
ohbm Documentation

Release 1.0

Human Brain Mapping

Sep 27, 2017

Contents

1	Under Development	3
1.1	Installation	3
1.2	Getting Started	3
1.3	ohbm	6
2	Indices and tables	19
	Python Module Index	21

This is a wrapper for the OHBM API, for which the original spec and code is available at the [ohbm](#), repo. This API is not under development, not intended for public use, and required authentication. If you are a developer or hacker, please submit issues and requests to the [issue board](#).

CHAPTER 1

Under Development

The tool currently is a wrapped for functions related to the following for the OHBM conference:

- Abstracts
- Attendees
- Authenticate
- Categories
- Events
- Exhibitors
- Planner
- Roomset
- System

Contents:

Installation

To install

```
pip install ohbm
```

Getting Started

This is a basic example to show firing up the API in Python, please post an issue if you would like others. Please be warned that none of these functions have been tested, so bugs are likely. To get them fixed, please [report an issue](#).

Authenticating the API

After you install the Python module, you can work with the API by way of the API object:

```
#!/usr/bin/python

from ohbm.api import Api

access_token="XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
api = Api(access_token)
```

Then each class of functions (eg, Abstracts or Events) can be viewed (with corresponding functions) from within that object:

[illegible]

Welcome, OHBM Trainer!

```
In [43]: api.
```

api.Abstracts	api.Categories	api.Planner	api.base
api.Attendees	api.Events	api.Roomset	api.key
api.Authenticate	api.Exhibitor	api.System	

```
In [43]: api.Abstacts.
```

```
api.Abstacts.api
api.Abstacts.getAbstract
api.Abstacts.getAbstractSearchOptions
api.Abstacts.getAbstractSearchResults
api.Abstacts.getAbstracts
api.Abstacts.get_result
```

```
In [43]: api.Abstracts.
```

Then, for example, to use one of those functions (get all abstracts) you might do the following:

```
# Retrieve all abstracts
abstracts = api.Abstracts.getAbstracts()
```

ohbm

ohbm package

Subpackages

ohbm.script package

Submodules

ohbm.script.post_install module

Module contents

ohbm.static package

Subpackages

ohbm.static.img package

Module contents

Module contents

ohbm.templates package

Module contents

Submodules

ohbm.abstracts module

abstracts: part of the ohbm-api

class ohbm.abstracts.**Abstracts** (*api=None*)

getAbstract (*abstractID*)

Returns an XML payload containing a full listing of one abstract. Abstract data consists primarily of the title, abstract number, purpose, materials & methods, results & conclusions. Note: The result set provides urls to the figures or documents which are actual files. Sample Request URL: <http://.../?do=cnt.getservice&service=getAbstract>

Parameter Options: :param *abstractID: Numeric value of an abstract. Sample XML Payload: See REST_getAbstract.xml

getAbstractSearchOptions ()

Returns an XML payload containing search options for the getAbstractSearchResults service. The results of this call are used to obtain valid search values which can be passed to the 'getAbstractSearchResults' request. Sample Request URL: <http://.../?do=cnt.getservice&service=getAbstractSearchOptions>

getAbstractSearchResults (searchText, categoryID=None, speakerID=None, abstractTypeID=None)

Returns an XML payload containing an abstract listing. This data set returned is smaller than the getAbstracts request and intend for listing results. Use the result in combination with the getAbstract call to pull addition information on a specific abstract. Sample Request URL: <http://.../?do=cnt.getservice&service=getAbstractSearchResults>

Parameter Options: :param *searchText: Text string used to perform a general search of the events. This parameter is used to find matches in the title, description, objectives or speaker names. :param categoryID: Numeric value containing a valid category. Use the 'getSearchOptions' call to obtain a valid list of categories :param speakerID: Numeric value containing a valid speaker. Use the 'getSearchOptions' call to obtain the list of possible speakers. :param abstractTypeID: Numeric value containing a valid abstract type. Use the 'getSearchOptions' call to obtain the list of possible abstract types. acceptedFlag: Boolean value (0 or 1). Default is 1.

getAbstracts (categoryID=None, speakerID=None, abstractTypeID=None, acceptedFlag=None)

Returns an XML payload containing a full listing of all abstract data. Abstract data consists primarily of the title, abstract number, purpose, materials & methods, results & conclusions. Sample Request URL: <http://.../?do=cnt.getservice&service=getAbstracts>

Parameter Options: :param categoryID: Numeric value containing a valid category. Use the 'getSearchOptions' call to obtain a valid list of categories :param speakerID: Numeric value containing a valid speaker. Use the 'getSearchOptions' call to obtain the list of possible speakers. :param abstractTypeID: Numeric value containing a valid abstract type. Use the 'getSearchOptions' call to obtain the list of possible abstract types. :param acceptedFlag: Boolean value (0 or 1). Default is 1.

get_result (url, args=None)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.api module

ohbm.app module

ohbm.attendees module

attendees: part of the ohbm-api

class ohbm.attendees.**Attendees** (api=None)

addAttendeeActivity (attendeeID, registrationNumber=None, eventIDs=None, eventID=None)

