

---

# libsaaas Documentation

*Release 0.1*

**Ducksboard**

Dec 29, 2017



---

## Contents

---

<b>1</b>	<b>Requirements</b>	<b>3</b>
1.1	Optional requirements . . . . .	3
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	Consulting original documentation . . . . .	6
2.2	Combining services . . . . .	6
2.3	Pluggable executors . . . . .	6
2.4	The saas script . . . . .	7
<b>3</b>	<b>Unicode support</b>	<b>9</b>
3.1	Internal usage . . . . .	9
3.2	Parsers . . . . .	10
<b>4</b>	<b>Supported services</b>	<b>11</b>
4.1	Basecamp . . . . .	11
4.2	BitBucket . . . . .	25
4.3	Bitly . . . . .	38
4.4	CartoDB . . . . .	44
4.5	Compete . . . . .	45
4.6	Desk . . . . .	45
4.7	Ducksboard . . . . .	56
4.8	Flurry . . . . .	59
4.9	Fullcontact . . . . .	63
4.10	GitHub . . . . .	65
4.11	GoogleAnalytics . . . . .	206
4.12	GoogleCalendar . . . . .	210
4.13	GoogleOAuth2 . . . . .	216
4.14	GoogleSpreadsheets . . . . .	216
4.15	Instagram . . . . .	219
4.16	Intercom . . . . .	224
4.17	Mailchimp . . . . .	227
4.18	Mixpanel . . . . .	235
4.19	MixRank . . . . .	238
4.20	Mozscape . . . . .	240
4.21	Insights . . . . .	241
4.22	Pingdom . . . . .	242
4.23	Pipedrive . . . . .	250

4.24	Recurly . . . . .	267
4.25	SegmentIO . . . . .	277
4.26	Spotify . . . . .	278
4.27	Stripe . . . . .	279
4.28	Trello . . . . .	286
4.29	Twilio . . . . .	312
4.30	UserVoice . . . . .	332
4.31	Analytics . . . . .	357
4.32	Zendesk . . . . .	358
<b>5</b>	<b>Indices and tables</b>	<b>367</b>

Libsaaas is a Python library that makes it easy to interact with various software as a service APIs. Think of it as an ORM for APIs.

Libsaaas handles all the nitty-gritty details, such as input and output format, authentication and URL schemes. You don't have to deal with serialization, extra headers and other quirks of various APIs. Everything's Python, you just pass Python objects to Python functions and get the results out:

```
>>> from libsaaas.services import zendesk
>>> service = zendesk.Zendesk('myapp', 'myuser', 's3cr3t')
>>> joe = service.users().search('joe@example.org')['users'][0]
>>> joes_tickets = service.user(joe['id']).tickets_requested()
>>> for ticket in joes_tickets:
...     print(ticket['description'])
```

Check out the list of currently supported services, add your own connectors and help us make using APIs less painful!



For basic operation libsaas does not depend on any external modules, it just uses the standard Python library. It is regularly tested on Python 2.6, 2.7, 3.2 and [PyPy](#).

### 1.1 Optional requirements

Some features of libsaas do need external libraries to be installed. To use one of the pluggable executors you will need the corresponding Python module, for instance [python-requests](#) for the Requests executor.





To use libsaas, you first create a service object and then use the method it provides to send and fetch data from a software as a service API.

With libsaas you always work with Python objects, serialization of data sent and received is provided by the library. Here's an example of using libsaas to check what's the most watched repository from all people that follow you.

```
from __future__ import print_function

from libsaas.services import github

# use basic authentication to create a token for libsaas
basic = github.GitHub('me@example.org', 'my-github-password')

auth = basic.authorizations().create({'scopes': 'repo,gist',
                                     'note': 'libsaas example'})

# use token authentication for the rest of the calls
gh = github.GitHub(auth['token'])

# go through your followers
for follower in gh.user().followers():
    username = follower['login']

    # get the source repositories owned by each follower
    repos = gh.user(username).repos().get(type='owner')
    sources = [repo for repo in repos if not repo['fork']]

    # handle the case where a user has no repos
    if not sources:
        print("{0} has no repositories".format(username))
        continue

    # print the most watched repo of each follower, excluding forks
    most = sorted(sources, key=lambda repo: repo['watchers'])[-1]
    print("{0}'s most watched repository: {1}".format(username, most['name']))
```

## 2.1 Consulting original documentation

The most productive way to use libsaas is to keep the original API documentation and the libsaas documentation open side-by-side. Since every API has its own data format, the abstraction provided by libsaas ends at providing you with Python objects. You should refer to the original service documentation in order to fully interpret the results.

## 2.2 Combining services

Libsaas is most useful when you combine access to different services. This allows you to quickly create mashups without worrying about all the little quirks of each API. Here's how you'd get the tickets solved yesterday in your Zendesk and accordingly tag users who reported those tickets in Mailchimp.

You could then run this script nightly and create an automatic mailing campaign to send quality surveys to users who's tickets have been solved recently.

```
from datetime import datetime, timedelta

from libsaas.services import mailchimp, zendesk

# create Zendesk and Mailchimp services
zd = zendesk.Zendesk('mycompany', 'username', 'password')
mc = mailchimp.Mailchimp('8ac789caf98879caf897a678fa76daf-us2')

# get tickets solved yesterday
yesterday = (datetime.now() - timedelta(days=1)).strftime('%Y-%m-%d')
tickets = zd.search('updated>{0} status:solved type:ticket'.format(yesterday))

# get emails of users who requested those tickets
user_ids = [ticket['requester_id'] for ticket in tickets['results']]
emails = [zd.user(user_id).get()['user']['email'] for user_id in user_ids]

# grab the ID of the "Users" list
lists = mc.lists(filters={'list_name': 'Users'})
list_id = lists['data'][0]['id']

# set the SOLVED variable for those users in Mailchimp to yesterday
batch = [{'EMAIL': email, 'SOLVED': yesterday} for email in emails]
mc.listBatchSubscribe(list_id, batch, double_optin=False, update_existing=True)
```

## 2.3 Pluggable executors

Everything out there uses HTTP, but there's more than one way to skin a request. By default libsaas uses Python's standard urllib2 to make HTTP requests, but it provides a few other executor modules and you can even plug in your own.

Here's an example of using the `Requests` executor in order to add a user agent and a timeout every time libsaas makes a HTTP request. The example code will unstar all gists a user has previously starred.

```
import libsaas
from libsaas.executors import requests_executor
from libsaas.services import github

# use the Requests executor with a custom timeout and make it always send a
```

```
# user agent string
uastring = 'libsaas {0}'.format(libsaas.__versionstr__)

requests_executor.use(timeout=5.0,
                      config={'base_headers': {'User-agent': uastring}})

# unstar all starred gists
gh = github.GitHub('my-github-token')

for gist in gh.gists().starred():
    gh.gist(gist['id']).unstar()
```

Another executor included in libsaas is the `Twisted` executor that makes it easy to integrate libsaas in Twisted programs. When using the Twisted executor, libsaas will return Deferreds that will fire with the fetched data.

## 2.4 The saas script

If you want to quickly interact with a SaaS API, you can use the `saas` command line program. It can execute any method provided by libsaas and has some limited discoverability features.

To use it, call it with the desired service as the first argument, followed by the service parameters and the path to the method you want to call.

