman","_expiration":"2016-04-26T18:10:35","_lastchanged":"2016-04-25T18:09:21","_maxRefresh":"86400000","_meetingappchanged":"2016-04-25T18:09:21","_meetingcode":"143am","_name":"Person","_originated":"2016-03-31T19:23:46","_url":"Person/301336","id":"301336"}, … …]

(Get) multiple objects using parent model / Walking the meeting api tree.

Starting with the model “Home/0”, you can easily walk through a meeting using Parent and Child lists. To fetch a list of child objects use the following URL format.
https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/[Model]/[id]/[ListName]

If you look at the Object Home/0, you will see the array:
["ModuleProgramBook/0", "ModuleSessionsByDay/0", "Index/KeynoteAndPlenary~1", "ModuleKokesAwardees/0", "Index/PeopleIndex~A", "ModuleKeywordIndex/0", "ModuleSupporters/0", "ModuleFloorPlans/0", "ModuleMeetingInfo/0"].

The following example will fetch the full objects of the above list. We will look at one of those objects: "ModuleProgramBook/0".

curl -v -H "Content-Type: application/json" -H 'X-Confex-MeetingApi-Config: {"size": 100, "page": 1, "Fields": ["FirstName", "LastName", "EmailAddress"]}' -X GET
https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Home/0/ChildList_FilterByIndice

Results:

> [{_lastchanged: "2016-08-26T20:48:10",_url: "ModuleProgramBook/0",_expiration: "2017-02-22T18:47:18",_meetingappchanged: "2016-08-26T20:48:10",_name: "ModuleProgramBook",_meetingcode: "[meetingcode]",
> **ChildList_Program: ["Program/1616","Program/1565","Program/1618","Program/1614","Program/1615", ..., "Program/1574","Program/1633"],**
> NavBarTitle: "Browse by Program", id: "0", label: "Browse by Program", _originated: "2016-05-18T20:51:20", _maxRefresh: "86400000" }, { ... }, ...]

NOTE: In the above example we have the object ModuleProgramBook/0, which contains a list of all the programs. You can then continue walk down the data tree using:
https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/ModuleProgramBook/0/ChildList_FilterByIndice

(Get) a single object, from the intersection of 2 other objects.

Getting the intersection of 2 objects is easy way to get the intersection of the shared lists.

https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Program/1616

https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Day/20161031

https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Day/20161031/Program/1616

You can combine 2 or more objects in this way. Any child or parent list that they share will be an intersection of the two. Therefore the /Day/20161031/Program/1616 will display only the sessions on the day of 10/31.

ORDER MATTERS: The resulting JSON blob will inherit all scalar refs from the last object, all shared array refs will be an intersection of the two or more objects. The order is preserved using the last objects sort order.

Fetching privileged data

Some models or fields on a model are embargoed till a certain date passes, however, third party vendors need this information before date because, because they need to test their applications or they are using it for financial data, contacting end-users, etc.

If you are a privileged user, you must contact the confex support person and notify them that you will need a “Vendor” or “ApiVendor” permission, these are the typical user groups setup for permissions to excluded objects or object fields. ApiVendor by default can access all models, but do not get permission

to sensitive fields, Vendor usergroup by default doesn't get access to all models but have access to exclude fields like phone number, email address, etc. If you need a 3rd/4th type of usergroup setup, let your confex support person know and they can modify the permissions of user group to accommodate.

Once you have your username and password setup with the correct usergroup, you may access privileged information by setting param on calling url.

Example:

1. curl -v -H "Content-Type: application/json" -X PUT -d '{"username":"jonboy@tristarpub.com", "password":"PASSWORD-GOES-HERE"}' https://[client].confex.com/[client]r/2018/meetingapi.cgi/Login/current
2. Look in the header and use the the usersession cookie.
   ... Set-Cookie: usersession=h-s6EVQHTMizuppDIOWSdg; path=/ …
3. curl -H "Content-Type: application/json;" --cookie "usersession=h-s6EVQHTMizuppDIOWSdg" -X GET https://[client].confex.com/[client]/2018/meetingapi.cgi/Program?privilege=1

Appendix A. Shared Objects

Legend: Important, Infrequently used, **To be deprecated**

Entries

- **Home** - Top level index - contains meeting information, top level indices, and configuration options for search.
- Subject - Glorified keywords with people whom may be responsible for some organizational aspect of child sessions, or papers.
- **Program -** Programs are an index of Symposium, Session, and Papers, they typically have associated organizers though most clients do not display their names.
- Symposium - Symposium are an index of Sessions, they typically have Organizers who are displayed, and child SlotData.
- **SlotData** - SlotData contains Date/Time/Place data, and a summary of Session||Paper data.
- **Session** - Sessions almost always have associated Person data, Title, Summary, and Child Papers. Occasionally, abstract information, file attachments will be included.
- Cluster - Organization bucket between Session/Paper, rarely used.
- **Paper** - Title, SubTitle, Abstract data, and Author/Presenter list.