Returns an XML payload containing the results of the attendee add activity method. The 'stat' attribute is used to indicate the success or failure of the request. Extended description of all options are listed below. Sample Request URL: [http://.../?do=cnt.getservice&service=addAttendeeActivity&{\[\]Parameter List}](http://.../?do=cnt.getservice&service=addAttendeeActivity&{[]Parameter List})

Parameter Options: :param *attendeeID: Numeric value used to identify the attendee record to update. attendeeID or registrationNumber required. :param registrationNumber: String value used to identify the attendee record to update. attendeeID or registrationNumber required. :param eventIDs: Comma delimited

list of event to be added to a person's activity list. eventIDs or eventID required :param eventID: Numeric value identifying an event to be added to a person's activity list. eventIDs or eventID required.

getAttendeesCreditActivity (*attendeeID=None, attendeesID=None, memberNumber=None, memberNumberPartial=None, lastNameInitialStart=None, lastNameInitialEnd=None, plannerID=None, claimedOnAfter=None, claimedOnBefore=None*)

Returns an XML payload containing and attendee(s) data and credit activity grouped by planner/activity. Credit activities consist of the claimed activities, the activities' displayed credit, the activities' credit types/accreditation bodies (based on the attendee's registration type), and the credit awarded for each credit type/accreditation body. Sample Request URL: [http://.../?do=cnt.getservice&service=getAttendeesCreditActivity&{\[\]Parameter List}](http://.../?do=cnt.getservice&service=getAttendeesCreditActivity&{[]Parameter List})

Parameter Options: :param attendeeID: Numeric value; Identifies the attendee record to retrieve. :param attendeesID: List of numeric values; Identifies attendee records to retrieve. :param memberNumber: String value. Identifies the attendee record to retrieve. :param memberNumberPartial: String value. Partial member number used to search/identify attendee records to retrieve. :param lastNameInitialStart: 1 character used to indicate the starting initial of the last name range to include. E.g. A. Must be used with lastNameInitialEnd :param lastNameInitialEnd: 1 character used to indicate the starting initial of the last name range to include. E.g. B. Must be used with lastNameInitialStart :param plannerID: Numeric value; Used to filter the activities. Identifies the planner/activity to include. :param claimedOnAfter: Date; Used to filter the activities. :param claimedOnBefore: Date; Used to filter the activities

getAttendeesCreditActivitySummary (*attendeeID=None, attendeesID=None, memberNumber=None, memberNumberPartial=None, lastNameInitialStart=None, lastNameInitialEnd=None, plannerID=None, claimedOnAfter=None, claimedOnBefore=None*)

Returns an XML payload containing and attendee(s) data and credit activity summaries grouped by planner/activity. The credit activities summaries consist of the total number of activities, display credits sum, awarded credit sum for each credit type/accreditation body. Sample Request URL: [http://.../?do=cnt.getservice&service=getAttendeesCreditActivitySummary&{\[\]Parameter List}](http://.../?do=cnt.getservice&service=getAttendeesCreditActivitySummary&{[]Parameter List})

Parameter Options: :param attendeeID: Numeric value; Identifies the attendee record to retrieve. :param attendeesID: List of numeric values; Identifies attendee records to retrieve. :param memberNumber: String value. Identifies the attendee record to retrieve. :param memberNumberPartial: String value. Partial member number used to search/identify attendee records to retrieve. :param lastNameInitialStart: 1 character used to indicate the starting initial of the last name range to include. E.g. A. Must be used with lastNameInitialEnd :param lastNameInitialEnd: 1 character used to indicate the starting initial of the last name range to include. E.g. B. Must be used with lastNameInitialStart :param plannerID: Numeric value; Used to filter the activities. Identifies the planner/activity to include. :param claimedOnAfter: Date; Used to filter the activities. :param claimedOnBefore: Date; Used to filter the activities

getAttendeesFormResponses (*formID, startInitial=None, endInitial=None*)

Returns an XML payload containing a full listing of attendee contact data and form responses for the specified form. For each attendee a 'formResponses' node returned in the XML payload. This service is only allowed to be run between 11 PM EST and 5 AM EST. Sample Request URL: <http://.../?do=cnt.getservice&service=getAttendeesFormResponse>

Parameter Options: :param *formID: Numeric value; Identifies the form for which to retrieve responses. :param attendeeID: Numeric value; Identifies the attendee record to update. :param startInitial: 1 character used to indicate the starting initial of the last name range to include. E.g. A :param endInitial: 1 character used to indicate the starting initial of the last name range to include. If the 'startInitial' is provided but no 'endInitial' is provided the system uses the 'startInitial' as the 'endInitial'. E.g. B

getAttendeesItineraryData (*attendeeID=None, startInitial=None, endInitial=None, insertInLastHoursSpan=None*)

Returns an XML payload containing a full listing of all attendee itinerary data. For each attendee their

itinerary will be returned in the XML payload. This service is only allowed to be run between 11 PM EST and 5 AM EST. Note: For clients using the Attendee Itinerary Update service, only itinerary items with the attributes='ReadWrite' can be modified. Sample Request URL: <http://.../?do=cnt.getservice&service=getAttendeesItineraryData>