This means that code like this:

```
>>> from libsaas.services import zendesk
>>> service = zendesk.Zendesk('myapp', 'myuser', 's3cr3t')
>>> service.user(364).tickets_requested()
```

Translates to:

```
$ saas zendesk --subdomain=myapp --username=myuser --password=s3cr3t \
  user 364 tickets_requested
```

The `saas` script can also choose executors and output some debugging information while executing:

```
$ saas mailchimp --api_key=8ac789caf98879caf897a678fa76daf-us2 \
  --executor requests campaignContent 8df731
```

Another useful feature is that `saas` will try to interpret every parameter as JSON, making it easier to use methods that need Python dictionaries:

```
$ saas github --token_or_password=my-gh-token \
  gist 125342 update '{"description": "my gist"}'
```



---

## Unicode support

---

Libsaas supports both 2.x and 3.x versions of Python, so some rules about where to use Unicode and where to use byte strings need to be followed.

As defined in the standards, URLs are ASCII-only. Characters from outside of ASCII should be encoded using so-called percent encoding. Encoding schemes, such as UTF-8 or ISO-8859-2 are mostly outside of the scope of HTTP. Libsaas tries to be flexible in what it accepts as user input and handles encoding and percent quoting automatically for you.

When you pass a string parameter to a libsaas method, it will be coerced to bytes before percent-encoding, and UTF-8 will be used as the encoding. This means that in Python 2 parameters of the *str* type will be used as-is and *unicode* parameters will be encoded according to UTF-8. In Python 3 *str* parameters will be UTF-8 encoded and *bytes* will be used as-is.

If you need to send characters outside of ASCII in different encoding than UTF-8, encode them yourself and hand the bytes off to libsaas.

Data returned from libsaas methods might be encoded differently, depending on the method in question. For most APIs using JSON it will be Unicode, because the JSON standard mandates the use of UTF-8, making it easy to convert bytes received from the service into Unicode characters. Some APIs though might return binary data, such as APIs exporting images or providing access to raw files. When in doubt, consult the upstream service documentation.

### 3.1 Internal usage

This section is only relevant if you are extending libsaas. Normal users that only interact with libsaas via the methods it provides can safely skip it.

Internally, libsaas uses a structure called `Request` to prepare and execute a HTTP request. There are four pieces of information this structure holds:

- the HTTP method to use
- the URL (without query parameters)
- the query parameters

- HTTP headers

Here's how each of them should be represented with regards to Unicode/bytes.

### 3.1.1 HTTP method

In both Python 2 and 3 this should be a *str* object, that only contains bytes from the ASCII range. Note that in Python 2 this represents a binary string and in Python 3 it means Unicode text. Since the HTTP method name can only include ASCII characters, the distinction is not important and using *str* makes it easy to write code that's compatible with both Python 2 and 3.

### 3.1.2 URL

Same as the HTTP method, it should be a string that only uses ASCII characters. Note that this means that libsaas methods should take care to encode user input adequately before handing the `Request` over to the executor and they need to be prepared to accept both byte strings and text.

### 3.1.3 Query parameters

This can either be a mapping of strings to strings or a simple string value. The executor will accept both byte strings and text for keys and values of the mapping, as well as numbers. It is the executor's responsibility to correctly encode and quote those value before making a HTTP request to the server.

### 3.1.4 HTTP headers

This should be a mapping of strings using ASCII characters to ASCII characters only (just like the HTTP method or the URL). According to RFC 2616 non-ASCII characters are allowed in headers, but should be mime-encoded (as defined in RFC 20147). If you need to use such values, encode them before handing the `Request` over to the executor.

## 3.2 Parsers

Data passed to parsers from the executor is always binary, which means that on Python 2 it will be a *str* and on Python 3 it will be *bytes*. It is the parser's responsibility to deal with any encoding issues.

## 4.1 Basecamp

**class** `basecamp.Basecamp` (*account\_id*, *token\_or\_username*, *password=None*)  
Create a Basecamp service.

### Variables

- **account\_id** (*int*) – Your Basecamp account id.
- **token\_or\_username** (*str*) – Either an OAuth 2.0 token, or the username if you want to use Basic authentication.
- **password** (*str*) – Only used with the Basic authentication, leave this as *None* when using OAuth.

### 4.1.1 GlobalAttachments

`Basecamp.attachments` ()  
Return the resource corresponding to all attachments.

`GlobalAttachments.create` (*obj*)  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`GlobalAttachments.delete` ()  
Delete this resource.

`GlobalAttachments.get` (*page=None*)  
Fetch all resources.

**Variables** **page** (*int*) – the page that will be return. If not indicated, first one is returned.

GlobalAttachments.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.1.2 Calendar

Basecamp.**calendar** (*calendar\_id*)

Return the resource corresponding to a single calendar.

## 4.1.3 Access

Calendar.**access** (*access\_id*)

Return the resource corresponding to a single access.

Access.**revoke** ()

Delete this resource.

## 4.1.4 Accesses

Calendar.**accesses** ()

Return the resource corresponding to all calendar accesses.

Accesses.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Accesses.**grant** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.1.5 CalendarEvent

Calendar.**calendar\_event** (*calendar\_event\_id*)

Return the resource corresponding to a single calendar event.

## 4.1.6 Comments

CalendarEvent.**comments** ()

Return the resource corresponding to all comments.

Comments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Comments.**delete** ()

Delete this resource.

Comments.**update** (*obj*)

Update this resource.



**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvent.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvent.delete()`

Delete this resource.

`CalendarEvent.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CalendarEvent.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.1.7 CalendarEvents

`Calendar.calendar_events()`

Return the resource corresponding to all calendar events.

`CalendarEvents.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvents.delete()`

Delete this resource.

`CalendarEvents.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CalendarEvents.past()`

`CalendarEvents.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Calendar.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Calendar.delete()`

Delete this resource.

`Calendar.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Calendar.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.1.8 Calendars

Basecamp.**calendars** ()

Return the resource corresponding to all calendars.

Calendars.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Calendars.**delete** ()

Delete this resource.

Calendars.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Calendars.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.1.9 GlobalDocuments

Basecamp.**documents** ()

Return the resource corresponding to all documents.

GlobalDocuments.**delete** ()

Delete this resource.

GlobalDocuments.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

GlobalDocuments.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.1.10 Events

Basecamp.**events** ()

Return the resource corresponding to all events.

Events.**delete** ()

Delete this resource.

Events.**get** (*since*, *page=None*)

Fetch all events.

**Variables**

- *since* (*str*) – a datetime.
- *page* (*int*) – the page that will be return. If not indicated, first one is returned.

Events.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.11 People

`Basecamp.people()`

Return the resource corresponding to all people.

`People.delete()`

Delete this resource.

`People.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.1.12 Person, CurrentPerson

`Basecamp.person(person_id=None)`

Return the resource corresponding to a single person. If `person_id` is `None`, current person will be returned.

### 4.1.13 AssignedTodos

`Person.assigned_todos()`

Return the resource corresponding to all assigned todos.

`AssignedTodos.delete()`

Delete this resource.

`AssignedTodos.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`AssignedTodos.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.14 Events

`Person.events()`

Return the resource corresponding to all events.

`Events.delete()`

Delete this resource.

`Events.get(since, page=None)`

Fetch all events.

**Variables**

- `since` (`str`) – a datetime.
- `page` (`int`) – the page that will be return. If not indicated, first one is returned.

`Events.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Person.delete()`  
Delete this resource.

`Person.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CurrentPerson.get()`

#### 4.1.15 Project

`Basecamp.project(project_id)`  
Return the resource corresponding to a single project.

#### 4.1.16 Access

`Project.access(access_id)`  
Return the resource corresponding to a single access.

`Access.revoke()`  
Delete this resource.

#### 4.1.17 Accesses

`Project.accesses()`  
Return the resource corresponding to all project accesses.

