
Meetup API Documentation

Release

Pat Ferate

Oct 28, 2017

1	Getting Started	3
1.1	Installation	3
1.2	Usage	3
1.2.1	Initialize Client	3
1.2.2	Execute API Calls	4
2	Module Details	5
2.1	Meetup API Module	5
2.1.1	API Classes	5
2.1.2	API Client Details	6
2.2	Meetup Exceptions Module	39
2.2.1	Exception Classes	39
3	Testing	41
3.1	Continuous Integration	42
4	Changelog	43
4.1	v0.1.1 2016-03-03	43
4.2	v0.1 2016-02-04	43
5	License	45
5.1	The MIT License (MIT)	45
6	Indices and tables	47
	Python Module Index	49

meetup-api is a Python client for [Meetup's RESTful API](#). It is compatible with both Python 2 and Python 3. Currently only API Key authentication is available (OAuth is planned for a future release).

This document will show you how to get up and running with the Meetup API Python Client.

Installation

Meetup API requires the following packages installed:

- requests
- six

Assuming you have [Python](#) already, install the package using pip:

```
$ pip install meetup-api
```

Usage

Initialize Client

To initialize your Meetup API Client, you will need to import the Client class and create a Client object. Before making and API requests, you will need to assign your API key to the object.

Three ways to assign your API key (in order of precedence):

1. Assign to attribute:

```
>>> import meetup.api
>>> client = meetup.api.Client()
>>> client.api_key = 'my_special_api_key_value'
```

2. Assign at initialization:

```
>>> import meetup.api
>>> client = meetup.api.Client('my_special_api_key_value')
```

3. Retrieved from environment variable:

```
$ export MEETUP_API_KEY=my_special_api_key_value
```

```
>>> import meetup.api
>>> client = meetup.api.Client()
```

Execute API Calls

```
>>> import meetup.api
>>> client = meetup.api.Client('my_special_api_key_value')
>>> group_info = client.GetGroup({'urlname': 'Meetup-API-Testing'})
>>>
>>> type(client)
<class 'meetup.api.Client'>
>>>
>>> type(group_info)
<class 'meetup.api.MeetupObject'>
>>>
>>> group_info.__dict__.keys()
dict_keys(['who', 'join_mode', 'link', 'created', 'country', 'name', 'id', 'visibility',
↪',
           'state', 'urlname', 'city', 'lat', 'timezone', 'members', 'lon',
↪'description',
           'organizer', 'category', 'next_event', 'group_photo'])
>>>
>>> group_info.id
1556336
>>>
>>> group_info.name
'Meetup API Testing Sandbox'
>>>
>>> group_info.link
'http://www.meetup.com/Meetup-API-Testing/'
```

A full listing of implemented API methods can be found at *Meetup API Module*.

Description of modules

Meetup API Module

- *API Classes*
- *API Client Details*
 - *API Client Method Index*
 - *API Client Methods*

API Classes

class `meetup.api.Client` (*api_key=None, api_url='http://api.meetup.com/', overlimit_wait=True*)
Meetup API Client.

There are 3 options for defining the API key prior to making API calls:

1. Pass it as a parameter (`api_key`)
2. Stored as an environment variable, if parameter is not defined. (Default: `MEETUP_API_KEY`)
3. Define it after the object is created. (`client.api_key = 'my_secret_api_key'`)

Parameters

- **api_key** – Meetup API Key, from https://secure.meetup.com/meetup_api/key/
- **api_url** – Meetup API URL, Keeping it flexible so that it can be generalized in the future.

- **overlimit_wait** – Whether or not to wait and retry if over API request limit. (Default: True)

class `meetup.api.MeetupObject` (**initial_data, **kwargs*)

Generic Meetup Object generated from dict and keyword arguments.

Key/Values from dict are accessible from object as attributes (e.g. `object.key`) Keyword arguments passed at initialization are also accessible in the same way. Keyword values overwrite values from dict.

Parameters

- **initial_data** – Initial values in a dict
- **kwargs** – Additional key/values to set

class `meetup.api.MeetupObjectList` (*initial_list*)

A custom, iterative list for MeetupObjects.

Items are stored as raw, jsonified API Responses. Items are converted to MeetupObjects on the fly.

class `meetup.api.RateLimit`

Rate limit information, as defined by Meetup. This data is received in the response header.

Attribute	HTTP Header	Description
limit	X-RateLimit-Limit	The maximum number of requests that can be made in a window of time
remaining	X-RateLimit-Remaining	The remaining number of requests allowed in the current rate limit window
reset	X-RateLimit-Reset	The number of seconds until the current rate limit window resets

limit = None

remaining = None

reset = None

API Client Details

The following are dynamically generated methods for the `meetup.api.Client` class.

API Client Method Index

- `CreateEvent()`
- `CreateEventComment()`
- `CreateEventCommentFlag()`
- `CreateEventRating()`
- `CreateGroupAbuseReports()`
- `CreateGroupMemberApprovals()`
- `CreateGroupPhoto()`
- `CreateGroupTopics()`
- `CreateGroupVenues()`
- `CreateMemberPhoto()`
- `CreateNotificationsRead()`
- `CreatePhoto()`
- `CreatePhotoAlbum()`
- `CreatePhotoComment()`
- `CreateProfile()`
- `CreateRecommendedGroupsIgnores()`

- *CreateRsvp ()*
- *CreateSelfAbuseReports ()*
- *CreateSelfBlocks ()*
- *DeleteEvent ()*
- *DeleteEventComment ()*
- *DeleteEventCommentLike ()*
- *DeleteEventCommentSubscribe ()*
- *DeleteGroupEventsWatchlist ()*
- *DeleteGroupMemberApprovals ()*
- *DeleteGroupTopics ()*
- *DeleteMemberPhoto ()*
- *DeletePhoto ()*
- *DeleteProfile ()*
- *DeleteSelfBlocks ()*
- *EditEvent ()*
- *EditEventCommentLike ()*
- *EditEventCommentSubscribe ()*
- *EditGroup ()*
- *EditGroupEventsAttendance ()*
- *EditGroupEventsPayments ()*
- *EditGroupEventsWatchlist ()*
- *EditMember ()*
- *EditProfile ()*
- *GetActivity ()*
- *GetCategories ()*
- *GetCities ()*
- *GetComments ()*
- *GetConcierge ()*
- *GetDashboard ()*
- *GetEvent ()*
- *GetEventComment ()*
- *GetEventCommentLikes ()*
- *GetEventComments ()*
- *GetEventRatings ()*
- *GetEvents ()*
- *GetFindGroups ()*
- *GetGroup ()*
- *GetGroupBoards ()*
- *GetGroupBoardsDiscussions ()*
- *GetGroupEventsAttendance ()*
- *GetGroupSimilarGroups ()*
- *GetGroupVenues ()*
- *GetGroups ()*
- *GetMember ()*
- *GetMembers ()*
- *GetNotifications ()*
- *GetOembed ()*
- *GetOpenEvents ()*
- *GetOpenVenues ()*
- *GetPhotoAlbums ()*
- *GetPhotoComments ()*
- *GetPhotos ()*
- *GetProfile ()*
- *GetProfiles ()*

- *GetRecommendedGroupTopics ()*
- *GetRecommendedGroups ()*
- *GetRecommendedVenues ()*
- *GetRsvp ()*
- *GetRsvps ()*
- *GetSelfBlocks ()*
- *GetStatus ()*
- *GetTopicCategories ()*
- *GetTopics ()*
- *GetVenues ()*

API Client Methods

`class meetup.api.Client`

CreateEvent (*description, duration, email_reminders, group_id, group_urlname, guest_limit, host_instructions, hosts, how_to_find_us, name, publish_status, question_{index}, rsvp_alerts, rsvp_close, rsvp_limit, rsvp_open, simple_html_description, time, venue_id, venue_visibility, waitlisting, why*)

All required parameters must be supplied. This method requires parameters encoded in application/x-www-form-urlencoded format as an HTTP POST.

Survey questions may be submitted in the format “question_{index}=questiontext” where “index” is an integer representing the order the questions should be presented in. e.g. question_0=question1&question_1=question2.

Organizers of the hosting group may optionally save this event as a draft by setting “publish_status” to “draft”. Drafts are then discoverable using the [/2/events]/(meetup_api/docs/2/events) API setting the “status” request parameter to “draft”

URI: /2/event

API Version: 2

Parameters

- **description** – Longer description of the event, in HTML. May not be longer than 50000 characters.
- **duration** – Event duration in milliseconds. When not specified, a default of 3 hours may be assumed by applications. To clear event duration, set this to 0
- **email_reminders** – Limited to organizers and event hosts, if false, disables event reminders
- **group_id** – Group hosting the event
- **group_urlname** – URL name of the Group hosting the event
- **guest_limit** – Number of guests members may include in their RSVP, 0 or more
- **host_instructions** – optional set of instructions to provide to hosts, only used when hosts are provided
- **hosts** – Up to 5 comma-separated valid member ids to be hosts for the event. If hosts is not provided, the authorized member is the default host
- **how_to_find_us** – The information provided by the event host for “How will members find you there?”. Visible when location is visible to the authenticated member
- **name** – Name of the event. May not be longer than 80 characters.

- **publish_status** – If you are an organizer of the group, you may set this to “draft” to save the event as a draft. Doing so will require a `status=draft` filter on `/2/event` queries.
- **question_{index}** – Those with permission may include up to 6 survey questions for the event with each being up to 250 characters. See the parameter notes section for more information
- **rsvp_alerts** – Limited to organizers and event hosts, if false, disables member RSVP alerts
- **rsvp_close** – Users with permission may set the RSVP close time for the event. The time may be specified in milliseconds since the epoch, or relative to the current time in the `d/w/m` format.
- **rsvp_limit** – Total number of RSVPs available for the event
- **rsvp_open** – Users with permission may set the RSVP open time for the event. The time may be specified in milliseconds since the epoch, or relative to the current time in the `d/w/m` format.
- **simple_html_description** – Description of the event, in simple HTML format. This value is translated to HTML to update the description. May not be longer than 50000 characters.
- **time** – Event start time in milliseconds since the epoch, or relative to the current time in the `d/w/m` format.
- **venue_id** – Numeric identifier of a venue
- **venue_visibility** – Controls the visibility of the event venue for non members of the hosting group. May be one of “public” or “members”
- **waitlisting** – Waiting list status may be one of: auto, manual, off
- **why** – We should do this because... May not be longer than 250 characters.