Parameter Options: :param attendeeID: Numeric value; Identifies the attendee record to update. :param startInitial: 1 character used to indicate the starting initial of the last name range to include. E.g. A :param endInitial: 1 character used to indicate the starting initial of the last name range to include. If the 'startInitial' is provided but no 'endInitial' is provided the system uses the 'startInitial' as the 'endInitial'. E.g. B :param insertedInLastHoursSpan: Used to indicate the number of hour of newly inserted records to include. The default is 24. This must be a valid integer between 1 and 26,280.

getAttendeesRegData (*attendeeID=None, startInitial=None, endInitial=None, formID=None*)

Returns an XML payload containing a full listing of all attendee purchases. For each attendee their purchase or order history will be returned in the XML payload. This service is only allowed to be run between 11 PM EST and 5 AM EST . Sample Request URL: <http://.../?do=cnt.getservice&service=getAttendeesRegData>

Parameter Options: :param attendeeID: Numeric value, Identifies the attendee record to update. :param startInitial: 1 character used to indicate the starting initial of the last name range to include. E.g. A :param endInitial: 1 character used to indicate the starting initial of the last name range to include. If the 'startInitial' is provided but no 'endInitial' is provided the system uses the 'startInitial' as the 'endInitial'. E.g. B :param formID: Numeric value of a demographic form

getPersonActivities (*attendeeID, lastNameInitialStart=None, lastNameInitialEnd=None*)

Returns an XML payload containing the results of the attendee get activity method. The 'stat' attribute is used to indicate the success or failure of the request. Extended description of all options are listed below. Sample Request URL: [http://.../?do=cnt.getservice&service=getPersonActivities&{\[\]Parameter List}](http://.../?do=cnt.getservice&service=getPersonActivities&{[]Parameter List})

Parameter Options: :param attendeeID: Numeric value; Identifies the attendee record to retrieve. :param lastNameInitialStart: 1 character used to indicate the starting initial of the last name range to include. E.g. A :param lastNameInitialEnd: 1 character used to indicate the starting initial of the last name range to include. If the 'lastNameInitialStart' is provided but no 'lastNameInitialEnd' is provided the system uses the 'lastNameInitialStart' as the 'lastNameInitialEnd'. E.g. B

getSpeaker (*speakerID, isScheduledFor, formIDs*)

Returns an XML payload containing a full listing of one speaker. Speaker data consists of their contact information, bio and photo. Note Photo's are returned as a url to the actual photo. This service also returns the schedule for the speaker. This includes all itinerary items that are 'active' (can be changed via includeActiveStatus parameter), and 'public'. Sample Request URL: <http://.../?do=cnt.getservice&service=getSpeaker>

Parameter Options: :param *speakerID: Numeric value of a speaker. includeActiveStatus: Numeric value of (0,1, or 99) - Determines if active, inactive, or ALL itinerary items are shown in the <schedule> block. :param isScheduledFor: Numeric value of (1,2 or 3) - This filters the speakers returned by schedule type: Valid values; 1 = For Event, 2 = For Abstract, 3 = For Event OR Abstract. Note that this parameter only affects whether or not a speaker record is included in the results set. It does not change the itinerary items that are returned in each speaker's schedule block. For example, if isScheduledFor=2, Any speaker who is linked to at least 1 Abstract will be shown. Any 'active', and 'public' itinerary items that are linked to an Event will still be shown in the schedule for this speaker. :param formIDs: Comma delimited list of form ids

getSpeakers (*isScheduledFor, includeActiveStatus, formIDs, modifiedOnAfter, modifiedOnBefore*)

Returns an XML payload containing a full listing of all speaker data. Speakers are defined as attendees whom are participating in the meeting in a specific capacity. E.g. Presenting in a session or Moderating a session or Presenting a Poster. This service also returns the schedule for each speaker. This includes all itinerary items that are 'active' (can be changed via includeActiveStatus parameter), and 'public'. Sample Request URL: <http://.../?do=cnt.getservice&service=getSpeakers>

Parameter Options: `:param isScheduledFor`: Numeric value of (1,2 or 3) - This filters the speakers returned by schedule type: Valid values; 1 = For Event, 2 = For Abstract, 3 = For Event OR Abstract. Note that this parameter only affects whether or not a speaker record is included in the results set. It does not change the itinerary items that are returned in each speaker's schedule block. For example, if `isScheduledFor=2`, Any speaker who is linked to at least 1 Abstract will be shown. Any 'active', and 'public' itinerary items that are linked to an Event will still be shown in the schedule for this speaker. `:param includeActiveStatus`: Numeric value of (0,1, or 99) - Determines if active, inactive, or ALL itinerary items are shown in the `<schedule>` block. `:param formIDs`: Comma delimited list of form ids `:param modifiedOnAfter`: Valid date string. Using this filter will return speakers who have updated contact information, schedule, or form responses after/on this date. `:param modifiedOnBefore`: Valid date string. Using this filter will return speakers who have updated contact information, schedule, or form responses before/on this date.

get_result (*url, args=None*)

`get_result` takes a url and adds the `apiKey` for the `get_result` from `utils` `:param url`: the url to get the result for `:param args`: a dictionary of {"argumentName":value} to pass to the URL

updateItinerary (*attendeeID, itineraryID, abstractID=None, eventID=None, exhibitorID=None, Description=None, startsOn=None, endsOn=None*)