`Accesses.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Accesses.grant(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.1.18 Attachments

`Project.attachments()`  
Return the resource corresponding to all attachments.

`Attachments.delete()`  
Delete this resource.

`Attachments.get(page=None)`  
Fetch all resources.

**Variables** `page` (*int*) – the page that will be return. If not indicated, first one is returned.

`Attachments.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.19 CalendarEvent

`Project.calendar_event (calendar_event_id)`  
Return the resource corresponding to a single calendar event.

### 4.1.20 Comments

`CalendarEvent.comments ()`  
Return the resource corresponding to all comments.

`Comments.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.delete ()`  
Delete this resource.

`Comments.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvent.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvent.delete ()`  
Delete this resource.

`CalendarEvent.get ()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CalendarEvent.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.21 CalendarEvents

`Project.calendar_events ()`  
Return the resource corresponding to all calendar events.

`CalendarEvents.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarEvents.delete ()`  
Delete this resource.

`CalendarEvents.get ()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CalendarEvents.past ()`

`CalendarEvents.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.22 Comment

`Project.comment (comment_id)`

Return the resource corresponding to a single comment.

`Comment.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comment.delete ()`

Delete this resource.

### 4.1.23 Document

`Project.document (document_id)`

Return the resource corresponding to a single document.

### 4.1.24 Comments

`Document.comments ()`

Return the resource corresponding to all comments.

`Comments.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.delete ()`

Delete this resource.

`Comments.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Document.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Document.delete ()`

Delete this resource.

`Document.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Document.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.25 Documents

`Project.documents()`

Return the resource corresponding to all documents.

`Documents.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Documents.delete()`

Delete this resource.

`Documents.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Documents.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.26 Events

`Project.events()`

Return the resource corresponding to all events.

`Events.delete()`

Delete this resource.

`Events.get(since, page=None)`

Fetch all events.

**Variables**

- `since` (*str*) – a datetime.
- `page` (*int*) – the page that will be return. If not indicated, first one is returned.

`Events.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.27 Message

`Project.message(message_id)`

Return the resource corresponding to a single message.

### 4.1.28 Comments

Message.**comments** ()

Return the resource corresponding to all comments.

Comments.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Comments.**delete** ()

Delete this resource.

Comments.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Message.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Message.**delete** ()

Delete this resource.

Message.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Message.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.1.29 Messages

Project.**messages** ()

Return the resource corresponding to all project messages.

Messages.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Messages.**delete** ()

Delete this resource.

Messages.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.



### 4.1.30 Todo

`Project.todo(todo_id)`

Return the resource corresponding to a single todo.

### 4.1.31 Comments

`Todo.comments()`

Return the resource corresponding to all comments.

`Comments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.delete()`

Delete this resource.

`Comments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todo.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todo.delete()`

Delete this resource.

`Todo.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Todo.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.32 Todolist

`Project.todolist(todolist_id)`

Return the resource corresponding to a single todolist.

### 4.1.33 Todos

`Todolist.todos()`

Return the resource corresponding to all todos.

`Todos.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todos.delete()`

Delete this resource.

`Todos.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todolist.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todolist.delete()`

Delete this resource.

`Todolist.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Todolist.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.34 Todolists

`Project.todolists()`

Return the resource corresponding to all todolists.

`Todolists.completed()`

`Todolists.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Todolists.delete()`

Delete this resource.

`Todolists.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Todolists.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.35 ProjectTopics

`Project.topics()`

Return the resource corresponding to all project topics.

`ProjectTopics.delete()`

Delete this resource.

`ProjectTopics.get (page=None)`  
 Fetch all topics.

**Variables** `page (int)` – the page that will be return. If not indicated, first one is returned.

`ProjectTopics.update (obj)`  
 Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.1.36 Upload

`Project.upload (upload_id)`  
 Return the resource corresponding to a single upload.

### 4.1.37 Comments

`Upload.comments ()`  
 Return the resource corresponding to all comments.

`Comments.create (obj)`  
 Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.delete ()`  
 Delete this resource.

`Comments.update (obj)`  
 Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Upload.create (obj)`  
 Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Upload.get ()`  
 For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### 4.1.38 Uploads

`Project.uploads ()`  
 Return the resource corresponding to all uploads.

`Uploads.create (obj)`  
 Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Uploads.delete ()`  
 Delete this resource.

Uploads.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Project.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Project.**delete** ()

Delete this resource.

Project.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Project.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.1.39 Projects

Basecamp.**projects** ()

Return the resource corresponding to all projects.

Projects.**archived** ()

Projects.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Projects.**delete** ()

Delete this resource.

Projects.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Projects.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.1.40 GlobalTodolists

Basecamp.**todolists** ()

Return the resource corresponding to all todolists.

GlobalTodolists.**completed** ()

GlobalTodolists.**delete** ()

Delete this resource.

GlobalTodolists.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`GlobalTodolists.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.1.41 Topics

`Basecamp.topics()`

Return the resource corresponding to all topics.

`Topics.delete()`

Delete this resource.

`Topics.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Topics.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2 BitBucket

**class** `bitbucket.BitBucket(username, password=None)`

Create a BitBucket service.

#### Variables

- **username** (`str`) – The username for the authenticated user.
- **password** (`str`) – The password for the authenticated user.

### 4.2.1 Email

`BitBucket.email(email_id)`

Return the resource corresponding to a single email

`Email.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Email.delete()`

Delete this email address from the user account

`Email.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Email.primary()`

Set this email as de primary email.

`Email.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.2 Emails

`BitBucket.emails()`

Return the resource corresponding to all the emails

`Emails.add(address)`

Add an email to the user account.

`Emails.delete()`

Delete this resource.

`Emails.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Emails.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.3 Repo

`BitBucket.repo(user, repo)`

Return the resource corresponding to one repository

## 4.2.4 Changeset

`Repo.changeset(changeset_md5)`

Return a resource corresponding to a changeset for this repo.

`Changeset.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Changeset.delete()`

Delete this resource.

`Changeset.diffstat()`

Return the diffstat for this changeset

`Changeset.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Changeset.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.5 Changesets

`Repo.changesets()`

Return a resource corresponding to all the changesets for this repo.

`Changesets.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Changesets.delete()`

Delete this resource.

`Changesets.get(start='tip', limit=15)`

Fetch changesets

### Variables

- **start** – Changesets start default is 'tip'
- **limit** – Limit of changesets, default is 15

`Changesets.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.6 Repolssue

`Repo.issue(id)`

Return a resource corresponding to an issue from this repo.

## 4.2.7 IssueComment

`RepoIssue.comment(comment_id)`

Return the resource corresponding to a single comment of this issue.

`IssueComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComment.delete()`

Delete this resource.

`IssueComment.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.8 IssueComments

`RepoIssue.comments()`

Return the resource corresponding to the comments of this issue.

`IssueComments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComments.delete()`

Delete this resource.

`IssueComments.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.delete()`

Delete this resource.

`RepoIssue.followers()`

Fetch the followers of this issue.

`RepoIssue.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`RepoIssue.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.9 RepoIssues

`Repo.issues()`

Return a resource corresponding to all issues from this repo.

## 4.2.10 IssueComment

`RepoIssues.component(component_id)`

Return the resources corresponding to one component of this issue.

`IssueComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.



`IssueComment.delete()`

Delete this resource.

`IssueComment.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.11 IssueComponents

`RepoIssues.components()`

Return the resource corresponding to the components of this issue.

`IssueComponents.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComponents.delete()`

Delete this resource.