CreateEventComment (*comment, event_id, in_reply_to, notifications*)

This method posts messages that appear under “Talk about this Meetup”. All of the required parameters must be supplied with the request, which must be an HTTP POST. The currently authenticated user must be a member of the group hosting the event

URI: `/2/event_comment`

API Version: 2

Parameters

- **comment** – The comment text
- **event_id** – The event related to this comment.
- **in_reply_to** – If this comment is a reply, the ID of the comment being replied to
- **notifications** – Notification control for authorized member on this comment thread. “on” will result in notifications being sent. “off” will opt the member out of notifications for this comment thread. Defaults to “on” unless the member previous opted out of notifications on the thread.

CreateEventCommentFlag (*comment_id, reason*)

This method creates a spam report for comment content All required parameters must be supplied.

URI: `/2/event_comment_flag`

API Version: 2

Parameters

- **comment_id** – The id of the comment
- **reason** – Reason for flagging the comment. May be one of inappropriate, spam

CreateEventRating (*attendee_count, event_id, rating*)

This method allows members to posts rating for an event after it's occurred. Only permitted for members who rsvp'd "yes" or "maybe" to the event

URI: /2/event_rating

API Version: 2

Parameters

- **attendee_count** – The number of attendees for the event (organizers/assistant organizers/co-organizers/event organizers/event hosts only)
- **event_id** – The ID of the event to fetch ratings data for
- **rating** – The member's rating (either 1, 2, 3, 4, or 5)

CreateGroupAbuseReports (*urlname, type*)

Submits a new abuse report for a target group. Abuse reports will be followed up on by our Community support team. This method requires the *reporting* scope for oauth-authenticated requests

URI: /{urlname}/abuse_reports

API Version: 3

Parameters

- **urlname** (*required*) – None
- **type** – A required identifier for type of abuse you are reporting. Acceptable values include *graphic_content*, *harmful_activities*, *licensed_services*, *not_community*, *nudity*, *other*, *promotion_focus*, *violence*

CreateGroupMemberApprovals (*urlname, member, send_copy, welcome_message*)

Approves one or more requests for group membership The *:urlname* path element may be any valid group urlname or domain name. If you need access to your group's welcome message, you can access it from the [/2/groups]/meetup_api/docs/2/groups/#response) method, providing a value of *welcome_message* for the fields parameter. To get a list of pending members, as an organizer, you can request the status *pending* in the [/2/profiles]/meetup_api/docs/2/profiles/#params) method.

URI: /{urlname}/member/approvals

API Version: 3

Parameters

- **urlname** (*required*) – None
- **member** – Comma-delimited numeric pending member IDs. The maximum allowed is 200
- **send_copy** – Optional Boolean value indicating whether or not the org should receive a copy of the message sent to the approved members
- **welcome_message** – Optional message to send to the members being approved. If not provided, the groups default welcome message will be sent. Max message size is 2000

CreateGroupPhoto (*await, group_id, group_urlname, main, photo*)

Uploads a new Meetup Group photo. To change other Group settings use the [Group Edit](/meetup_api/docs/:urlname/#edit) endpoint The request must be a POST of content-type multipart/form-data. The supplied photo will be added to the provided groups general photo album. If authenticating with OAuth, no parameters in the multipart form data should be included in the signature base string. You must be an organizer of the group in order to load a photo.

URI: /2/group_photo

API Version: 2

Parameters

- **await** – If true, this ensures a response will not be returned until the upload is accessible
- **group_id** – Group ID for the target group. This may be used as an alternative to group_urlname
- **group_urlname** – Group urlname. This may be used as an alternative to group_id
- **main** – Set to ‘true’ to have this photo become the group’s main photo. Set it to ‘false’ otherwise. Defaults to true
- **photo** – The photo, encoded as multipart/form-data. The maximum file size allowed is 10MB

CreateGroupTopics (*urlname, topic_id*)

Associates topics with a given Meetup Group. Limited to organizers of the group. OAuth authenticated requests require an additional [group_edit](/meetup_api/auth/#oauth2-scopes) permission. A group can have at most 15 topics

URI: /{urlname}/topics

API Version: 3

Parameters

- **urlname** (*required*) – None
- **topic_id** – Comma-delimited list of topic ids to associate with group

CreateGroupVenues (*urlname, address_1, address_2, city, country, hours, name, phone, state, visibility, web_url*)

Interface for creating new Meetup venues The :urlname path element may be any valid group urlname or domain name. The required parameters are highlighted below. To avoid creating duplicate Meetup venues, this endpoint will attempt to resolve potentially similar venues based on name and address before creating a new public venues. If potential matches are detected, this method will return a 409 Conflict response containing a list of potential matches

URI: /{urlname}/venues

API Version: 3

Parameters

- **urlname** (*required*) – None
- **address_1** – Primary address of the venue
- **address_2** – Secondary address info
- **city** – City name of the venue
- **country** – 2 character country code of the venue
- **hours** – Open hours information about the venue

- **name** – Unique name of the venue
- **phone** – Optional phone number for the venue
- **state** – If in the US or CA, the state code for the venue
- **visibility** – Optional value indicating the venues visibility to others. May be one of private or public. Defaults to 'public'
- **web_url** – Optional web url for the venue

CreateMemberPhoto (*await, main, photo, sync_matching_photo, sync_photo*)

Uploads a photo to be associated with a Member The request must be a POST of content-type multipart/form-data. The supplied photo will be added to the currently authenticated user's member profile. If authenticating with OAuth, no parameters in the multipart form data should be included in the signature base string.

URI: /2/member_photo

API Version: 2

Parameters

- **await** – If true, this ensures a response will not be returned until the upload is accessible
- **main** – Set to "true" to have this photo become the member's main profile photo. Otherwise, it will become the main photo only when none other is selected. If the authenticated member does not already have a main profile photo set it will remain so.
- **photo** – The photo, encoded as multipart/form-data. The maximum file size allowed is 10MB
- **sync_matching_photo** – When set to true and main is set to true, this will replace all group profile photos matching the current photo with the provided replacement
- **sync_photo** – When set to true, this parameter will sync all of the group profile photos for the member with the provided photo_id

CreateNotificationsRead (*fields, since_id*)

Marks groups of [notifications](/meetup_api/docs/notifications/) as read.

URI: /notifications/read

API Version: 3

Parameters

- **fields** – Request that additional fields (separated by commas) be included in the output.
- **since_id** – The id of the newest notification item, typically the first in the list returned by the notifications endpoint

CreatePhoto (*await, caption, event_id, photo, photo_album_id*)

Uploads a photo for a given event A photo album or event ID must be supplied with the request, which must be a POST of content-type *multipart/form-data*. The currently authenticated user must be a member of the group that is hosting the event. Parameters other than photo may be passed in as a query string or within the posted content. If authenticating with OAuth, no parameters in the multipart form data should be included in the signature base string

URI: /2/photo

API Version: 2

Parameters

- **await** – If true, this ensures a response will not be returned until the upload is accessible
- **caption** – Caption for the photo
- **event_id** – Identifier of an event. If there is no album for this event, one will be created.
- **photo** – The photo, encoded as multipart/form-data. The maximum file size allowed is 10MB
- **photo_album_id** – Identifier of an existing photo album, which may be an event or group album

CreatePhotoAlbum (*group_id, title*)

This method creates photo albums within a Meetup group All of the required parameters must be supplied with the request, which must be an HTTP POST. The currently authenticated user must be an organizer (main, assistant, or co-) of the specified group

URI: /2/photo_album

API Version: 2

Parameters

- **group_id** – Group to create the album in
- **title** – Title of the new album

CreatePhotoComment (*comment, photo_id*)

This method posts comments that appear below photos All of the required parameters must be supplied with the request, which must be an HTTP POST. The currently authenticated user must be a member of the group hosting the photo

URI: /2/photo_comment

API Version: 2

Parameters

- **comment** – The comment text
- **photo_id** – The photo related to this comment.

CreateProfile (*answer_{qid}, group_id, group_urlname, intro, new_photo, photo_id, site_name, site_url*)

This method allows an authenticated member to join a group by creating a profile This method requires an HTTP POST. All required parameters must be supplied. An intro and answers may be required based on the group the member is joining. To find out if a group requires an intro or answers to questions, query for the group through one of the [Groups methods](/meetup_api/docs/2/groups) providing setting the `__fields__` parameter to `__join_info__` and inspecting the `__join_info__` in the results. Answers to the questions must be named using the convention `__answer_{question_id}__`.

URI: /2/profile

API Version: 2

Parameters

- **answer_{qid}** – Answers to questions from groups API join_info question fields
- **group_id** – Id of group to join
- **group_urlname** – Urlname of group to join
- **intro** – Provides a Member an opportunity to tell the group about themselves

- **new_photo** – file upload for a new member photo
- **photo_id** – photo_id of the photo to use for this profile
- **site_name** – Name of member’s site. Max length is 32
- **site_url** – Link to member’s site. Max length is 80

CreateRecommendedGroupsIgnores (*urlname*)

Provides a form of feed back by requesting to remove a group from future recommendations Only the path parameter :urlname is required. You are limited to 100 of these requests in a 24 hour period of time.

URI: /recommended/groups/ignores/{urlname}

API Version: 3

Parameters *urlname* (*required*) – None

CreateRsvp (*agree_to_refund*, *answer_{qid}*, *comments*, *event_id*, *guests*, *member_id*, *opt_to_pay*, *rsvp*)

Creates or updates an existing RSVP At least one of the required parameter(s) must be supplied with the request.