Returns an XML payload containing the results of the attendee itinerary data update. The 'stat' attribute is used to indicate the success or failure of the request. Extended description of all options are listed below. Sample Request URL: [http://.../?do=cnt.getservice&service=updateItinerary&{\[\]Parameter List}](http://.../?do=cnt.getservice&service=updateItinerary&{[]Parameter List})

Parameter Options: `:param *attendeeID`: Identifies the attendee record to update. Always required. `:param *itineraryID`: Identifies the itinerary record to update. Valid options: update, delete; The update option is assumed if no action is provided. `:param abstractID`: Identifies the abstract record being added, updated or removed `:param eventID`: Identifies the event record being added, updated or removed `:param exhibitorID`: Identifies the exhibitor record being added, updated or removed title. Title of the activity. Limit 300 characters. Required if record is not linked to an event, abstract or exhibitor. If linked to an event, abstract or exhibitor, title is ignored. `:param Description`: Description of the event. Limit 500 characters. `:param startsOn`: Date & Time of activity. Format: mm/dd/yyyy hh:mm tt, e.g. 05/01/2011 12:00 PM. Date / Time will be ignored if not a valid format or if passed with an event or abstract which already have a time associated. `:param endsOn`: Date & Time of activity. Format: mm/dd/yyyy hh:mm tt, e.g. 05/01/2011 12:00 PM. Date / Time will be ignored if not a valid format or if passed with an event or abstract which already have a time associated.

ohbm.auth module

auth: part of the ohbm-api

class ohbm.auth.Authenticate (*api=None*)

authenticateBase (*url, username, password*)

a general function for authenticating, and returning result `:param *url`: the URL to send the request to `:param *username`: the username `:param *password`: the password

authenticateCompany (*username, password*)

Returns an XML payload used to indicate if the sign in credentials provided are valid. When valid credentials are provided the XML payload will return the company contact's first name, last name, company and email. Sample Request URL: <http://.../?do=cnt.getservice&service=authenticateCompany>

Parameter Options: `:param *username`: Text string of the person's username `:param *password`: Text string of the person's password

authenticatePerson (*username, password*)

Returns an XML payload used to indicate if the sign in credentials provided are valid. When valid credentials are provided the XML payload will return the person's first name, last name, company and email.

Sample Request URL: <http://.../?do=cnt.getservice&service=authenticatePerson>

Parameter Options: :param **username*: Text string of the person's username :param **password*: Text string of the person's password

authenticateVendor (*username, password*)

Returns an XML payload used to indicate if the sign in credentials provided are valid. When valid credentials are provided the XML payload will return the vendor company and email. Sample Request URL: <http://.../?do=cnt.getservice&service=authenticateVendor>

Parameter Options: :param **username*: Text string of the person's username :param **password*: Text string of the person's password

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.categories module

categories: part of the ohbm-api

class ohbm.categories.**Categories** (*api=None*)

getCategories (*categoryTypeID=None, includeAdminOnlyCategories=None*)

Returns an XML payload containing a full listing of all category data. Category data consists primarily of the category name and sub-categories. Sample Request URL: <http://.../?do=cnt.getservice&service=getCategories>

Parameter Options: :param categoryTypeID: Integer valid values: 1= Event, 2=Exhibitor & 3=Abstract If not provided the system defaults to sending Event categories. :param includeAdminOnlyCategories: 0 or 1 (represents true or false).

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.events module

events: part of the ohbm-api

class ohbm.events.**Events** (*api=None*)

getEvent (*eventID*)

Returns an XML payload containing the details of a single event. Sample Request URL: <http://.../?do=cnt.getservice&service=getEvent> Parameter Options: :param apiKey: Valid API key :param eventID: Numeric value of an event.

getEventCreditsSummary (*eventID*)

getEventCreditsSummary Returns an XML payload containing the credit details of a single event. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventCreditsSummary> Parameter Options: :param **eventID*: Numeric value of an event.

getEventCreditsByRegTypeSummary (*eventID*)

Returns an XML payload containing the credit details grouped by registration type of a single event. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventCreditsByRegTypeSummary> Parameter Options: :param **apiKey*: Valid API key :param **eventID*: Numeric value of an event.

getEventSearchOptions ()

getEventSearchOptions returns an XML payload containing search options for the getSearchResults service. The results of this call are used to obtain valid search values which can be passed to the 'getSearchResults' request. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventSearchOptions>

getEventSearchResults (*searchText=None, categoryID=None, locationID=None, speakerID=None, startsOnDate=None, startsOnTime=None, statusFlag=None*)

Returns an XML payload containing an event listing. This data set returned is smaller than the getEvents request and intend for listing results. Use the result in combination with the getEvent call to pull addition information on a specific event. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventSearchResults>