`IssueComponents.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueComponents.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.12 IssueMilestone

`RepoIssues.milestone(milestone_id)`

Return the resource corresponding to one milestone of this issue.

`IssueMilestone.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueMilestone.delete()`

Delete this resource.

`IssueMilestone.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueMilestone.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.13 IssueMilestones

`RepoIssues.milestones()`

Return the resources corresponding to the milestones of this issue.

`IssueMilestones.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueMilestones.delete()`

Delete this resource.

`IssueMilestones.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueMilestones.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.14 IssueVersion

`RepoIssues.version(version_id)`

Return the resource corresponding to one version of this issue.

`IssueVersion.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueVersion.delete()`

Delete this resource.

`IssueVersion.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IssueVersion.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.15 IssueVersions

`RepoIssues.versions()`

Return the resource corresponding to the versions of this issue.

`IssueVersions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueVersions.delete()`

Delete this resource.

`IssueVersions.get()`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

`IssueVersions.update(obj)`

Update this resource.

**Variables `obj`** – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssues.create(obj)`

Create a new Issue.

**Variables `obj`** – a Python object with the needed params that can be: `title`: The title of the new issue `content`: The content of the new issue `component`: The componen associated with the issue `milestone`: The milestone associated with the issue `version`: The version associated with the issue `responsible`: The username of the person responsible for the issue `priority`: The priority of the issue. Valid priorities are:

- trivial
- minor
- major
- critical
- blocker

`status`: The status of the issue. Val statuses are:

- new
- open
- resolved
- on hold
- invalid
- duplicate
- wontfix

`kind`: The kind of the issue. Valid kinds are:

- bug
- enhancement
- proposal
- task

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.filter(filters)`

Search through the issues applying filters.

Look at <https://confluence.atlassian.com/display/BITBUCKET/Issues> to get a complete list of possible filters.

**Variables `filters`** (*dict of str to str or tuple of str*) – A dictionary of filters.

Keys are strings corresponding to the filter names and values are ether string filter values or tuples, in which case their conditions are implicitly ORed. For example, {"title": ("~one", "~two")} would mean issues with the title containing either “one” or “two”

`RepoIssues.get` (*search=None, start=None, limit=None*)  
Fetch issues for this repository based on the filter parameters.

`RepoIssues.search` (*search=None*)  
Search through issues.

**Variables** *search* – the query string parameter.

`RepoIssues.update` (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.2.16 RepoLink

`Repo.link` (*id*)  
Return a resource corresponding to a link from this repo.

`RepoLink.create` (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoLink.delete` ()  
Delete this resource.

`RepoLink.get` ()  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`RepoLink.update` (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.2.17 RepoLinks

`Repo.links` ()  
Return a resource corresponding to all the links from this repo.

`RepoLinks.create` (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoLinks.delete` ()  
Delete this resource.

`RepoLinks.get` ()  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`RepoLinks.update` (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.2.18 RepoPrivileges

`Repo.privileges` (*specific\_user=None*)

Return a resource corresponding to all privileges from this repo, either for everyone or for a specific user.

`RepoPrivileges.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoPrivileges.delete` ()

Delete this resource.

`RepoPrivileges.get` (*filter=None*)

`RepoPrivileges.grant` (*privilege*)

Grant a privilege on the repo.

**Variables** `privilege` (*str*) – The privilege to grant.

`RepoPrivileges.revoke` ()

Revoke privileges on the repo from the user.

`RepoPrivileges.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.19 Service

`Repo.service` (*service\_id*)

Return a resource corresponding to one service for this repo.

`Service.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Service.delete` ()

Delete this resource.

`Service.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Service.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.20 Services

`Repo.services` ()

Return a resource corresponding to all the services for this repo.

`Services.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Services.delete()`  
Delete this resource.

`Services.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Services.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Repo.branches()`  
Fetch the repository branches.

`Repo.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Repo.delete()`  
Delete this resource.

`Repo.events(start=0, limit=15, etype=None)`  
Fetch events for this repository.

**Variables**

- **start** – Event start, default is 0.
- **limit** – Event result limit, default is 15.
- **type** – Event type, for example 'issue\_comment'.

`Repo.followers()`  
Fetch the followers of this repo.

`Repo.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Repo.invite(user, permission)`  
Invite a user to participate in the repository, with the given permissions.

**Variables**

- **user** (*str*) – The email of the user to invite.
- **permission** (*str*) – The permission to grant (either read or write)

`Repo.tags()`  
Fetch the repository tags.

`Repo.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.21 Repos

`BitBucket.repos()`

Return the resource corresponding to all the repositories

`Repos.create(name, scm=None, is_private=False)`

Create a new repository.

#### Variables

- **name** – the repository name.
- **scm** – the type of repository you want to create, can be: `git`: for git repository `hg`: for mercurial repository

`Repos.delete()`

Delete a repository.

`Repos.get(*args, **kwargs)`

Fetch all repositories you have access to.

`Repos.search(name=None)`

Search for repositories with the given name.

`Repos.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.2.22 User

`BitBucket.user(user_id=None)`

Return the resource corresponding to all the users

### 4.2.23 Group

`User.group(group_name)`

Return a resource corresponding a single user's groups.

This resource only exists for User resources that specify a concrete username.

### 4.2.24 GroupMember

`Group.member(member)`

Return the resource corresponding to a member of the group.

`GroupMember.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`GroupMember.delete()`

Delete this resource.

`GroupMember.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`GroupMember.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.25 GroupMembers

`Group.members ()`

Return the resource corresponding to all members of the group.

`GroupMembers.create (username)`

`GroupMembers.delete ()`

Delete this resource.

`GroupMembers.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`GroupMembers.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Group.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Group.delete ()`

Delete this resource.

`Group.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Group.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.2.26 GroupPrivileges

`User.group_privileges (group=None, repo=None)`

Return a resource corresponding to the group privileges for a user.

This resource only exists for User resources that specify a concrete username.

`GroupPrivileges.get (filter=None, private=None)`

Fetch the group privileges.

**Variables**

- `filter (str)` – Only return specific privileges (read, write, admin).
- `private (bool)` – Only include private repositories.

`GroupPrivileges.grant (group, repo, privilege)`

Grant a privilege for a repository to a group.



GroupPrivileges.**revoke** (*group, repo*)  
 Revoke privileges for a repository from a group.

## 4.2.27 Groups

User.**groups** ()  
 Return a resource corresponding to all of the user's groups.

This resource only exists for User resources that specify a concrete username.

Groups.**create** (*obj*)  
 Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Groups.**delete** ()  
 Delete this resource.

Groups.**get** ()  
 For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Groups.**update** (*obj*)  
 Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

User.**create** (*obj*)  
 Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

User.**delete** ()  
 Delete this resource.

User.**events** (*start=0, limit=15, etype=None*)  
 Fetch events for this user.

### Variables

- **start** – Event start, default is 0.
- **limit** – Event result limit, default is 15.
- **type** – Event type, for example 'issue\_comment'.

User.**followers** ()  
 Fetch the followers of this user.

User.**follows** ()  
 Fetch the list of repositories the authenticated user follows.

User.**get** ()  
 For single-object resources, fetch the object's data. For collections, fetch all of the objects.

User.**update** (*obj*)  
 Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.3 Bitly

**class** bitly.**Bitly** (*token*)

Create a Bitly service.

**Variables** *token* (*str*) – an OAuth 2.0 token.

### 4.3.1 HighValue

Bitly.**highvalue** ()

Return the resource corresponding to all high-value links.

HighValue.**get** (*limit*)

Returns a specified number of “high-value” bitly links that are popular across bitly at this particular moment.