The RSVP is recorded for the currently authenticated member account, unless a *member_id* is supplied, in which case the authenticated member must have permissions to edit the event or a host.

Member’s RSVP’ing as themselves may supply answers to a Meetup event’s survey questions by supplying answers as request parameters corresponding to question ids in the format *answer_{question_id}*. To discover available Meetup event survey questions, supply the [Events 2](/meetup_api/docs/2/events/) method with the ‘fields’ parameter set to ‘survey_questions’.

Repeated attempts to RSVP to the same event given the same member credentials will result in an update to previous RSVP. The state of the event may or may not allow for this update to occur depending on the organizer-defined constraints and pre-requisites for the event.

Note: waitlist responses may be coerced into yes responses when there is space available and yes responses may be coerced into waitlist response when space is not.

If the event requires payment you are required to send an “agree_to_refund” parameter set to the true or false. This represents the authorized members agreement to understanding the event’s refund policy.

URI: /2/rsvp

API Version: 2

Parameters

- **agree_to_refund** – For events with fees, the authorized member must agree to the event’s refund policy. This must be set to either true or false
- **answer_{qid}** – Answers to event survey questions. Answers may not be longer than 250 characters. Organizers and hosts my not edit or create answers on behalf of members
- **comments** – A comment to post along with the RSVP
- **event_id** – The event that you are RSVP’ing to
- **guests** – Number of guests also coming to the event.
- **member_id** – Organizers and event hosts may RSVP on behalf of a member by specifying an ID here. As when editing RSVPs on the site, organizers may enter a “yes” for a member even if the event requires payment.
- **opt_to_pay** – For events with fees, the authorized member may opt to pay as part of the RSVP request. This may be set to true or false

- **rsvp** – The RSVP setting - value must be either “yes”, “no” or “waitlist”

CreateSelfAbuseReports (*comments, content_tag, member_id, type, url*)

Submits a new abuse report for a target member. Abuse reports will be followed up on by our Community support team. This method requires the oauth *reporting* scope for oauth-authenticated requests

URI: /self/abuse_reports

API Version: 3

Parameters

- **comments** – An optional string of text that describes why you are submitting this report
- **content_tag** – An optional identifier for flagged content that identifies both the type and id, where possible, of the content reported. The type and id should be separated by a ‘:’ character, e.g *event_comment:{event_comment_id}*. Valid content_tag types include boards, chapter, convo, convo_message, email, event_comment, event_photo, member_photo, ml, reply. If member_id is not provided, content_tag should be provided to imply member_id
- **member_id** – A numeric identifier for the member being reported. If not provided, this will be inferred by the author of the content associated with the provided *content_tag*
- **type** – A required identifier for type of abuse you are reporting. Acceptable values include abuse, copyright, dangerous, fake, harass, inappropriate, join, other, photo, spam
- **url** – An optional URL for the location of the reported content if one exists

CreateSelfBlocks (*member_id, comments, report*)

Blocks a target member from various interactions with the authenticated member on the platform A valid member_id path parameter for the target member is required. This method requires the oauth *reporting* scope for oauth-authenticated requests

URI: /self/blocks/{member_id}

API Version: 3

Parameters

- **member_id** (*required*) – None
- **comments** – An optional string of text describing why you have chosen to block this member
- **report** – An optional value that represents a type of abuse the target member is being blocked for. Acceptable values include one of the following: fake, harass, inappropriate, spam

DeleteEvent (*id*)

Deletes a specified meetup Only authorization parameters are needed.

URI: /2/event/{id}

API Version: 2

Parameters *id* (*required*) – None

DeleteEventComment (*id, fields*)

Delete a single event comment or reply Only authorization parameters are needed.

URI: /2/event_comment/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **fields** – comma-separate list of optional fields

DeleteEventCommentLike (*id*)

Unlike a given Event comment Only authorization parameters are needed. :id in the path should be the id of the comment you are unliking

URI: /2/event_comment_like/{id}

API Version: 2

Parameters **id** (*required*) – None

DeleteEventCommentSubscribe (*id*)

Unsubscribe to notifications for updates to a given comment thread Only authorization parameters are needed. :id in the path should be the id of comment being replied to which you are unsubscribing to notifications from

URI: /2/event_comment_subscribe/{id}

API Version: 2

Parameters **id** (*required*) – None

DeleteGroupEventsWatchlist (*id, urlname*)

Remove yourself from an event watch list No parameters are required. You should only call this method if you request the fields parameter ‘self’ in any events methods and get back a self.actions field containing ‘unwatch’

URI: /{urlname}/events/{id}/watchlist

API Version: 3

Parameters

- **id** (*required*) – None
- **urlname** (*required*) – None

DeleteGroupMemberApprovals (*urlname, anon, ban, explanation, member, send_copy*)

Declines one or more requests for group membership The :urlname path element may be any valid group urlname or domain name. To get a list of pending members, as an organizer, you can request the status *pending* in the [/2/profiles]/(meetup_api/docs/2/profiles/#params) method.

URI: /{urlname}/member/approvals

API Version: 3

Parameters

- **urlname** (*required*) – None
- **anon** – Optional Boolean value indicating whether your email should be revealed to the members. Default is false.
- **ban** – Optional Boolean value indicating whether or not to ban the member in the future. Default is false
- **explanation** – Optional explanation to send to the members being declined. Max message size is 2000
- **member** – Comma-delimited numeric pending member IDs. The maximum allowed is 200

- **send_copy** – Optional Boolean value indicating whether or to send a copy to the member issuing the decline. Default is true

DeleteGroupTopics (*urlname, topic_id*)

Disassociates topics with a given Meetup Group. Limited to organizers of the group. OAuth authenticated requests require an additional [group_edit](/meetup_api/auth/#oauth2-scopes) permission. Groups must have at one topic. Attempts to remove all topics will result in a failed request

URI: /{urlname}/topics

API Version: 3

Parameters

- **urlname** (*required*) – None
- **topic_id** – Comma-delimited list of topic ids to disassociate with group

DeleteMemberPhoto (*id*)

Delete the specified member photo Only authorization parameters are needed. Authorized user must be the member owning the photo

URI: /2/member_photo/{id}

API Version: 2

Parameters **id** (*required*) – None

DeletePhoto (*id*)

Delete specified event photo Only authorization parameters are needed. Authorized user must be the original poster of the photo or the organizer of the group the photo was posted in

URI: /2/photo/{id}

API Version: 2

Parameters **id** (*required*) – None

DeleteProfile (*gid, mid, exit_comment*)

Deletes a member's group profile All parameters are optional. Only the authorized member is permitted leave groups they do not organize. This method may not be used by organizers to remove members.

URI: /2/profile/{gid}/{mid}

API Version: 2

Parameters

- **gid** (*required*) – None
- **mid** (*required*) – None
- **exit_comment** – Optional message to the organizer when leaving

DeleteSelfBlocks (*member_id*)

Unblocks a previously blocked member from various interactions with the authenticated member on the platform A valid member_id path parameter for the target member is required. This method requires the oauth *reporting* scope for oauth-authenticated requests.

URI: /self/blocks/{member_id}

API Version: 3

Parameters **member_id** (*required*) – None

EditEvent (*id, announce, announce_message, description, duration, email_reminders, group_id, guest_limit, host_instructions, hosts, how_to_find_us, lat, lon, name, publish_status, question_edit_{id}, question_{index}, rsvp_alerts, rsvp_close, rsvp_limit, rsvp_open, simple_html_description, time, venue_id, venue_visibility, waitlisting, why*)

Update an existing Meetup All parameters are optional. This method requires an HTTP POST.

If a lat & lon are provided, the event's venue will be updated accordingly.

Survey questions may be submitted in the format "question_{index}=questiontext" where "index" is an integer representing the order the questions should be presented in. e.g. question_0=question1&question_1=question2.

To view existing survey questions supply the request parameter fields=survey_questions with the [Event Get](#get) To edit questions you may submit questions in the format "question_edit_{id}=updatedquestion" where id is the id of the question. To clear an existing question for the event survey, just submit question_edit_{id} with an empty value.

Organizers may publish a draft event by posting with the "publish_status" request parameter set to "published". Organizers and hosts of the event may also optionally announce an unannounced event by posting with the "announce" request parameter set to "true"

URI: /2/event/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **announce** – Organizers and hosts may set this to true to announce a Meetup.
- **announce_message** – Organizers and hosts may provide an optional message in this field when announcing a Meetup. May not be longer than 500 characters.
- **description** – Longer description of the event, in HTML. May not be longer than 50000 characters.
- **duration** – Event duration in milliseconds. When not specified, a default of 3 hours may be assumed by applications. To clear event duration, set this to 0
- **email_reminders** – Limited to organizers and event hosts, if false, disables event reminders
- **group_id** – Group to hold the event
- **guest_limit** – number of guests members may include in their RSVP, 0 or more
- **host_instructions** – optional set of instructions to provide to hosts, only used when hosts are provided
- **hosts** – Up to 5 valid member ids to be hosts for the event.
- **how_to_find_us** – The information provided by the event host for "How will members find you there?". Visible when location is visible to the authenticated member
- **lat** – Updates to the venue's latitude and longitude. When present, both must be provided
- **lon** – Updates to the venue's latitude and longitude. When present, both must be provided
- **name** – Event name. May not be longer than 80 characters.
- **publish_status** – If you are an organizer of the group, you may set this to "draft" or "published". Setting state to "draft" will require a status=draft filter on /2/event queries.