Parameter Options: :param searchText: Text string used to perform a general search of the events. This parameter is used to find matches in the title, description, objectives or speaker names. :param categoryID: Numeric value containing a valid category. Use the 'getSearchOptions' call to obtain a valid list of categories :param locationID: Numeric value containing a valid location. Use the 'getSearchOptions' call to obtain a valid list of locations :param speakerID: Numeric value containing a valid speaker. Use the 'getSearchOptions' call to obtain the list of possible speakers. :param startsOnDate: Date value containing a valid date. Use the 'getSearchOptions' call to obtain a valid list of dates. :param startsOnTime: Time value in 12 hour format. E.g. 6:30PM :param statusFlag: Valid values: 1-Active; 0-Inactive; -1-All; Used to indicate if the event is active or inactive. Generally inactive events are events that have been cancelled or removed by the client.

getEvents (*statusFlag=None, publicUseFlag=None, approvedFlag=None, categoryID=None, plannerID=None, startsOnDate=None*)

Returns an XML payload containing a full listing of all event or session data. Event or session data consists primarily of the name, date, time description and location of events. Note: the tag, AI_ATTENDEEID, may be used within data elements and serves as a placeholder to be replaced by a valid attendee ID. This method is only available in its entirety between 7PM and 7AM. Between 7AM and 7PM the method will at most 5 event entries.

Parameters statusFlag – Valid values: 1-Active; 0-Inactive; -1-All; Used to indicate if the event is active or

inactive. Generally inactive events are events that have been cancelled or removed by the client. :param publicUseFlag: Valid values: 1=Public; 0=Private; -1=All; Used to determine the types of events to include. Public events are viewable to the general population. Private events are events clients are withholding from the general viewing public :param approvedFlag: Valid values: 1=Approved; 0=Not Approved; -1=Both; Used to determine the types of events to include. Note that a site setting can be used to use approvedFlag=1 by default. However, passing the parameter via the url will override the site setting parameter. :param categoryID: Numeric value; Use the 'getCategories' REST call to pull a list of valid categories. :param plannerID: Numeric value containing a valid planner. :param startsOnDate: Date value containing a valid date. Use the 'getSearchOptions' call to obtain a valid list of dates.

getEventsCreditsByRegTypeSummary (*statusFlag=None, publicUseFlag=None, approvedFlag=None, categoryID=None, plannerID=None*)

service=getEventsCreditsByRegTypeSummary Returns an XML payload containing a full listing of event or session credit data grouped by registration type. Event or session data consists primarily of the name, date, time description and location of events. Credit data primarily consists of the total credits claimed and people who have claimed those credits. Note: The tag, AI_ATTENDEEID, may be used within data elements and serves as a placeholder to be replaced by a valid attendee ID. The data excludes people who do not have a registration type.

This method is only available in its entirety between 7PM and 7AM. Between 7AM and 7PM the method will at most 5 event entries. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventsCreditsByRegTypeSummary>

Parameter Options: :param statusFlag: Valid values: 1-Active; 0-Inactive; -1-All; Used to indicate if the event is active or inactive. Generally inactive events are events that have been cancelled or removed by the client. :param publicUseFlag: Valid values: 1=Public; 0=Private; -1=All; Used to determine the types of events to include. Public events are viewable to the general population. Private events are events clients are withholding from the general viewing public :param approvedFlag: Valid values: 1=Approved; 0=Not Approved; -1=Both; Used to determine the types of events to include. Note that a site setting can be used to use approvedFlag=1 by default. However, passing the parameter via the url will override the site setting parameter. :param categoryID: Numeric value; Use the 'getCategories' REST call to pull a list of valid categories. :param plannerID: Numeric value containing a valid planner. startsOnDate: Date value containing a valid date. Use the 'getSearchOptions' call to obtain a valid list of dates.

getEventsCreditsSummary (*statusFlag=None, publicUseFlag=None, approvedFlag=None, categoryID=None, plannerID=None, startsOnDate=None*)

Returns an XML payload containing a full listing of event or session credit data. Event or session data consists primarily of the name, date, time description and location of events. Credit data primarily consists of the total credits claimed and people who have claimed those credits. Note: The tag, `AI_ATTENDEEID`, may be used within data elements and serves as a placeholder to be replaced by a valid attendee ID. This method is only available in its entirety between 7PM and 7AM. Between 7AM and 7PM the method will at most 5 event entries. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventsCreditsSummary>

Parameter Options: :param statusFlag: Valid values: 1-Active; 0-Inactive; -1-All; Used to indicate if the event is active or inactive. Generally inactive events are events that have been cancelled or removed by the client. :param publicUseFlag: Valid values: 1=Public; 0=Private; -1=All; Used to determine the types of events to include. Public events are viewable to the general population. Private events are events clients are withholding from the general viewing public :param approvedFlag: Valid values: 1=Approved; 0=Not Approved; -1=Both; Used to determine the types of events to include. Note that a site setting can be used to use approvedFlag=1 by default. However, passing the parameter via the url will override the site setting parameter. :param categoryID: Numeric value; Use the 'getCategories' REST call to pull a list of valid categories. :param plannerID: Numeric value containing a valid planner. :param startsOnDate: Date value containing a valid date. Use the 'getSearchOptions' call to obtain a valid list

of dates.