**Variables** *limit* (*int*) – the maximum number of high-value links to return.

### 4.3.2 Link

Bitly.**link** (*link*)

Return the resource corresponding to a single link.

Link.**category** ()

Returns the detected categories for a document, in descending order of confidence.

Link.**clicks** (*unit=None*, *units=None*, *timezone=None*, *rollup=None*, *limit=None*,  
*unit\_reference\_ts=None*)

Returns the number of clicks on a single bitly link.

**Variables**

- **unit** (*str*) – timspace: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

Link.**content** (*content\_type=None*)

Returns the “main article” from the linked page, as determined by the content extractor, in either HTML or plain text format.

**Variables** *content\_type* (*str*) – specifies whether to return the content as html or plain text. if not indicated, defaults to ‘html’.

Link.**countries** (*unit=None*, *units=None*, *timezone=None*, *limit=None*, *unit\_reference\_ts=None*)

Returns metrics about the countries referring click traffic to a single bitly link.

**Variables**

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`Link.encoders_count()`

Returns the number of users who have shortened a single bitly link.

`Link.info()`

Returns metadata about a single bitly link.

`Link.language()`

Returns the significant languages for the bitly link.

`Link.location()`

Returns the significant locations for the bitly link or None if locations do not exist.

`Link.referrers(unit=None, units=None, timezone=None, limit=None, unit_reference_ts=None)`

Returns metrics about the pages referring click traffic to a single bitly link.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`Link.referrers_by_domain(unit=None, units=None, timezone=None, limit=None, unit_reference_ts=None)`

Returns metrics about the pages referring click traffic to a single bitly link, grouped by referring domain.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`Link.referring_domains` (*unit=None, units=None, timezone=None, limit=None, unit\_reference\_ts=None*)

Returns metrics about the domains referring click traffic to a single bitly link.

**Variables**

- **unit** (*str*) – *timspan*: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`Link.shares` (*unit=None, units=None, timezone=None, rollup=None, limit=None, unit\_reference\_ts=None*)

Returns metrics about a shares of a single link.

**Variables**

- **unit** (*str*) – *timspan*: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`Link.social` ()

Returns the “social score” for a specified bitly link.

### 4.3.3 RealTime

`Bitly.realtime` ()

Return the resource corresponding to a single object.

`RealTime.bursting_phrases` ()

Returns phrases that are receiving an uncharacteristically high volume of click traffic, and the individual links (hashes) driving traffic to pages containing these phrases.

`RealTime.clickrate` (*phrase*)

Returns the click rate for content containing a specified phrase.

**Variables** `phrase` (*str*) – the phrase for which you’d like to get the click rate.

`RealTime.hot_phrases` ()

Returns phrases that are receiving a consistently high volume of click traffic, and the individual links (hashes) driving traffic to pages containing these phrases.

### 4.3.4 Search

`Bitly.search()`

Return the resource corresponding to all links.

`Search.get(limit=None, offset=None, query=None, lang=None, cities=None, domain=None, fields=None)`

Search links receiving clicks across bitly by content, language, location, and more

#### Variables

- **limit** (*int*) – the maximum number of links to return.
- **offset** (*int*) – which result to start with (defaults to 0).
- **query** (*str*) – the string to query for.
- **lang** (*str*) – favor results in this language (two letter ISO code).
- **cities** (*str*) – show links active in this city.
- **domain** (*str*) – restrict results to this web domain.
- **fields** (*str*) – which fields to return in the response (comma-separated). May be any of: domain, initial\_epoch, h2, h3, site, lastindexed, keywords, last\_indexed\_epoch, title, initial, summaryText, content, score, summaryTitle, type, description, cities, lang, url, referrer, aggregate\_link, lastseen, page, ogtitle aggregate\_link. By default, all will be returned.

### 4.3.5 User

`Bitly.user()`

Return the resource corresponding to a single user.

`User.clicks(unit=None, units=None, timezone=None, rollup=None, limit=None, unit_reference_ts=None)`

Returns the aggregate number of clicks on all of the authenticated user's bitly links.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

`User.countries(unit=None, units=None, timezone=None, rollup=None, limit=None, unit_reference_ts=None)`

Returns aggregate metrics about the countries referring click traffic to all of the authenticated user's bitly links.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.

- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

User.**info** (*login=None, full\_name=None*)  
 Return or update information about a user.

#### Variables

- **login** (*str*) – the bitly login of the user whose info to look up. If not given, the authenticated user will be used.
- **full\_name** (*str*) – set the users full name value (only available for the authenticated user).

User.**link\_history** (*link=None, limit=None, offset=None, created\_before=None, created\_after=None, modified\_after=None, expand\_client\_id=None, archived=None, private=None, user=None*)  
 Returns entries from a user’s link history in reverse chronological order.

**:var link the bitly link to return metadata for (when specified, overrides all other options).**

**:var limit** the max number of results to return. :vartype login: int

**:var offset** the numbered result at which to start (for pagination). :vartype offset: int

**:var created\_before** timestamp as an integer unix epoch. :vartype created\_before: int

**:var created\_after** timestamp as an integer unix epoch. :vartype created\_after: int

**:var modified\_after** timestamp as an integer unix epoch. :vartype modified\_after: int

**:var expand\_client\_id** whether to provide additional information about encoding application.

**:var archived** whether to include or exclude archived history entries. Defaults to ‘off’.

**:var private** whether to include or exclude private history entries. Defaults to ‘both’.

**Variables** **user** (*str*) – the user for whom to retrieve history entries (if different from authenticated user).

User.**network\_history** (*limit=None, offset=None, expand\_client\_id=None, expand\_user=None*)  
 Returns entries from a user’s network history in reverse chronological order.

**:var limit** the max number of results to return. :vartype login: int

**:var offset** the numbered result at which to start (for pagination). :vartype offset: int

**:var expand\_client\_id** whether to provide additional information about encoding application.

**:var expand\_user** include extra user info in response. **:vartype expand\_user:** bool

User.**popular\_links** (*unit=None, units=None, timezone=None, limit=None, unit\_reference\_ts=None*)  
Returns the authenticated user's most-clicked bitly links (ordered by number of clicks) in a given time period.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

User.**referrers** (*unit=None, units=None, timezone=None, rollup=None, limit=None, unit\_reference\_ts=None*)  
Returns aggregate metrics about the pages referring click traffic to all of the authenticated user's bitly links.

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

User.**referring\_domains** (*unit=None, units=None, timezone=None, rollup=None, limit=None, unit\_reference\_ts=None*)  
Returns aggregate metrics about the domains referring click traffic to all of the authenticated user's bitly links

#### Variables

- **unit** (*str*) – timspan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.

- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

User.**share\_counts** (*unit=None, units=None, timezone=None, rollup=None, limit=None, unit\_reference\_ts=None*)

Returns the number of shares by the authenticated user in a given time period.

#### Variables

- **unit** (*str*) – timsan: minute, hour, day, week or month. When unit is minute the maximum value for units is 60. if\* not indicated, defaults to day.
- **units** (*int*) – an integer representing the time units to query data for. If -1 is passed, it will return all units of time.
- **timezone** (*str*) – an integer hour offset from UTC (-14..14) or a timezone string. If not indicated, defaults to America/New\_York.
- **rollup** (*bool*) – returns data for multiple units rolled up to a single result instead of a separate value for each period of time.
- **limit** (*int*) – the number of rows it will return. Default is 100.
- **unit\_reference\_ts** (*int*) – an epoch timestamp, indicating the most recent time for which to pull metrics. If not indicated, it defaults to now.