- **question_edit_{id}** – Those with permission may include up to 6 survey questions with each being up to 250 characters, including new questions. To delete a question submit this parameter with an empty value
- **question_{index}** – Those with permission may include up to 6 survey questions with each being up to 250 characters, including edited questions, for the event. See the parameter notes section for more information
- **rsvp_alerts** – Limited to organizers and event hosts, if false, disables member RSVP alerts
- **rsvp_close** – Users with permission may set the RSVP close time for the event. The time may be specified in milliseconds since the epoch, or relative to the current time in the d/w/m format. To unset rsvp_close, set this to 0
- **rsvp_limit** – Total number of RSVPs available for the event. To remove this limit, set this to 0
- **rsvp_open** – Users with permission may set the RSVP open time for the event. The time may be specified in milliseconds since the epoch, or relative to the current time in the d/w/m format.
- **simple_html_description** – Description of the event, in simple HTML format. This value is translated to HTML to update the description. May not be longer than 50000 characters.
- **time** – event start time in milliseconds since the epoch, or relative to the current time in the d/w/m format.
- **venue_id** – Numeric identifier of a venue. To unset the event’s venue, set this to 0
- **venue_visibility** – Controls the visibility of venue. May be one of “public” or “members”
- **waitlisting** – Waiting list status may be one of: auto, manual, off
- **why** – We should do this because... May not be longer than 250 characters.

EditEventCommentLike (*id*)

Like a given Event comment Only authorization parameters are needed. :id in the path should be the id of the comment you are liking

URI: /2/event_comment_like/{id}

API Version: 2

Parameters *id* (*required*) – None

EditEventCommentSubscribe (*id*)

Subscribe to notifications on updates to a given comment thread Only authorization parameters are needed. :id in the path should be id of comment being replied to which you are subscribing to notifications for

URI: /2/event_comment_subscribe/{id}

API Version: 2

Parameters *id* (*required*) – None

EditGroup (*add_topics*, *country*, *description*, *dryrun*, *ga_code*, *join_mode*, *list_addr*, *list_mode*, *name*, *photo_req*, *question_edit_{id}*, *question_{index}*, *questions_req*, *remove_topics*, *urlname*, *visibility*, *welcome_message*, *who*, *zip*, *{service}_uri*)

Allows organizers to edit their Meetup group information. To change group topics, see the [add](/meetup_api/docs/:urlname/topics/#add) and [remove](/meetup_api/docs/:urlname/topics/#remove) topics endpoints. To change group photo use the [Group photo upload](/meetup_api/docs/2/group_photo/#create) endpoint. OAuth authenticated requests require an additional [group_edit](/meetup_api/auth/#oauth2-scopes) permission. All parameters are optional.

URI: /{urlname}

API Version: 3

Parameters

- **add_topics** – Comma-delimited list of topic ids to associate with group
- **country** – The ISO_3166-1 country code for the country which contains the city
- **description** – Summary of what the Meetup group is about in simple HTML format
- **dryrun** – Boolean parameter that will cause this endpoint to apply all validation rules without actually saving changes in which case the response will only reflect the group's current attributes
- **ga_code** – Google Analytics code for group
- **join_mode** – Controls how member's are let into the group. May be one of 'open' meaning any Meetup member may join, 'closed' meaning group is not currently accepting new members, or 'approval' meaning members must be approved by an organizer. Note, the 'closed' options is only available to groups that already have a 'closed' join_mode
- **list_addr** – Mailing list prefix. By default this is the Group's urlname.
- **list_mode** – Defines policy for who can post to the group mailing list. May be one of 'open' meaning any Member can post, 'off' meaning no one can post, 'moderated' meaning messages must be approved, or 'orgs_only' meaning only organizers may post to the list
- **name** – Display name of the group. Can be at most 60 characters
- **photo_req** – Indicates that a member must provide a photo before joining. Expects true or false values
- **question_edit_{id}** – Edits a current profile question identified by an id in the parameter name. The index updated index should also be encoded in the parameter name. To delete a question, set this to an empty string. Groups that require profile questions must have at least one question
- **question_{index}** – A new profile question defined in the order of index provided in the request parameter name
- **questions_req** – Indicates that provide questions are required before joining. Expects true or false values
- **remove_topics** – Comma-delimited list of topic ids to disassociate with group
- **urlname** – Name used for the groups web address on meetup.com. Must be between 6 and 60 characters
- **visibility** – Restricts group visibility for non-members. May be one of 'public', 'public_limited' or 'members'. Note, the 'members' option is only available to groups that already have 'members' visibility
- **welcome_message** – Message sent to members after they join. Can be at most 2000 characters
- **who** – What members of the group will be called. Can be at most 32 characters

- **zip** – The ZIP code of the city
- **{service}_uri** – A URI for a social network service. Service must be one of facebook, flickr, linkedin, other, tumblr, twitter

EditGroupEventsAttendance (*id, urlname, guests, headcount, member, status*)

Takes Member attendance for an Event. Limited for use by administrative members. The `:urlname` path element may be any valid group urlname or domain name. The `:id` path element must be a valid alphanumeric Meetup event identifier. Highlighted fields are required.

URI: `/{urlname}/events/{id}/attendance`

API Version: 3

Parameters

- **id** (*required*) – None
- **urlname** (*required*) – None
- **guests** – The number of guests accompanying member. Maximum of 99 is allowed.
- **headcount** – Sets the overall headcount for the event. This may not necessarily correlate with the list of attendees in this group if the event is part of a joint Meetup event
- **member** – A valid ID of member in the group
- **status** – An attendance status for the member. Must be one of: noshow, absent, attended

EditGroupEventsPayments (*id, urlname, amount, member, paid_on, quantity*)

Allows organizers of a group to note payments made by members for an event. This is the ‘Mark Paid’ feature seen in the RSVP listings on event details pages and affects the ‘pay_status’ response fields in `[2/rsvps]/(meetup_api/docs/2/rsvps/#response)` for paid events Only organizer may submit payment information for members and only one payment may be submitted for a member for a given event

URI: `/{urlname}/events/{id}/payments`

API Version: 3

Parameters

- **id** (*required*) – None
- **urlname** (*required*) – None
- **amount** – The monetary amount of money the member submitted
- **member** – Member Id of member who made a payment
- **paid_on** – The time the payment was made in milliseconds from the epoch. Defaults to now
- **quantity** – The number of payments made. Defaults to 1

EditGroupEventsWatchlist (*id, urlname*)

Add yourself to an event watch list to get notified when a spot becomes available No parameters are required. You should only call this method if you request the fields parameter ‘self’ in any events methods and get back a `self.actions` field containing ‘watch’

URI: `/{urlname}/events/{id}/watchlist`

API Version: 3

Parameters

- **id** (*required*) – None
- **urlname** (*required*) – None

EditMember (*id, add_topics, bio, bio_privacy, birthday, city, city_id, country, facebook_privacy, gender, groups_privacy, hometown, lang, lat, lon, messaging_pref, name, photo_id, photos_privacy, radius, remove_topics, sync_photo, topics_privacy, zip*)

Edit the authorized member's attributes This method requires tls. Only the authorized user may edit their own properties. A city may be provided by either a city_id or a combination of lat and lon

URI: /2/member/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **add_topics** – Comma-delimited list of topics ids to add to your alert list
- **bio** – Free form text passage about you. must be less than 250 characters
- **bio_privacy** – Controls the visibility of the member's bio. May be one of visible, hidden
- **birthday** – Day you were born. Format should be in the form of yyyy or mmddyyyy. A value of -1 indicates that birthday data should be cleared.
- **city** – City name for your location
- **city_id** – Valid city id from /2/cities method
- **country** – Valid country code for your location
- **facebook_privacy** – Controls the visibility of the member's facebook connection. May be one of visible, hidden. If the member has not connected their Facebook account, attempts to set this preference will do nothing.
- **gender** – Your gender (used for better recommendations). Valid values are be one of other, none, female, male
- **groups_privacy** – Controls the visibility of the member's groups. May be one of visible, hidden
- **hometown** – Hometown of member. Can not be longer than 64 characters
- **lang** – Language preference used on the site. Valid values are en_us, de, es, fr, it, pt
- **lat** – latitude of city
- **lon** – longitude of city
- **messaging_pref** – This specifies the member's preference for being contacted from members on the site. Possible values are "orgs_only" meaning only group organizers can contact you, "groups_only" meaning only members of your group can contact you, or "all_members" meaning all members may contact you.
- **name** – The name of the current member
- **photo_id** – A valid photo_id from the member's photos to set as the main profile photo. A value of 0 will unset the current photo
- **photos_privacy** – Controls the visibility of the member's photos. May be one of visible, hidden
- **radius** – radius, in miles to search for city given a lat and lon. default 25.0, max 100.0
- **remove_topics** – Comma-delimited list of topic ids to remove from your alert list

- **sync_photo** – When set to true, this parameter will sync all of the group profile photos for the member with the provided `photo_id`
- **topics_privacy** – Controls the visibility of the member’s topics. May be one of visible, hidden
- **zip** – Valid zip code for city

EditProfile (*gid, mid, add_role, answer_{qid}, intro, new_photo, photo_id, remove_role, site_name, site_url, title*)

Update a member’s group profile The *gid* and *mid* in this method’s path are group and member IDs, both needed to uniquely identify the member profile to be edited. An authenticated user may update their own profile by substituting “self” for the *mid*. All POST parameters are optional unless required by the group. Only the organizer of the group may edit the member’s *title* and *role* within the group. Those are the only things the organizer may edit. When requesting to set a role with *add_role* the member’s previous role will be cleared. Members may edit all other fields of their own profiles with the exception of *title*, *add_role*, and *remove_role*. This method requires an HTTP POST.

URI: /2/profile/{gid}/{mid}

API Version: 2

Parameters

- **gid** (*required*) – None
- **mid** (*required*) – None
- **add_role** – Allows those with permission to assign one of the following roles: coorganizer, event_organizer, assistant_organizer
- **answer_{qid}** – Answers to questions from groups API join_info question fields
- **intro** – Provides a Member an opportunity to tell the group about themselves
- **new_photo** – file upload for a new member photo
- **photo_id** – photo_id of the photo to use for this profile. set to 0 to unset the current photo
- **remove_role** – Allows those with permission to remove one of the following roles: coorganizer, event_organizer, assistant_organizer
- **site_name** – Name of member’s site. Max length is 32
- **site_url** – Link to member’s site. Max length is 80
- **title** – An organizer-defined member title.

GetActivity (*member_id, page_start*)

API method for retrieving the activity feed for a member’s groups

URI: /activity

API Version: 1

Parameters

- **member_id** – Returns activity from this member’s groups. Must be authenticated as this member
- **page_start** – Starting timestamp for item to return.