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.exhibitor module

exhibitor: part of the ohbm-api

class ohbm.exhibitor.**Exhibitor** (*api=None*)

getExhibitor (*exhibitorID*)

Returns an XML payload containing a full listing of one exhibitor. Exhibitor data consists primarily of the company name, address and contact information. Sample Request URL: <http://.../?do=cnt.getservice&service=getExhibitor>

Parameter Options: :param *exhibitorID: Numeric value of an exhibitor

getExhibitorSearchOptions ()

Returns an XML payload containing search options for the getExhibitorSearchResults service. The results of this call are used to obtain valid search values which can be passed to the 'getExhibitorSearchResults'

request. Sample Request URL: <http://.../?do=cnt.getservice&service=getExhibitorSearchOptions> Sample XML Payload: See REST_getExhibitorSearchOptions.xml

getExhibitorSearchResults (*searchText=None, categoryID=None*)

Returns an XML payload containing an exhibitor listing. This data set returned is smaller than the getExhibitors request and intend for listing results. Use the result in combination with the getExhibitor call to pull addition information on a specific event. Sample Request URL: <http://.../?do=cnt.getservice&service=getExhibitorSearchResults>

Parameter Options:

Parameters searchText – Text string used to perform a general search of the events. This parameter is used to find

param matches in the title, description, objectives or speaker names. :param categoryID: Numeric value containing a valid category. Use the ‘getSearchOptions’ call to obtain a valid list of categories

getExhibitors ()

service=getExhibitors Returns an XML payload containing a full listing of all exhibitor data. Exhibitor data consists primarily of the company name, address and contact information. Sample Request URL: <http://.../?do=cnt.getservice&service=getExhibitors>

Parameter Options: :param apiKey: Valid API key

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {“argumentName”:value} to pass to the URL

ohbm.planner module

planner: part of the ohbm-api

class ohbm.planner.Planner (*api=None*)

getActivities (*plannerID=None, activityID=None, startsOn=None, startsOnOrAfter=None, startsOnOrBefore=None, insertedOn=None, insertedOnOrAfter=None, insertedOnOrBefore=None*)

Returns an XML payload containing activity details. This payload includes the activity’s title, codes, start/end dates, description, location, and ACCME PARS information. This payload also contains a listing of the activity’s public role assignments. Sample Request URL: <http://.../?do=cnt.getservice&service=getActivities>

Parameter Options: :param plannerID: Numeric identifier for a valid planner. Use this parameter to request the details of a single planner. :param activityID: Numeric identifier for a valid activity. Use this this parameter to request the details of a single activity. :param startsOn: Date value containing a valid date. Use this parameter to request the details of any activity that begins on this date. :param startsOnOrAfter: Date value containing a valid date. Use this parameter to request the details of any activity that begins on or after this date. :param startsOnOrBefore: Date value containing a valid date. Use this parameter to request the details of any activity that begins on or before this date. :param insertedOn: Date value containing a valid date. Use this parameter to request the details of any activity that was added to the system on this date. :param insertedOnOrAfter: Date value containing a valid date. Use this parameter to request the details of any activity that was added to the system on or after this date. :param insertedOnOrBefore: Date value containing a valid date. Use this parameter to request the details of any activity that was added to the system on or before this date.

getActivity (*plannerID, activityID*)

Returns an XML payload containing an activity’s details. This payload includes the activity’s title, codes, start/end dates, description, location, and ACCME PARS information. This payload also contains a listing

of the activity's public role assignments and event (if the activity is one-to-one) Sample Request URL: <http://.../?do=cnt.getservice&service=getActivity>

Parameter Options: :param *plannerID: Numeric identifier for a valid planner. Use this parameter to request the details of a single planner. :param *activityID: Numeric identifier for a valid activity. Use this parameter to request the details of a single activity.

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.roomset module

roomset: part of the ohbm-api

class ohbm.roomset .Roomset (*api=None*)

getRoomSetSearchResults (*searchText=None, locationID=None, roomID=None, startsOn-Date=None, code=None, postFlag=None*)

Returns an XML payload containing a Room Set listing. This data set returned is smaller than the getRoomSets request and intend for listing results. Use the result in combination with the getRoomSet call to pull addition information on a specific RoomSet. Sample Request URL: <http://.../?do=cnt.getservice&service=getRoomSetSearchResults>

Request Parameters :param searchText: Text string used to perform a general search of the Room Sets. Searches for matches in function name, code, location, and room. :param locationID: Numeric value containing a valid locationID. Use the 'getRoomSetSearchOptions' call to obtain a valid list of locations :param roomID: Numeric value containing a valid roomID. Use the 'getRoomSetSearchOptions' call to obtain a valid list of rooms. :param startsOnDate: Date value containing a valid date. Use the 'getSearchOptions' call to obtain a valid list of dates. :param code: text value containing a valid code. Use the 'getSearchOptions' call to obtain the list of possible Room Set Codes. :param postFlag: Boolean value (0 or 1).

getRoomset (*roomSetID*)

getRoomSet Returns an XML payload containing a full listing of one Room Set. Room Set data consists primarily of Room, Location, Room Set Details (supplies), and Vendors. Note: The result set provides urls to the figures or documents which are actual files. Sample Request URL: <http://.../?do=cnt.getservice&service=getRoomSet>

Parameter Options: :param *roomSetID: Numeric value of a Room Set.

getRoomsetSearchOptions ()

getRoomSetSearchOptions Returns an XML payload containing search options for the getRoomSetSearchResults service. The results of this call are used to obtain valid search values which can be passed to the 'getRoomSetSearchResults' request. Sample Request URL: <http://.../?do=cnt.getservice&service=getRoomSetSearchOptions> Sample XML Payload: See **REST_** getRoomSetSearchOptions.xml

getRoomsets (*postFlag=None*)

getRoomSets Returns an XML payload containing a full listing of all Room Set data. Sample Request URL: <http://.../?do=cnt.getservice&service=getRoomSets>

Parameter Options: :param postFlag: 0 or 1 (represents true or false).