User.**tracking\_domain\_list** ()

Returns a list of tracking domains a user has configured.

## 4.4 CartoDB

**class** cartodb.**CartoDB** (*subdomain, api\_key*)

Create a CartoDB service.

#### Variables

- **subdomain** (*str*) – The account-specific part of the CartoDB domain, for instance use *mycompany* if your CartpDB domain is *mycompany.cartodb.com*.
- **api\_key** (*str*) – The API key.

### 4.4.1 Service methods

CartoDB.**sql** (*q*)

SQL request to the CartoDB account

**Variables** **q** (*str*) – The sql query

CartoDB.**viz** (*type=None*)

Get the list of visualizations (undocumented endpoint)

**Variables** **type** – The visualization type



## 4.5 Compete

**class** `compete.Compete` (*api\_key*)

Create an Compete service.

**Variables** `api_key` (*str*) – The API key.

### 4.5.1 Site

`Compete.site` (*domain*)

Return the resource corresponding to a single site.

### 4.5.2 Metric

`Site.metric` (*metric\_id*)

Return the resource corresponding to a single metric for the site.

`Metric.get` (*latest=None, start\_date=None, end\_date=None*)

Fetch the object's data.

#### Variables

- **latest** (*int*) – Returns the latest N months or days. If omitted, it returns data for the most recent 13 months for a monthly metric. For daily metrics, it returns data for the most recent 30 days.
- **start\_date** (*str*) – Return specific start date. If omitted, it returns data for the most recent 13 months for a monthly metric. For daily metrics, it returns data for the most recent 30 days.
- **end\_date** (*str*) – Returns specific end date. If omitted, it returns data for the most recent 13 months for a monthly metric. For daily metrics, it returns data for the most recent 30 days.

## 4.6 Desk

**class** `desk.Desk` (*subdomain, api\_key, api\_secret=None, access\_token=None, access\_token\_secret=None*)

Create a Desk service.

#### Variables

- **subdomain** (*str*) – The account-specific part of the Desk domain, for instance use *mycompany* if your Desk domain is *mycompany.desk.com*, or the full domain if using Desk whitelabel, for instance *support.mycompany.com*. If the parameter contains a dot, it is treated as a full domain, otherwise as a subdomain.
- **api\_key** (*str*) – The API key.
- **api\_secret** (*str*) – API secret.
- **access\_token** (*str*) – OAuth 1.0a access token.
- **access\_token\_secret** (*str*) – OAuth 1.0a access token secret. requests.

### 4.6.1 Article

`Desk.article (article_id)`

Return the resource corresponding to a single article.

`Article.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Article.delete ()`

Delete this resource.

`Article.get (embed=None, fields=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Article.update (obj)`

### 4.6.2 Case

`Desk.case (case_id, is_external=False)`

Return the resource corresponding to a single case.

**Variables**

- `case_id (bool)` – The case id
- `is_external` – Use the external id

### 4.6.3 Replies

`Case.replies ()`

Return the resource corresponding to the case replies

`Replies.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Replies.delete ()`

Delete this resource.

`Replies.get (embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Replies.update (obj)`

### 4.6.4 Reply

`Case.reply (reply_id)`

Return the resource corresponding to a single reply

`Reply.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Reply.delete()`

Delete this resource.

`Reply.get(embed=None, fields=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Reply.update(obj)`

`Case.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Case.delete()`

Delete this resource.

`Case.get(embed=None, fields=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Case.history(per_page=None, page=None)`

The case history endpoint will display a paginated list of all events/actions that have happened to the case

Upstream documentation: <http://dev.desk.com/API/cases#history>

`Case.message()`

Retrieve the original message for this case.

Upstream documentation: <http://dev.desk.com/API/cases#message-show>

`Case.update(obj)`

## 4.6.5 Cases

`Desk.cases()`

Return the resource corresponding to all the cases.

`Cases.get(embed=None, fields=None, per_page=None, page=None)`

Retrieve a paginated list of all cases.

Upstream documentation: <http://dev.desk.com/API/cases#list>

`Cases.search(name=None, first_name=None, last_name=None, email=None, phone=None, company=None, twitter=None, labels=None, case_id=None, subject=None, description=None, status=None, priority=None, assigned_group=None, assigned_user=None, channels=None, notes=None, attachments=None, created=None, updated=None, since_created_at=None, max_created_at=None, since_updated_at=None, max_updated_at=None, since_id=None, max_id=None, per_page=None, page=None, embed=None, fields=None, **case_custom_fields)`

Search cases based on a combination of parameters with pagination.

Upstream documentation: <http://dev.desk.com/API/cases#search>

## 4.6.6 Companies

`Desk.companies()`

Return the resource corresponding to all companies.

`Companies.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Companies.get(embed=None, fields=None, per_page=None, page=None)`

Retrieve a paginated list of all companies

Upstream documentation: <http://dev.desk.com/API/companies/#list>

`Companies.search(q, per_page=None, page=None, sort_field=None, sort_direction=None)`

Search companies based on a search parameter with pagination.

Upstream documentation: <http://dev.desk.com/API/companies/#search>

## 4.6.7 Company

`Desk.company(company_id)`

Return the resource corresponding to a single company.

## 4.6.8 Cases

`Company.cases()`

`Cases.get(embed=None, fields=None, per_page=None, page=None)`

Retrieve a paginated list of all cases.

Upstream documentation: <http://dev.desk.com/API/cases#list>

`Cases.search(name=None, first_name=None, last_name=None, email=None, phone=None, company=None, twitter=None, labels=None, case_id=None, subject=None, description=None, status=None, priority=None, assigned_group=None, assigned_user=None, channels=None, notes=None, attachments=None, created=None, updated=None, since_created_at=None, max_created_at=None, since_updated_at=None, max_updated_at=None, since_id=None, max_id=None, per_page=None, page=None, embed=None, fields=None, **case_custom_fields)`

Search cases based on a combination of parameters with pagination.

Upstream documentation: <http://dev.desk.com/API/cases#search>

## 4.6.9 Customers

`Company.customers()`

`Customers.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

Customers.**get** (*embed=None, fields=None, per\_page=None, page=None*)

Retrieve a paginated list of all customers

Upstream documentation: <http://dev.desk.com/API/customers#list>

Customers.**search** (*first\_name=None, last\_name=None, full\_name=None, email=None, phone=None, twitter=None, external\_id=None, since\_created\_at=None, max\_created\_at=None, since\_updated\_at=None, max\_updated\_at=None, since\_id=None, max\_id=None, per\_page=None, page=None, \*\*custom\_fields*)

Search customers based on a combination of parameters with pagination.

Upstream documentation: <http://dev.desk.com/API/customers#search>

Company.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Company.**get** (*embed=None, fields=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Company.**update** (*obj*)

## 4.6.10 Customer

Desk.**customer** (*customer\_id=None*)

Return the resource corresponding to a single customer.

## 4.6.11 Cases

Customer.**cases** ()

Cases.**get** (*embed=None, fields=None, per\_page=None, page=None*)

Retrieve a paginated list of all cases.

Upstream documentation: <http://dev.desk.com/API/cases#list>

Cases.**search** (*name=None, first\_name=None, last\_name=None, email=None, phone=None, company=None, twitter=None, labels=None, case\_id=None, subject=None, description=None, status=None, priority=None, assigned\_group=None, assigned\_user=None, channels=None, notes=None, attachments=None, created=None, updated=None, since\_created\_at=None, max\_created\_at=None, since\_updated\_at=None, max\_updated\_at=None, since\_id=None, max\_id=None, per\_page=None, page=None, embed=None, fields=None, \*\*case\_custom\_fields*)

Search cases based on a combination of parameters with pagination.