GetCategories (*fields, member, shortname*)

Returns a list of Meetup group categories No parameters required parameters

URI: /2/categories

API Version: 2

Parameters

- **fields** – Parameter for requesting optional response properties
- **member** – order by recommendations for authorized member (deprecated)
- **shortname** – (default order) ascending

GetCities (*country, distance, lat, lon, query, radius, size, smart, state*)

Returns Meetup cities. This method supports search by latitude/longitude/radius, by country/state, by query term/zip, or a combination of all of these. Location-only searches by lat and lon return all cities within a radius of the provided coordinates. Searches with a query return up to 10 cities matching the term, and can be sorted by size or distance to a given coordinate. ‘smart’ ordering can be used to return the match(es) with the highest member_count, unless a smaller size match exists nearby the given coordinates. Query searches are supported for country but not country and state This method requires no authentication but is subject to [request limiting](/meetup_api/docs/#limits) based on client IP . This search is location based. A lat and lon must be supplied together or a query or the search will be based on your geo-located ip

URI: /2/cities

API Version: 2

Parameters

- **country** – A valid country code
- **distance** – When lat/lon/query provided, sort matches by distance to coordinates
- **lat** – Latitude to search
- **lon** – Longitude to search
- **query** – Search term and/or zip to look for (if this is specified, max result size limited to 10)
- **radius** – When searching by lat/lon only, specify a radius to search (default 50 miles)
- **size** – (default order) number of members in the city: descending
- **smart** – Sort by size, but if a match is close to given lat/lon, float to top. Only works when query provided
- **state** – A valid state code for the given country, if the country has states

GetComments (*ctime, group_id, group_urlname, groupnum, topic*)

API method for accessing meetup group comments At least one of the required parameter(s) must be supplied with the request.

URI: /comments

API Version: 1

Parameters

- **ctime** – the date the comment was posted
- **group_id** – Return comments in groups with these ID numbers [separated by commas]
- **group_urlname** – Return comments for the group with this custom URL path

- **groupnum** – Return comments for the group with given topic and number
- **topic** – Return comments for the group with given topic and number

GetConcierge (*category_id, city, country, fields, lat, lon, page_token, radius, self_groups, state, text_format, time, topic_id, with_friends, zip*)

Recommends upcoming meetups for the authorized member in a given location and in thier groups For geo-based requests, you may provide a location in one of three ways. By *lat* and *lon*, by *zip*, or by *country, city*, and optionally a *state*, if the provided country has states.

If the server is unable to produce recommendations in a suitable amount of time, a 503 error will be returned. If no parameters are specified, Meetups are recommended for the upcoming week in the member’s default location.

Pagination works a little differently in this method than in others. Rather than using “offset” and “page” request parameters, this method uses an opaque “page_token” request parameter to determine the page of results returned. If there are more results, the “next” property of the [meta section]/(meetup_api/docs/#meta) of the response will contain the next page’s page_token. The number of results returned is not deterministic but a best-effort attempt will be made to return at least some.

URI: /2/concierge

API Version: 2

Parameters

- **category_id** – Comma delimited list of category ids to limit recommendations to
- **city** – A valid city
- **country** – A valid country code
- **fields** – Request that additional fields (separated by commas) be included in the output
- **lat** – A valid latitude, limits the returned group events to those within radius miles
- **lon** – A valid longitude, limits the returned group events to those within radius miles
- **page_token** – An opaque string used to page through results. This can be found appended to the ‘next’ link in the meta section of the response.
- **radius** – Radius, in miles for geographic requests, defaults to the member’s preferred radius or 0.5 – maximum 100. May also be specified as “smart”, a dynamic radius based on the number of active groups in the area
- **self_groups** – set to “include” or “exclude” groups the authorized member belongs to. The default is “include”. This includes groups in locations that may differ than the provided location
- **state** – If searching in a country with states, a valid 2 character state code
- **text_format** – Format of the description text, “html”, “plain”, or “simplehtml”. Defaults to “html”
- **time** – ascending time is the default and only ordering
- **topic_id** – Comma delimited list of topics to help inform recommendation
- **with_friends** – Boolean parameter. When set to true, events hosted by groups you have friends in will be recommended
- **zip** – A valid US zip code, limits the returned groups to those within radius miles

GetDashboard (*fields*)

A dashboard of aggregated Meetup information for the authorized member no parameters are required

URI: /dashboard

API Version: 3

Parameters **fields** – Request that additional fields (separated by commas) be included in the output

GetEvent (*id, fields, limited_events, text_format*)

Retrieve a single meetup Only authorization parameters are needed. Some groups, while remaining private, still wish to show some information about their events. You can include these events in results using the *limited_events* request parameter.

URI: /2/event/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **fields** – Request that additional fields (separated by commas) be included in the output.
- **limited_events** – Include limited event information for private groups that wish to expose only a small amount of information about their events. This includes just: id, name, utc_offset, time, duration, yes_rsvp_count, waitlist_count, group, visibility, timezone. Value must be true or false.
- **text_format** – Format of the description in the response, “html” or “plain”. Defaults to “html”

GetEventComment (*id, fields*)

Retrieve a single event comment or reply Only authorization parameters are needed.

URI: /2/event_comment/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **fields** – comma-separate list of optional fields

GetEventCommentLikes (*comment_id, created*)

Api for listing likes of a given event comment At least one of the required parameters must be supplied with the request.

URI: /2/event_comment_likes

API Version: 2

Parameters

- **comment_id** – Return likes for a given comment_id
- **created** – Order by the time the member liked like comment (default: descending)

GetEventComments (*comment_id, event_id, fields, group_id, member_id, name, thread, time*)

This method returns messages that appear under “Talk about this Meetup”. To post messages, see the corresponding write method. At least one of the required parameters must be supplied with the request.

URI: /2/event_comments

API Version: 2

Parameters

- **comment_id** – Return comments for a given set of comment IDs, separated by commas
- **event_id** – Return comments on these events, separated by commas.
- **fields** – Optionally accepts the value “member_photo” or “notifications”
- **group_id** – Return comments in groups with these ID numbers, separated by commas
- **member_id** – Return comments for the given member_ids, separated by commas
- **name** – Order by the name of the member
- **thread** – Order by comment threads, those with the most recent activity are listed first. Only one event may be provided and desc not supported. Also note that with this ordering, *replies* are excluded from pagination accounting. Only top-level comments will be capped at the page size.
- **time** – Order by the time that each rating was posted (default: descending)

GetEventRatings (*event_id, member_id, rating, time*)

API method for accessing Meetup comments At least one of the required parameters must be supplied with the request.

URI: /2/event_ratings

API Version: 2

Parameters

- **event_id** – The ID of the event to fetch ratings data for
- **member_id** – The ID of a member to filter ratings on
- **rating** – Order by the value for each rating posted
- **time** – Order by the time that each rating was posted

GetEvents (*event_id, fields, group_domain, group_id, group_urlname, limited_events, member_id, rsvp, status, text_format, time, venue_id*)

Access Meetup events using a group, member, or event id. Events in private groups are available only to authenticated members of those groups. To search events by topic or location, see [Open Events](/meetup_api/docs/2/open_events). At least one of the required parameter(s) must be supplied with the request. Some groups, while remaining private, still wish to show some information about their events. You can include these events in results using the *limited_events* request parameter.

URI: /2/events

API Version: 2

Parameters

- **event_id** – Multiple ids may be separated with commas
- **fields** – Request that additional fields (separated by commas) be included in the output
- **group_domain** – Group custom domain
- **group_id** – Multiple ids may be separated with commas
- **group_urlname** – Path to group from meetup.com, no slashes

- **limited_events** – Include limited event information for private groups that wish to expose only a small amount of information about their events. This includes just: id, name, utc_offset, time, duration, yes_rsvp_count, waitlist_count, group, visibility, timezone. Value must be true or false.
- **member_id** – Single member id, to find events in this member’s groups
- **rsvp** – Filters events by the currently authenticated member’s RSVP status. May be a comma delimited list of “yes”, “no”, “waitlist”, “maybe” or “none”
- **status** – Status may be “upcoming”, “past”, “proposed”, “suggested”, “cancelled”, “draft” or multiple separated by a comma. The default is “upcoming”, which includes Meetups that are happening now according to their **duration**. Meetups that are “proposed” or “suggested” do not have a date assigned; the former are listed on the site as *official* while the latter appear as *in the making*. Drafts are only visible to organizers of groups hosting the events.
- **text_format** – Format of the description text, “html” or “plain”. Defaults to “html”
- **time** – (default order) ascending
- **venue_id** – Multiple ids may be separated with commas

GetFindGroups (*category, country, distance, fallback_suggestions, fields, filter, lat, location, lon, members, most_active, newest, radius, self_groups, text, upcoming_events, zip*)

Text, location, category and friend-based group searches All parameters are optional. If you do not supply some explicit form of location, the results will be based on your registered Meetup profile location, falling back on your IP’s geographic location.

If we can’t find Meetup groups matching your criteria you may optionally request that a small set of suggestions are returned in their place. To do so, send a request parameter named *fallback_suggestions* set to *true*. You can infer that a list of Meetup groups return are fallback suggestions when the *X-Total-Count* pagination response header has a 0 value.

See the ‘filter’ parameter for more information controlling the results returned. Ordering does not apply to friend-filtered queries.