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.scripts module

ohbm.settings module

ohbm.system module

system: part of the ohbm-api

class ohbm.system.**System** (*api=None*)

getEventTypes (*eventType=None*)

Returns an XML payload containing a listing of event types. Event type data contains the type name and roles that may be assigned to an event of this type. Sample Request URL: <http://.../?do=cnt.getservice&service=getEventTypes>

Parameter Options: :param eventType: Integer valid values: 1= Event, 2=Abstract If not provided the system defaults to sending Event roles.

getForm (*formID*)

Returns an XML payload containing form data. Form data includes form settings, questions and answers. Sample Request URL: <http://.../?do=cnt.getservice&service=getForm>

Parameter Options: :param *formID: Numeric value containing a valid form ID.

getLocations ()

Returns an XML payload containing a full listing of location and room data. Sample Request URL: <http://.../?do=cnt.getservice&service=getLocations>

getMeetingTime (*meetingTimeID*)

Returns an XML payload containing a meeting time/time range. This payload includes the meeting time's type, code, name, description, and start and end date. Sample Request URL: <http://.../?do=cnt.getservice&service=getMeetingTime>

Parameter Options: :param meetingTimeID: Numeric value containing a valid meeting time id.

getMeetingTimes ()

Returns an XML payload containing a listing of all active meeting times/time ranges. Sample Request URL: <http://.../?do=cnt.getservice&service=getMeetingTimes>

getMenu (*menuID*)

Returns an XML payload containing a menu item and its content pages. Sample Request URL: [http://.../?do=cnt.getservice&service=getMenu&{\[\]parameters list}](http://.../?do=cnt.getservice&service=getMenu&{[]parameters list})

Parameter Options: :param *menuID: Numeric value containing a valid menu id.

getPublicFields (*eventType=None*)

Returns an XML payload containing public fields for one of the following: Event, Exhibitor, Abstract, Speaker. Field data will contain the field ID and public label. Sample Request URL: <http://.../?do=cnt.getservice&service=getPublicFields>

Parameter Options: :param eventType: Integer valid values: 1= Event, 2=Exhibitor, 3=Abstract & 4=Speaker If not provided the system defaults to sending Event fields.

getRegistrationTypes ()

Returns an XML payload containing registration type data. Registration type data will contain the registration type name, code, description, and evaluation links. Note: The tag, AI_ATTENDEEID, may be used within data elements and serves as a placeholder to be replaced by a valid attendee ID. The tag, AI_EVENTID, may be used within data elements and serves as a placeholder to be replaced by a valid event ID. Sample Request URL: <http://.../?do=cnt.getservice&service=getRegistrationTypes>

getScheduledRoles (*eventType=None*)

Returns an XML payload containing roles used by Events or Abstracts. Role data will contain the role name, type, and public label. Sample Request URL: <http://.../?do=cnt.getservice&service=getScheduledRoles> :param eventType: Integer valid values: 1= Event, 2=Abstract If not provided the system defaults to sending Event roles.

get_result (*url, args=None*)

get_result takes a url and adds the apiKey for the get_result from utils :param url: the url to get the result for :param args: a dictionary of {"argumentName":value} to pass to the URL

ohbm.utils module

utils: part of the nidm-api general functions for the api

ohbm.utils.capitalize (*word*)

capitalize an input word. :param word: the string to capitalize

ohbm.utils.get_installdir ()

returns installation directory of nidm-api

ohbm.utils.get_url (*url*)

get_url returns a result from a url using requests module :param url: the url to retrieve

ohbm.utils.ordered_to_dict (*ordered_dict*)

ordered_to_dict converts an input ordered dict into a standard dict :param ordered_dict: the ordered dict

ohbm.utils.parse_item (*result, item*)

parse_item is a general function for looking one level into a result dictionary, for example returning result['event'] if it exists, where 'event' is the item. For function calls with two levels (['events']['event'] see parse_items. :param item: the name of the item (eg, event,exhibitor)

ohbm.utils.parse_items (*result, item*)

parse_items is a general function for looking TWO levels into a result dictionary, for example returning result['events']['event'] if it exists, where 'event' is the item. For function calls with one levels (['event'] see parse_item. :param item: the name of the item (eg, event,exhibitor)