Upstream documentation: <http://dev.desk.com/API/cases#search>

Customer.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Customer.**get** (*embed=None, fields=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Customer.update` (*obj*)

## 4.6.12 Customers

`Desk.customers` ()

Return the resource corresponding to all customers.

`Customers.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Customers.get` (*embed=None, fields=None, per\_page=None, page=None*)

Retrieve a paginated list of all customers

Upstream documentation: <http://dev.desk.com/API/customers#list>

`Customers.search` (*first\_name=None, last\_name=None, full\_name=None, email=None, phone=None, twitter=None, external\_id=None, since\_created\_at=None, max\_created\_at=None, since\_updated\_at=None, max\_updated\_at=None, since\_id=None, max\_id=None, per\_page=None, page=None, \*\*custom\_fields*)

Search customers based on a combination of parameters with pagination.

Upstream documentation: <http://dev.desk.com/API/customers#search>

## 4.6.13 Group

`Desk.group` (*group\_id*)

Return the resource corresponding to a single group.

`Group.get` (*embed=None, fields=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Group.group_filters` (*per\_page=None, page=None*)

Retrieve a paginated list of all filters for the given group.

Upstream documentation: <http://dev.desk.com/API/groups#list-filters>

`Group.users` (*per\_page=None, page=None*)

Retrieve a paginated list of all users for the given group.

Upstream documentation: <http://dev.desk.com/API/groups#list-users>

## 4.6.14 Groups

`Desk.groups` ()

Return the resource corresponding to all groups.

`Groups.get` (*embed=None, fields=None, per\_page=None, page=None*)

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

### 4.6.15 Insights

`Desk.insights()`

Return the resource corresponding to insights.

`Insights.meta()`

Retrieve Insights meta data for the authenticated site.

Upstream documentation: <http://dev.desk.com/API/insights/#meta-show>

`Insights.report(resolution=None, min_date=None, max_date=None, dimension1_name=None, dimension1_values=None, dimension2_name=None, dimension2_values=None, metrics=None, sort_by=None, sort_order=None, dimension1_per_page=None, dimension1_page=None)`

Create a report.

Upstream documentation: <http://dev.desk.com/API/insights/#report-create>

### 4.6.16 Macro

`Desk.macro(macro_id)`

Return the resource corresponding to a single macro.

### 4.6.17 Actions

`Macro.action(action_id)`

Return the resource corresponding to single macro action

`Actions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Actions.delete()`

Delete this resource.

`Actions.get(embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Actions.update(obj)`

### 4.6.18 Actions

`Macro.actions()`

Return the resource corresponding to macro actions

`Actions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Actions.delete()`

Delete this resource.

Actions.**get** (*embed=None, fields=None, per\_page=None, page=None*)

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Actions.**update** (*obj*)

Macro.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Macro.**delete** ()

Delete this resource.

Macro.**get** (*embed=None, fields=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Macro.**update** (*obj*)

### 4.6.19 Macros

Desk.**macros** ()

Return the resource corresponding to all macros.

Macros.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Macros.**delete** ()

Delete this resource.

Macros.**get** (*embed=None, fields=None, per\_page=None, page=None*)

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

### 4.6.20 SiteSettings

Desk.**site\_settings** ()

Return the resource corresponding to the site settings.

SiteSettings.**get** (*embed=None, fields=None, per\_page=None, page=None*)

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

### 4.6.21 Topic

Desk.**topic** (*topic\_id*)

Return the resource corresponding to a single topic.



## 4.6.22 Articles

`Topic.articles()`

Return the resource corresponding to topic articles

## 4.6.23 Translation

`Articles.translation(translation_id)`

Return the resource corresponding to a single translation

`Translation.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Translation.delete()`

Delete this resource.

`Translation.get(embed=None, fields=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Translation.update(obj)`

## 4.6.24 Translations

`Articles.translations()`

Return the resource corresponding to the article translations

`Translations.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Translations.delete()`

Delete this resource.

`Translations.get(embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Translations.update(obj)`

`Articles.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Articles.delete()`

Delete this resource.

`Articles.get(embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Articles.**search** (*text=None, topic\_ids=None, per\_page=None, page=None*)  
Perform a search across all public articles.

Upstream documentation: <http://dev.desk.com/API/articles#search>

Articles.**update** (*obj*)

#### 4.6.25 Translation

Topic.**translation** (*translation\_id*)  
Return the resource corresponding to a single translation

Translation.**create** (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Translation.**delete** ()  
Delete this resource.

Translation.**get** (*embed=None, fields=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Translation.**update** (*obj*)

#### 4.6.26 Translations

Topic.**translations** ()  
Return the resource corresponding to the topic translations

Translations.**create** (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Translations.**delete** ()  
Delete this resource.

Translations.**get** (*embed=None, fields=None, per\_page=None, page=None*)  
Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

Translations.**update** (*obj*)

Topic.**create** (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Topic.**delete** ()  
Delete this resource.

Topic.**get** (*embed=None, fields=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`Topic.update (obj)`

## 4.6.27 Topics

`Desk.topics ()`

Return the resource corresponding to all topics.

`Topics.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Topics.delete ()`

Delete this resource.

`Topics.get (embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

## 4.6.28 User

`Desk.user (user_id)`

Return the resource corresponding to a single user.

`User.get (embed=None, fields=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

`User.preference (preference_id)`

Show a single user preference

Upstream documentation: <http://dev.desk.com/API/users/#preferences-show>

`User.preferences (per_page=None, page=None)`

List all of the user's preferences.

Upstream documentation: <http://dev.desk.com/API/users/#preferences-list>

`User.update_preference (preference_id, obj)`

Update a user preference

Upstream documentation: <http://dev.desk.com/API/users/#preferences>

## 4.6.29 Users

`Desk.users ()`

Return the resource corresponding to all users.

`Users.get (embed=None, fields=None, per_page=None, page=None)`

Returns a paginated list of elements

Upstream documentation: <http://dev.desk.com/API/using-the-api/>

## 4.7 Ducksboard

**class** ducksboard.**Ducksboard** (*apikey\_or\_username*, *password=""*)

Create a Ducksboard service.

### Variables

- **apikey\_or\_username** (*str*) – Your apikey or your username if you want to get or reset your API key.
- **password** (*str*) – Only used with your username to get or reset your API key.

### 4.7.1 Account

Ducksboard.**account** (*account\_id*)

Return the resource corresponding to a single account.

Account.**delete** ()

Delete this resource.

Account.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.7.2 Accounts

Ducksboard.**accounts** ()

Return the resource corresponding to all the accounts.

Accounts.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Accounts.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.7.3 Dashboard

Ducksboard.**dashboard** (*slug*)

Return the resource corresponding to a single dashboard.

### 4.7.4 Token

Dashboard.**token** (*token*)

Token.**delete** ()

Delete this resource.

Token.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.7.5 Tokens

Dashboard.**tokens** ()

Tokens.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Tokens.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Dashboard.**accessed** ()

Update the access time of a dashboard. The last accessed dashboard is the one that gets displayed by default when accessing the application.

Dashboard.**delete** ()

Delete this resource.

Dashboard.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Dashboard.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Dashboard.**widgets** ()

Get a collection of widgets from a dashboard.

## 4.7.6 Dashboards

Ducksboard.**dashboards** ()

Return the resource corresponding to all the dashboards.