URI: /find/groups

API Version: 3

Parameters

- **category** – Comma-delimited list of numeric category ids
- **country** – A valid two character country code, defaults to US
- **distance** – Order by distance
- **fallback_suggestions** – boolean indicator of whether or not to return a list of curated suggestions for groups if we can’t find groups matching your criteria
- **fields** – Request that additional fields (separated by commas) be included in the output.
- **filter** – Determines which groups are returned. If ‘all’ (default), the text and category parameters are applied. If ‘friends’, groups your friends are in are returned. The value of this parameter may be one of all, friends
- **lat** – Approximate latitude
- **location** – Raw text location query
- **lon** – Approximate longitude
- **members** – Order by number of members

- **most_active** – Order by group with most active members
- **newest** – Order by date group was founded
- **radius** – Radius in miles. May be 0.0-100.0, ‘global’ or ‘smart’, a dynamic radius based on the number of active groups in the area
- **self_groups** – set to ‘include’ or ‘exclude’ meetups the authorized member belongs to; default is ‘include’
- **text** – Raw full text search query
- **upcoming_events** – If true, filters text and category based searches on groups that have upcoming events. Defaults to false
- **zip** – Zipcode of location to limit search to

GetGroup (*urlname, fields*)

Fetches a Meetup Group by urlname Only the path param :urlname is required

URI: /{urlname}

API Version: 3

Parameters

- **urlname** (*required*) – None
- **fields** – A comma-delimited list of optional fields to append to the response

GetGroupBoards (*urlname*)

Listings of Group discussion boards The :urlname path element may be any valid group urlname or domain name

URI: /{urlname}/boards

API Version: 3

Parameters **urlname** (*required*) – None

GetGroupBoardsDiscussions (*bid, did, urlname*)

Listing Group discussion posts The :urlname path element may be any valid group urlname or domain name. The :bid path element maybe any valid board ID for this group. The :did may be any valid discussion ID for this board

URI: /{urlname}/boards/{bid}/discussions/{did}

API Version: 3

Parameters

- **bid** (*required*) – None
- **did** (*required*) – None
- **urlname** (*required*) – None

GetGroupEventsAttendance (*id, urlname, filter, member*)

Lists attendance records for Meetup events. Limited for use by administrative members. The :urlname path element may be any valid group urlname or domain name. The :id path element must be a valid alphanumeric Meetup event identifier

URI: /{urlname}/events/{id}/attendance

API Version: 3

Parameters

- **id** (*required*) – None
- **urlname** (*required*) – None
- **filter** – A named filter to apply to the attendance list. These are roughly equivalent to the set of filters you will see in the attendance tool on the site. These filters correspond with attendance records as well as each member’s original RSVP status. The filter value be one of: maybe, waitlist, yes, absent, all, attended, noshow, excused, no. The default is ‘attended’. The ‘absent’ filter represents all members not in attendance including members with a ‘noshow’ status. An ‘excused’ absence is an absent member marked as such by an administrative member
- **member** – Raw text used to search for member by name. This may only be applied when the filter parameter is set to ‘all’. The provided text must consist of at least 2 characters.

GetGroupSimilarGroups (*urlname*)

Renders a list of similar groups The *:urlname* path element may be any valid group urlname or domain name.

URI: /{urlname}/similar_groups

API Version: 3

Parameters **urlname** (*required*) – None

GetGroupVenues (*urlname*)

Returns venues a group has previously hosted events at The *:urlname* path element may be any valid group urlname or domain name. All parameters are optional

URI: /{urlname}/venues

API Version: 3

Parameters **urlname** (*required*) – None

GetGroups (*category_id, city, country, domain, fields, group_id, group_urlname, groupnum, id, lat, location, lon, member_id, members, name, organizer_id, radius, state, topic, zip*)

Fetch information about Meetup Groups. At least one of the required parameter(s) must be supplied with the request. Answers to questions are expected to follow the naming convention: answer_{question_id}=myanswer when joining with the [Profile Create](/meetup_api/docs/2/profile) method.

URI: /2/groups

API Version: 2

Parameters

- **category_id** – Only return groups in the specified category. [one category allowed]
- **city** – A valid country code, city and for the US, State. limits the returned groups to those within radius miles
- **country** – A valid country code, city and for the US, State. limits the returned groups to those within radius miles
- **domain** – one or more custom group domains, separated by commas
- **fields** – optional result fields, separated by commas.
- **group_id** – one or more separated by commas
- **group_urlname** – one or more separated by commas, includes no slashes
- **groupnum** – Return the group with this topic and number

- **id** – group creation time [newest first]
- **lat** – A valid latitude and longitude, limits the returned groups to those within radius miles
- **location** – group location, country, state [if present], city
- **lon** – A valid latitude and longitude, limits the returned groups to those within radius miles
- **member_id** – one or more separated by commas, for groups this member belongs to
- **members** – number of members [largest first]
- **name** – the name of the group
- **organizer_id** – one or more organizer IDs, separated by commas
- **radius** – Radius, in miles for geographic requests, default 25 – maximum 100
- **state** – A valid country code, city and for the US, State. limits the returned groups to those within radius miles
- **topic** – Only return groups in the specified topic [one topic allowed]
- **zip** – A valid US zip code, limits the returned groups to those within radius miles.

GetMember (*id, fields*)

Retrieve a single member Only authorization parameters are needed.

URI: /2/member/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **fields** – comma-separate list of optional fields

GetMembers (*fields, group_id, group_urlname, groupnum, joined, member_id, name, service, topic, visited*)

API method for accessing members of Meetup Groups At least one of the required parameter(s) must be supplied with the request. If any of the groups you specify are private, you will not see its members listed in the result set unless you are a member of that group.

URI: /2/members

API Version: 2

Parameters

- **fields** – Request that additional fields (separated by commas) be included in the output.
- **group_id** – Return members in groups with these ID numbers, separated by commas
- **group_urlname** – Return members for the group with the given custom URL path
- **groupnum** – Return members for the group with given topic and number
- **joined** – time member joined Meetup
- **member_id** – Return the member with this ID
- **name** – the name of the member (default order)
- **service** – Match users by the external services they’ve linked to their member account, specified as “servicename:identifier”. For example, “service=twitter:@MeetupAPI” finds any member account that lists @MeetupAPI as its Twitter name (none, currently). You

can query against several at a time by separating them with commas. Facebook identifiers should be provided as numeric values

- **topic** – Return members for the group with given topic and number
- **visited** – member’s most recent activity

GetNotifications (*fields*)

Returns all recent Meetup notifications for the authorized member. To mark notifications read use [/notifications/read](/meetup_api/docs/notifications/read/) endpoint. To get the authenticated Member’s current unread count, request it in an [HTTP header](/meetup_api/docs/#meta-headers). No parameters are required

URI: /notifications

API Version: 3

Parameters fields – Request that additional fields (separated by commas) be included in the output.

GetOembed (*maxwidth, url*)

oEmbed implementation

This method does not require authentication. It requires only a url parameter and responds according to the [OEmbed specification](http://www.oembed.com/). Any of the following base URLs may refer to embeddable content:

- <http://www.meetup.com/>
- <http://meetup.com/>
- <http://meetu.ps/>

An optional `__maxwidth__` parameter may be provided.

URI: /oembed

API Version: 1

Parameters

- **maxwidth** – maximum width to display
- **url** – url of resource to be embedded

GetOpenEvents (*and_text, category, city, country, distance, fields, lat, limited_events, lon, radius, state, status, text, text_format, time, topic, trending, zip*)

Searches for recent and upcoming public events hosted by Meetup groups. Its search window is the past one month through the next three months, and is subject to change. Open Events is optimized to search for current events by location, category, topic, or text, and only lists Meetups that have **3 or more RSVPs**. The number of results returned with each request is not guaranteed to be the same as the page size due to secondary filtering. If you’re looking for a particular event or events within a particular group, use the standard [Events](/meetup_api/docs/2/events/) method. At least one of the required parameter(s) must be supplied with the request.

For geo-based requests, you may provide a location in one of three ways. By *lat* and *lon*, by *zip*, or by *country*, *city*, and optionally a *state*, if the provided country has states.

Some groups, while remaining private, still wish to show some information about their events. You can include these events in results using the *limited_events* request parameter.

URI: /2/open_events

API Version: 2

Parameters

- **and_text** – Changes the interpretation of the “text” field from OR’d terms to AND’d terms
- **category** – Return events in the specified category or categories specified by commas. This is the category id returned by the Categories method.
- **city** – A valid city
- **country** – A valid country code
- **distance** – ordering is approximate and will not exactly match the values in the “distance” field.
- **fields** – Request that additional fields (separated by commas) be included in the output
- **lat** – A valid latitude, limits the returned group events to those within radius miles
- **limited_events** – Include limited event information for private groups that wish to expose only a small amount of information about their events. This includes just: id, name, utc_offset, time, duration, yes_rsvp_count, waitlist_count, group, visibility, timezone. Value must be true or false.
- **lon** – A valid longitude, limits the returned group events to those within radius miles
- **radius** – Radius, in miles for geographic requests, default 25.0 – maximum 100. May also be specified as “smart”, a dynamic radius based on the number of active groups in the area
- **state** – If searching in a country with states, a valid 2 character state code
- **status** – Status may be “upcoming”, “past” or both separated by a comma. The default is “upcoming” only
- **text** – Events that contain the given term or terms somewhere in their content. The terms are OR’d by default. Separate terms with ” AND ” for events that have combined terms. To have terms automatically AND’d, set the “and_text” to true
- **text_format** – Format of the description text, “html” or “plain”. Defaults to “html”
- **time** – (default order) ascending
- **topic** – Return events in the specified topic or topics specified by commas. This is the topic “urlkey” returned by the Topics method. If all supplied topics are unknown, a 400 error response is returned with the code “badtopic”.
- **trending** – you will likely want to specify “desc=true” to get the best trending results first.
- **zip** – A valid US zip code, limits the returned groups to those within radius miles

GetOpenVenues (*city, country, distance, fields, group_urlname, lat, lon, radius, rating, rating_count, state, text, zip*)

Searches for public venues within a given geo space. To search for specific venues that your group has used, use the [Venues](/meetup_api/docs/2/venues) method At least one of the required parameter(s) must be supplied with the request. If you are not able to provide a relative location you may supply a group_urlname which this method will use to infer location from. You can perform prefix searching by appending a * character to your query

URI: /2/open_venues

API Version: 2

Parameters

- **city** – A valid city
- **country** – A valid country code.
- **distance** – (default order) ordering is approximate and will not exactly match the values in the “distance” field.
- **fields** – Request that additional fields (separated by commas) be included in the output
- **group_urlname** – Returns venues with location relative to the group associated with this urlname
- **lat** – A valid latitude, limits the returned venues to those within radius miles
- **lon** – A valid longitude, limits the returned venues to those within radius miles
- **radius** – Radius, in miles for geographic requests, default 25.0 – maximum 100.0
- **rating** – average member rating
- **rating_count** – number of member ratings.
- **state** – For the US, a valid 2 character state code
- **text** – Venues that contain the given term or terms somewhere in their content. Separate terms with ” AND ” for venues that have combined terms. Append a trailing * to treat this as a prefix search
- **zip** – A valid US zip code, limits the returned venues to those within radius miles

GetPhotoAlbums (*event_id, group_id, photo_album_id, time, title, updated*)

This method returns photo albums associated with Meetup groups. To create albums, see the corresponding write method. At least one of the required parameter(s) must be supplied with the request.