Person Data

- **Person** - Person data.
- **Role** - Role links an Entry and a Person, contains additional information specific to Person/Entry pair.

- Address - Address for a Role or Person.

Entry Object Attachments

- Booth - Booth information for Exhibitor sessions.
- Discussion/Comment - Discussion links entry to a discussion in meetingapp.
- **MediaFiles/FileMap** - MediaFiles links entry to FileMap data, FileMap data contains link and status information for an attachment.

High Level Indices

- **Index** - Generic ElasticSearch index. Custom client indices are built using this object type.
    - Examples: Index/GraduateStudent~Y, Index/Plenary~1, Index/FileMap~ExtendedAbstracts, Index/AnyField~Value
- **ModuleProgramBook** - Index of SlotData/sessions/papers by program.
- **ModuleSessionsByDay/Day** - Index of days, Day is index of SlotData.
- ModuleSubjectIndex - Index of Subjects.
- **ModulePeopleIndex** - Index of people, typically authors or presenters.
- ModuleExhibitorProgram - Index of Exhibitors, which are stored on Session objects.
- ModuleKeywordIndex - Index of Keywords.
- Keyword - Bucket of Person, Sessions, Papers, SlotData associated with keyword.
- ModulePosterSessions - Index of Poster sessions.
- ModuleSymposiumBook - Index of all Symposium organized by Program.
- ModuleWithdrawnPapers - Index of withdrawn papers.

Meeting Information Objects

- ModuleDownloadLinks - Downloads applicable to the entire meeting
- ModuleFloorPlans - Floor plans
- ModuleSupporters - Place for listing financial supports of the meeting.
- ModuleMeetingInfo - Meeting data, start times, local restaurants, etc.

Search functionality (Confex use only without authorization):

- SearchResults - (TB Deprecated) V1 User Search Utility, creates list of Session/Paper/People - includes search bar, and filtering.
- Search - V2 User Search Utility, creates list of Session/Paper/People - includes search bar, filtering, aggregation, better relevance, and faster results.
- Query - Canned queries for internal use.
- SimilarModels - Find similar models using built in ElasticSearch functionality.

User-specific Models

- Login - login model, returns user login information and sets cookies for access
- Favorites - generic model for saving favorites of models. (IE Schedule, Entries,

People)

- Event - Event, personal events that the person wants added to schedule
- Note - Personal notes
- CreditEval - Status, and links to credit evaluation system
- ProfileExchange - Status and link to create, accept, decline profile exchange

MeetingApp Models

- Template - Backbone templates for client code
- Css - CSS for client
- Sync - V1 method for syncing data, still useful for fetching all models based on date
- SyncMeetingApp - V2 method for syncing data

Appendix B. Quick start for third party vendors.

The following objects are typically used by third party vendors:
Program, Symposium, Session, Paper, SlotData, Person, Role, Address,

Program - Organization Object, it typically contains a list of Symposium, Session, and or Papers.

Symposium - Sometimes clients will use symposium to further organize sessions and papers. It will include a title, and possibly a description, and there might be a child list of people associated with the symposium itself.

Session: Sessions are the last bucket, they will almost always have an organizer or chair or moderator. Many sessions will include a synopsis of paper/talks which will be presented during the Sessions "Slot" period.

Paper: A list of presenter roles, Title, Subtitle, and abstract data.

Person: A person's data, along with a ist of associated roles, addresses, etc.

Role: A role links a person to an entry object (IE, Symposium/Session/Paper), it has additional information, such as whether or not they are presenting and the associated address. (IE. A user can have 5 affiliations but, publish a paper under only 1 of them).

SlotData: Date/Time/Location/Summary data of associated Session or Paper.

To fetch all of the above:
https://[client].confex.com/[client]/[meetingcode]/meetingapi/[Objecttype]

- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Program
- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Symposium

- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Session
- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Paper
- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Person
- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Role
- https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/SlotData

Example: curl -v -H "Content-Type: application/json" -H 'X-Confex-MeetingApi-Config: {"size": 1000, "page": 1]}' -X GET https://[client].confex.com/[client]/[meetingcode]/meetingapi.cgi/Person

Appendix C. Vendor Wish list/comments

We welcome suggestions for improvement of the Confex Meeting API, including this documentation. Please route such requests through the Confex representative who supplied you with this document. Our API developer will try to respond to all requests, indicating whether or not weplan on implementing the request and, if so, indicating the time frame for that implementation. . Please include the email address andr name of the technical contact to whom our developer should respond.