Module contents

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

O

- `ohbm`, [17](#)
- `ohbm.abstracts`, [6](#)
- `ohbm.attendees`, [7](#)
- `ohbm.auth`, [10](#)
- `ohbm.categories`, [11](#)
- `ohbm.events`, [11](#)
- `ohbm.exhibitor`, [13](#)
- `ohbm.planner`, [14](#)
- `ohbm.roomset`, [15](#)
- `ohbm.system`, [16](#)
- `ohbm.utils`, [17](#)

A

Abstracts (class in ohbm.abstracts), 6
 addAttendeeActivity() (ohbm.attendees.Attendees method), 7
 Attendees (class in ohbm.attendees), 7
 Authenticate (class in ohbm.auth), 10
 authenticateBase() (ohbm.auth.Authenticate method), 10
 authenticateCompany() (ohbm.auth.Authenticate method), 10
 authenticatePerson() (ohbm.auth.Authenticate method), 10
 authenticateVendor() (ohbm.auth.Authenticate method), 11

C

capitalize() (in module ohbm.utils), 17
 Categories (class in ohbm.categories), 11

E

Events (class in ohbm.events), 11
 Exhibitor (class in ohbm.exhibitor), 13

G

get_installdir() (in module ohbm.utils), 17
 get_result() (ohbm.abstracts.Abstracts method), 7
 get_result() (ohbm.attendees.Attendees method), 10
 get_result() (ohbm.auth.Authenticate method), 11
 get_result() (ohbm.categories.Categories method), 11
 get_result() (ohbm.events.Events method), 13
 get_result() (ohbm.exhibitor.Exhibitor method), 14
 get_result() (ohbm.planner.Planner method), 15
 get_result() (ohbm.roomset.Roomset method), 15
 get_result() (ohbm.system.System method), 17
 get_url() (in module ohbm.utils), 17
 getAbstract() (ohbm.abstracts.Abstracts method), 6
 getAbstracts() (ohbm.abstracts.Abstracts method), 7
 getAbstractSearchOptions() (ohbm.abstracts.Abstracts method), 7

getAbstractSearchResults() (ohbm.abstracts.Abstracts method), 7
 getActivities() (ohbm.planner.Planner method), 14
 getActivity() (ohbm.planner.Planner method), 14
 getAttendeesCreditActivity() (ohbm.attendees.Attendees method), 8
 getAttendeesCreditActivitySummary() (ohbm.attendees.Attendees method), 8
 getAttendeesFormResponses() (ohbm.attendees.Attendees method), 8
 getAttendeesItineraryData() (ohbm.attendees.Attendees method), 8
 getAttendeesRegData() (ohbm.attendees.Attendees method), 9
 getCategories() (ohbm.categories.Categories method), 11
 getEvent() (ohbm.events.Events method), 11
 getEventCreditsByRegTypeSummary() (ohbm.events.Events method), 11
 getEventCreditSummary() (ohbm.events.Events method), 11
 getEvents() (ohbm.events.Events method), 12
 getEventsCreditsByRegTypeSummary() (ohbm.events.Events method), 12
 getEventsCreditsSummary() (ohbm.events.Events method), 13
 getEventSearchOptions() (ohbm.events.Events method), 11
 getEventSearchResults() (ohbm.events.Events method), 12
 getEventTypes() (ohbm.system.System method), 16
 getExhibitor() (ohbm.exhibitor.Exhibitor method), 13
 getExhibitors() (ohbm.exhibitor.Exhibitor method), 14
 getExhibitorSearchOptions() (ohbm.exhibitor.Exhibitor method), 13
 getExhibitorSearchResults() (ohbm.exhibitor.Exhibitor method), 14
 getForm() (ohbm.system.System method), 16
 getLocations() (ohbm.system.System method), 16
 getMeetingTime() (ohbm.system.System method), 16
 getMeetingTimes() (ohbm.system.System method), 16

`getMenu()` (ohbm.system.System method), 16
`getPersonActivities()` (ohbm.attendees.Attendees method), 9
`getPublicFields()` (ohbm.system.System method), 16
`getRegistrationTypes()` (ohbm.system.System method), 16
`getRoomset()` (ohbm.roomset.Roomset method), 15
`getRoomsets()` (ohbm.roomset.Roomset method), 15
`getRoomsetSearchOptions()` (ohbm.roomset.Roomset method), 15
`getRoomSetSearchResults()` (ohbm.roomset.Roomset method), 15
`getScheduledRoles()` (ohbm.system.System method), 16
`getSpeaker()` (ohbm.attendees.Attendees method), 9
`getSpeakers()` (ohbm.attendees.Attendees method), 9

O

ohbm (module), 17
ohbm.abstracts (module), 6
ohbm.attendees (module), 7
ohbm.auth (module), 10
ohbm.categories (module), 11
ohbm.events (module), 11
ohbm.exhibitor (module), 13
ohbm.planner (module), 14
ohbm.roomset (module), 15
ohbm.system (module), 16
ohbm.utils (module), 17
`ordered_to_dict()` (in module ohbm.utils), 17

P

`parse_item()` (in module ohbm.utils), 17
`parse_items()` (in module ohbm.utils), 17
Planner (class in ohbm.planner), 14

R

Roomset (class in ohbm.roomset), 15

S

System (class in ohbm.system), 16

U

`updateItinerary()` (ohbm.attendees.Attendees method), 10