URI: /2/photo_albums

API Version: 2

Parameters

- **event_id** – Return photo albums for these event ids, separated by commas
- **group_id** – Return albums in groups with these ID, separated by commas
- **photo_album_id** – Return albums with these IDs, separated by commas
- **time** – creation date and time
- **title** – title of the album
- **updated** – album updated field

GetPhotoComments (*fields, member_id, name, photo_id, time*)

This method returns comments on meetup photos. To post messages, see the corresponding write method At least one of the required parameters must be supplied by the request.

URI: /2/photo_comments

API Version: 2

Parameters

- **fields** – Request that additional fields (separated by commas) be included in the output.
- **member_id** – Return comments for the given member_ids, separated by commas. The member ids must match up with one of the provided photo ids

- **name** – the name of the member
- **photo_id** – Return comments on these photos, separated by commas
- **time** – Order by the time that each comment was posted

GetPhotos (*event_id, fields, group_id, group_urlname, member_id, photo_album_id, photo_id, tagged, time*)

This method returns photos by member, group, album, event, photo ID, or tagged member. At least one of the required parameter(s) must be supplied with the request.

URI: /2/photos

API Version: 2

Parameters

- **event_id** – Event ids, separated by commas. These may contain alphanumeric autoscheduled event ids, only photos of reified events will be returned
- **fields** – comma-delimited optional response properties such as member_country, member_city, member_state, and self
- **group_id** – Group IDs, separated by commas
- **group_urlname** – Group urlnames, separated by commas
- **member_id** – Uploaded by members with these IDs, separated by commas
- **photo_album_id** – Photo Album IDs, separated by commas
- **photo_id** – Photo IDs, separated by commas
- **tagged** – Tagged with members with these IDs, separated by commas
- **time** – creation date and time

GetProfile (*gid, mid, fields*)

Retrieves a single group profile Only authorization parameters are needed.

URI: /2/profile/{gid}/{mid}

API Version: 2

Parameters

- **gid** (*required*) – None
- **mid** (*required*) – None
- **fields** – Request that additional fields (separated by commas) be included in the output

GetProfiles (*fields, group_id, group_urlname, groupnum, interesting, joined, member_id, name, role, status, topic, updated, visited*)

This method returns member *profiles* associated with a particular group. Meetup members have separate profiles for each group they join. At least one of the required parameter(s) must be supplied with the request. If any of the groups you specify are private, you will not see its members listed in the result set unless you are a member of that group. If querying for member_id alone, the member's 'hide groups' preference will be honored. 'Interesting' ordered searches only support searching by group_urlname or group_id. Only one of those may be supplied.

URI: /2/profiles

API Version: 2

Parameters

- **fields** – comma delimited list of optional response properties. A value of “membership_dues” will populate membership dues for the authorized user or members of the groups the authorized user organizes
- **group_id** – Return profiles in the group with this ID
- **group_urlname** – Return profiles for the group with the given custom URL path
- **groupnum** – Group identification by topic, deprecated
- **interesting** – Order which may be interesting to the authorized member
- **joined** – time member joined this group
- **member_id** – the id of the member
- **name** – the name of the member
- **role** – if “leads”, only profiles for members of the leadership team are included
- **status** – Status filter for members. Only organizers may see pending. Request must also contain a *group_id* or *group_urlname*. Status may be one of active, pending
- **topic** – Group identification by topic, deprecated
- **updated** – profile updated field
- **visited** – last visit to group pages (default order)

GetRecommendedGroupTopics (*exclude_topics, lang, other_topics, text*)

Recommends suggestions for group topics based on a text search or other topics

URI: /recommended/group_topics

API Version: 3

Parameters

- **exclude_topics** – A comma-delimited list of topic ids to exclude from the recommendations
- **lang** – Defines a language preference for ordering results. Valid values are frptitdeesen_us. You may also substitute this with the Accept-Language header
- **other_topics** – A comma-delimited list of topic ids to inform recommendations
- **text** – Free form text search

GetRecommendedGroups (*category, country, fields, instant_join_only, lat, location, lon, radius, topic_id, zip*)

Returns groups Meetup finds relevant to you All parameters are optional. You may change the ‘location’ and ‘radius’ for the request. If you do not supply a location your request will be based on your IP’s geographic location. If the server is unable to produce recommendations in a suitable amount of time, a 503 error will be returned.

URI: /recommended/groups

API Version: 3

Parameters

- **category** – A valid category id which limits recommended groups to a particular category
- **country** – A valid two character country code, defaults to US
- **fields** – Request that additional fields (separated by commas) be included in the output.

- **instant_join_only** – Recommend only groups without join requirements and that can be joined instantly
- **lat** – Approximate latitude
- **location** – Raw text location query
- **lon** – Approximate longitude
- **radius** – Radius in miles. May be 0.0-100.0, ‘global’ or ‘smart’, a dynamic radius based on the number of active groups in the area
- **topic_id** – Comma delimited list of up to 100 topic ids to help inform recommendations
- **zip** – Zip code you are searching for recommendations in

GetRecommendedVenues (*category, country, group_id, group_urlname, lat, location, lon, min_groups, radius, used_between, zip*)

Returns venues Meetup finds relevant to you based on location and category. This method does not yet support sorting or pagination. All parameters are optional. If you do not supply a location your request will be based on your IP’s geographic location. If the server is unable to produce recommendations in a suitable amount of time, a 503 error will be returned.

URI: /recommended/venues

API Version: 3

Parameters

- **category** – Comma-delimited list of up to 200 category ids to help inform recommendations
- **country** – A valid two character country code, defaults to US
- **group_id** – Comma-delimited list of up to 200 group ids to help inform recommendations
- **group_urlname** – Comma-delimited list of up to 200 group urlnames to help inform recommendations
- **lat** – Approximate latitude
- **location** – Raw text location query
- **lon** – Approximate longitude
- **min_groups** – The minimum number of groups that have hosted events at this venue
- **radius** – Radius in miles
- **used_between** – Return venues that have been used within the given time range, defined by two times separated with a single comma. Each end of the range may be specified with relative dates, such as “1m” for one month from now, or by absolute time in milliseconds since the epoch. If an endpoint is omitted, the range is unbounded on that end. The default value is unbounded on both ends (though restricted to the search window described above).
- **zip** – Zip code you are searching for recommendations in

GetRsvp (*id, fields*)

Retrieve a single RSVP Only authorization parameters are needed.

URI: /2/rsvp/{id}

API Version: 2

Parameters

- **id** (*required*) – None
- **fields** – Request that additional fields (separated by commas) be included in the output

GetRsvps (*event, event_id, fields, name, rsvp, social*)

Query for Event RSVPs by event At least one of the required parameter(s) must be supplied with the request.

URI: /2/rsvps

API Version: 2

Parameters

- **event** – the id or time of the event
- **event_id** – Multiple alphanumeric ids may be separated with commas
- **fields** – Parameter for requesting optional response properties, set to other_services for a list of third party services
- **name** – the name of the attendee
- **rsvp** – Filters response on RSVP status. “yes” if member RSVP’d yes otherwise “no”
- **social** – social connections of the authenticated member

GetSelfBlocks (*member_id*)

Checks the block status for a target member relative to the authenticated member A valid member_id path parameter for the target member is required.

URI: /self/blocks/{member_id}

API Version: 3

Parameters **member_id** (*required*) – None

GetStatus ()

Returns the current API service status No parameters are needed

URI: /status

API Version: 3

GetTopicCategories (*fields, lat, lon, radius, shortname*)

Returns a list of Meetup topic categories No parameters required parameters

URI: /2/topic_categories

API Version: 2

Parameters

- **fields** – Parameter for requesting optional response properties
- **lat** – Use a given lat/lon/radius (miles) to search best_topics for instead of using the member’s lat/lon. When present, all three are required.
- **lon** – Use a given lat/lon/radius (miles) to search best_topics for instead of using the member’s lat/lon. When present, all three are required.
- **radius** – Use a given lat/lon/radius (miles) to search best_topics for instead of using the member’s lat/lon. When present, all three are required.
- **shortname** – (default order) ascending

GetTopics (*alrtees, member_id, members, name, search, topic*)

API method for accessing meetup topics At least one of the required parameter(s) must be supplied with the request.

URI: /topics

API Version: 1

Parameters

- **alertees** – total alertees in the topic
- **member_id** – Return topics a target member is subscribed to
- **members** – total members in the topic
- **name** – the topic name
- **search** – Return topics related to a list of search terms [separate search keywords with +’s]
- **topic** – the topic key

GetVenues (*distance, event_id, fields, group_id, group_urlname, rating, rating_count, venue_id*)

Search for Meetup venues by one of your groups, events, or venue identifiers. For a full text search on public venues use [OpenVenues](/meetup_api/docs/2/open_venues). At least one of the required parameter(s) must be supplied with the request. You can use either `group_urlname` or `group_id` to filter by groups but not both.

URI: /2/venues

API Version: 2

Parameters

- **distance** – (default order) ordering is approximate and will not exactly match the values in the “distance” field.
- **event_id** – multiple ids may be separated with commas
- **fields** – Request that additional fields (separated by commas) be included in the output
- **group_id** – multiple ids may be separated with commas
- **group_urlname** – path to group from meetup.com, no slashes
- **rating** – average member rating
- **rating_count** – number of member ratings.
- **venue_id** – multiple ids may be separated with commas

Meetup Exceptions Module

Exception Classes

exception `meetup.exceptions.ApiKeyError`

There is a problem with the client API key.

exception `meetup.exceptions.ApiMethodError`

The called API method is not defined or does not exist.

exception `meetup.exceptions.ApiParameterError`

The called API method is missing a required parameter.

exception `meetup.exceptions.ClientException`

Meetup Client Exception base class.

exception `meetup.exceptions.HttpClientError`

Called when the server tells us there was a client error (4xx).

exception `meetup.exceptions.HttpMethodError`

The requested HTTP Method is not valid.

exception `meetup.exceptions.HttpNotAccessibleError`

Called when the server sends a 410 error.

exception `meetup.exceptions.HttpNotFoundError`

Called when the server sends a 404 error.

exception `meetup.exceptions.HttpServerError`

Called when the server tells us there was a server error (5xx).

exception `meetup.exceptions.HttpTooManyRequests`

Called when the server sends a 429 error (when you've gone over your request rate limit)

exception `meetup.exceptions.HttpUnauthorized`

Called when the server sends a 401 error (when you don't provide a valid key)

exception `meetup.exceptions.MeetupBaseException`

All Meetup exceptions inherit from this exception.

exception `meetup.exceptions.MeetupHttpBaseException`

All Meetup HTTP Exceptions inherit from this exception.

Before contributing to Meetup API, make sure your patch passes the test suite and your code style passes the code linting suite.

Meetup API uses `Tox` to execute testing and linting procedures. `Tox` is the only dependency you need to run linting or the test suite, the remainder of the requirements will be installed by `Tox` into environment specific virtualenv paths. Before testing, make sure you have `Tox` installed:

```
pip install tox
```

To run the full test and lint suite against your changes, simply run `Tox`. `Tox` should return without any errors. You can run `Tox` against all of the environments by running:

```
tox
```

To target a specific environment:

```
tox -e py27
```

The `tox` configuration has the following environments configured. You can target a single environment to limit the test suite:

Env Name	Description
py26	Run the test suite using Python 2.6
py27	Run the test suite using Python 2.7
py34	Run the test suite using Python 3.4
py35	Run the test suite using Python 3.5
coverage	Run the test suite and check code coverage
flake8	Run code linting using <code>flake8</code> . This currently runs <code>pyflakes</code> , <code>pep8</code> , and other linting tools.
docs	Test documentation compilation with Sphinx.

Continuous Integration

The Meetup API test suite is exercised by Travis CI on every push to the repo at GitHub. You can check out the current build status: <https://travis-ci.org/pferate/meetup-api>

CHAPTER 4

Changelog

v0.1.1 2016-03-03

- Adding MeetupObjectList, for handling multiple MeetupObjects

v0.1 2016-02-04

- Initial release

The MIT License (MIT)

Copyright (c) 2016 Pat Ferate

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 6

Indices and tables

- genindex
- modindex

m

`meetup.api`, 5

`meetup.exceptions`, 39

A

ApiKeyError, 39
ApiMethodError, 39
ApiParameterError, 39

C

Client (class in meetup.api), 5
ClientException, 39
CreateEvent() (meetup.api.meetup.api.Client method), 8
CreateEventComment() (meetup.api.meetup.api.Client method), 9
CreateEventCommentFlag() (meetup.api.meetup.api.Client method), 9
CreateEventRating() (meetup.api.meetup.api.Client method), 10
CreateGroupAbuseReports() (meetup.api.meetup.api.Client method), 10
CreateGroupMemberApprovals() (meetup.api.meetup.api.Client method), 10
CreateGroupPhoto() (meetup.api.meetup.api.Client method), 10
CreateGroupTopics() (meetup.api.meetup.api.Client method), 11
CreateGroupVenues() (meetup.api.meetup.api.Client method), 11
CreateMemberPhoto() (meetup.api.meetup.api.Client method), 12
CreateNotificationsRead() (meetup.api.meetup.api.Client method), 12
CreatePhoto() (meetup.api.meetup.api.Client method), 12
CreatePhotoAlbum() (meetup.api.meetup.api.Client method), 13
CreatePhotoComment() (meetup.api.meetup.api.Client method), 13
CreateProfile() (meetup.api.meetup.api.Client method), 13

CreateRecommendedGroupsIgnores() (meetup.api.meetup.api.Client method), 14
CreateRsvp() (meetup.api.meetup.api.Client method), 14
CreateSelfAbuseReports() (meetup.api.meetup.api.Client method), 15
CreateSelfBlocks() (meetup.api.meetup.api.Client method), 15

D

DeleteEvent() (meetup.api.meetup.api.Client method), 15
DeleteEventComment() (meetup.api.meetup.api.Client method), 15
DeleteEventCommentLike() (meetup.api.meetup.api.Client method), 16
DeleteEventCommentSubscribe() (meetup.api.meetup.api.Client method), 16
DeleteGroupEventsWatchlist() (meetup.api.meetup.api.Client method), 16
DeleteGroupMemberApprovals() (meetup.api.meetup.api.Client method), 16
DeleteGroupTopics() (meetup.api.meetup.api.Client method), 17
DeleteMemberPhoto() (meetup.api.meetup.api.Client method), 17
DeletePhoto() (meetup.api.meetup.api.Client method), 17
DeleteProfile() (meetup.api.meetup.api.Client method), 17
DeleteSelfBlocks() (meetup.api.meetup.api.Client method), 17

E

EditEvent() (meetup.api.meetup.api.Client method), 17
EditEventCommentLike() (meetup.api.meetup.api.Client method), 19

- EditEventCommentSubscribe() (meetup.api.meetup.api.Client method), 19
- EditGroup() (meetup.api.meetup.api.Client method), 19
- EditGroupEventsAttendance() (meetup.api.meetup.api.Client method), 21
- EditGroupEventsPayments() (meetup.api.meetup.api.Client method), 21
- EditGroupEventsWatchlist() (meetup.api.meetup.api.Client method), 21
- EditMember() (meetup.api.meetup.api.Client method), 22
- EditProfile() (meetup.api.meetup.api.Client method), 23
- ## G
- GetActivity() (meetup.api.meetup.api.Client method), 23
- GetCategories() (meetup.api.meetup.api.Client method), 23
- GetCities() (meetup.api.meetup.api.Client method), 24
- GetComments() (meetup.api.meetup.api.Client method), 24
- GetConcierge() (meetup.api.meetup.api.Client method), 25
- GetDashboard() (meetup.api.meetup.api.Client method), 25
- GetEvent() (meetup.api.meetup.api.Client method), 26
- GetEventComment() (meetup.api.meetup.api.Client method), 26
- GetEventCommentLikes() (meetup.api.meetup.api.Client method), 26
- GetEventComments() (meetup.api.meetup.api.Client method), 26
- GetEventRatings() (meetup.api.meetup.api.Client method), 27
- GetEvents() (meetup.api.meetup.api.Client method), 27
- GetFindGroups() (meetup.api.meetup.api.Client method), 28
- GetGroup() (meetup.api.meetup.api.Client method), 29
- GetGroupBoards() (meetup.api.meetup.api.Client method), 29
- GetGroupBoardsDiscussions() (meetup.api.meetup.api.Client method), 29
- GetGroupEventsAttendance() (meetup.api.meetup.api.Client method), 29
- GetGroups() (meetup.api.meetup.api.Client method), 30
- GetGroupSimilarGroups() (meetup.api.meetup.api.Client method), 30
- GetGroupVenues() (meetup.api.meetup.api.Client method), 30
- GetMember() (meetup.api.meetup.api.Client method), 31
- GetMembers() (meetup.api.meetup.api.Client method), 31
- GetNotifications() (meetup.api.meetup.api.Client method), 32
- GetOembed() (meetup.api.meetup.api.Client method), 32
- GetOpenEvents() (meetup.api.meetup.api.Client method), 32
- GetOpenVenues() (meetup.api.meetup.api.Client method), 33
- GetPhotoAlbums() (meetup.api.meetup.api.Client method), 34
- GetPhotoComments() (meetup.api.meetup.api.Client method), 34
- GetPhotos() (meetup.api.meetup.api.Client method), 35
- GetProfile() (meetup.api.meetup.api.Client method), 35
- GetProfiles() (meetup.api.meetup.api.Client method), 35
- GetRecommendedGroups() (meetup.api.meetup.api.Client method), 36
- GetRecommendedGroupTopics() (meetup.api.meetup.api.Client method), 36
- GetRecommendedVenues() (meetup.api.meetup.api.Client method), 37
- GetRsvp() (meetup.api.meetup.api.Client method), 37
- GetRsrvps() (meetup.api.meetup.api.Client method), 38
- GetSelfBlocks() (meetup.api.meetup.api.Client method), 38
- GetStatus() (meetup.api.meetup.api.Client method), 38
- GetTopicCategories() (meetup.api.meetup.api.Client method), 38
- GetTopics() (meetup.api.meetup.api.Client method), 38
- GetVenues() (meetup.api.meetup.api.Client method), 39
- ## H
- HttpClientError, 40
- HttpMethodError, 40
- HttpNotAccessibleError, 40
- HttpNotFoundError, 40
- HttpServerError, 40
- HttpTooManyRequests, 40
- HttpUnauthorized, 40
- ## L
- limit (meetup.api.RateLimit attribute), 6
- ## M
- meetup.api (module), 5
- meetup.api.Client (class in meetup.api), 8
- meetup.exceptions (module), 39
- MeetupBaseException, 40
- MeetupHttpBaseException, 40

MeetupObject (class in meetup.api), 6
MeetupObjectList (class in meetup.api), 6

R

RateLimit (class in meetup.api), 6
remaining (meetup.api.RateLimit attribute), 6
reset (meetup.api.RateLimit attribute), 6