Dashboards.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Dashboards.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.7.7 Datasource

Ducksboard.**data\_source** (*label*)

Return the resource corresponding to a datasource. Datasources can only be accessed using the API key

Datasource.**delete** ()

Delete all data for a given data source.

Datasource.**last** (*count=None*)

Get the last count values for a given data source, ordered by their timestamp, newest data first. This resource can be used for all data sources.

**Variables** *count* (*int*) – The amount of data returned. It might be less than the count parameter. The default value for count is 3 and the maximum is 100.

Datasource.**push** (*obj*)

Send a value or a list of values. Each value can have a timestamp associated with it. Timestamps should be UNIX timestamps expressed as numbers. If no timestamp is specified, the value is assumed to be timestamped with the current time.

**Variables** **obj** – a Python object representing the value to push to the data source. See <http://dev.ducksboard.com/apidoc/push-api/#post-values-label>

Datasource.**since** (*seconds=None*)

Get the values from up to seconds ago for a given data source, ordered by their timestamp, newest data first. This resource can be used for all data sources.

**Variables** **seconds** (*int*) – The first value returned might actually be from later than seconds ago. The default value for seconds is 3600 and the maximum is 7776000.

Datasource.**timespan** (*timespan=None, timezone=None*)

Get the last value for a series of periods for a given data source. The number of values returned depends on the timespan parameter. If a certain period is empty, meaning that no values from inside of it are found, the value of the previous period is substituted, or null if no previous values were found. See <http://dev.ducksboard.com/apidoc/pull-api-http/#get-values-label-timespan-timespan-timespan-timezone-timezone>

**Variables**

- **timespan** (*str*) – The allowed values for timespan are daily, weekly and monthly, with the default of monthly.
- **timezone** (*str*) – The limits of periods are actually dependent on the timezone parameter, as depending on which timezone you want to see the data in, the last value of each period might be different. The default for timezone is UTC.

## 4.7.8 Account

Ducksboard.**user** ()

Return the resource corresponding to your user.

Account.**delete** ()

Delete this resource.

Account.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.7.9 Widget

Ducksboard.**widget** (*widget\_id*)

Return the resource corresponding to a single widget.

Widget.**copy** (*dashboard\_slug*)

Copy a widget to another dashboard, specified by a slug. A new widget is created, with the same parameters as the copied one. The position is chosen automatically if not specified.

**Variables** **dashboard\_slug** (*str*) – dashboard slug destination

Widget.**delete** ()

Delete this resource.

Widget.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Widget.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.7.10 Widgets

`Ducksboard.widgets()`

Return the resource corresponding to all the widgets.

`Widgets.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Widgets.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Widgets.positions(positions)`

Update the positions of multiple widgets at once.

**Variables** `positions` (*dict*) – The object keys are widget IDs. the values should include a row and column field. Both of them default to 1 if not present.

## 4.8 Flurry

`class flurry.Flurry(api_access_code)`

Create a Flurry service.

**Variables** `api_access_code` (*str*) – The API access code.

### 4.8.1 Application

`Flurry.application(application_api_key)`

Returns the resource corresponding to a single application.

### 4.8.2 Event

`Application.event(event_name)`

Returns the resource corresponding to a single event.

`Event.get(start_date, end_date, version_name=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **version\_name** (*str*) – optional parameter indicating application's version.

### 4.8.3 Events

`Application.events()`

Return the resource corresponding to all events.

`Events.get(start_date, end_date, version_name=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **version\_name** (*str*) – optional parameter indicating application's version.

### 4.8.4 Metrics

`Application.metrics()`

Returns the resource corresponding to all metrics.

`Metrics.active_users(*args, **kwargs)`

Returns the total number of unique users who accessed the application per day.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.active_users_by_month(*args, **kwargs)`

Returns the total number of unique users who accessed the application per month.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.active_users_by_week(*args, **kwargs)`

Returns the total number of unique users who accessed the application per week

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.



- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.avg_page_views_per_session(*args, **kwargs)`

Returns the average page views per session for each day.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.avg_session_length(*args, **kwargs)`

Returns the average length of a user session per day.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.median_session_length(*args, **kwargs)`

Returns the median length of a user session per day.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.new_users(*args, **kwargs)`

Returns the total number of unique users who used the application for the first time per day.

#### Variables

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user's country.
- **version\_name** (*str*) – optional parameter indicating application's version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.page_views (*args, **kwargs)`

Returns the total number of page views per day.

**Variables**

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user’s country.
- **version\_name** (*str*) – optional parameter indicating application’s version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.retained_users (*args, **kwargs)`

Returns the total number of users who remain active users of the application per day.

**Variables**

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user’s country.
- **version\_name** (*str*) – optional parameter indicating application’s version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Metrics.sessions (*args, **kwargs)`

Returns the total number of times users accessed the application per day.

**Variables**

- **start\_date** (*str*) – the first date to look metrics for.
- **end\_date** (*str*) – the last date to look metrics for.
- **country** (*str*) – optional parameter indicating user’s country.
- **version\_name** (*str*) – optional parameter indicating application’s version.
- **group\_by** (*str*) – group data by DAYS, WEEKS or MONTHS. By default, it will group data by days.

`Application.get ()`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

## 4.8.5 Applications

`Flurry.applications ()`

Return the resource corresponding to all applications.

`Applications.get ()`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

## 4.9 Fullcontact

**class** `fullcontact.Fullcontact` (*api\_key*)

Create a Fullcontact service.

**Variables** `api_key` (*str*) – The API key.

### 4.9.1 Enhanced

`Fullcontact.enhanced` (*email*)

Return the resource corresponding to a single person

`Enhanced.get` ()

Fetch a single object.

### 4.9.2 Location

`Fullcontact.locations` ()

Return the resource corresponding to all locations.

`Location.enrichment` (*place, includeZeroPopulation=False, casing=None*)

Return a collection of lostructured location data for a indicated place.

#### Variables

- **place** (*str*) – The place you are interested in.
- **includeZeroPopulation** (*bool*) – Will include 0 population census locations.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

`Location.normalizer` (*place, includeZeroPopulation=False, casing=None*)

Return structured location data for a indicated place.

#### Variables

- **place** (*str*) – The place you are interested in.
- **includeZeroPopulation** (*bool*) – Will include 0 population census locations.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

### 4.9.3 Name

`Fullcontact.names` ()

Return the resource corresponding to all names.

`Name.deducer` (*email=None, username=None, casing=None*)

Take a username or email address provided as a string and attempts to deduce a structured name.

#### Variables

- **email** (*str*) – It allows you to pass an email address.
- **username** (*str*) – It allows you to pass a username.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

`Name.normalizer` (*q, casing=None*)

Take quasi-structured name data provided as a string and outputs the data in a structured manner.

#### Variables

- **q** (*str*) – Name you would like to be normalized.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

Name **.parser** (*q, casing=None*)

Determine what the given name and family name for a ambiguous name.

#### Variables

- **q** (*str*) – Name you would like to be parsed.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

Name **.similarity** (*q1, q2, casing=None*)

Return a score indicating how two names are similar.

#### Variables

- **q1** (*str*) – First name to compare.
- **q2** (*str*) – Second name to compare.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

Name **.stats** (*name=None, givenName=None, familyName=None, casing=None*)

Determine more about a name.

#### Variables

- **name** (*str*) – It can be used when you only know a single name and you are uncertain whether it is the given name or family name.
- **givenName** (*str*) – It can be used when you know that the name is a first name.
- **familyName** (*str*) – It can be used when you know that the name is a last name.
- **casing** (*str*) – One of: uppercase, lowercase or titlecase.

### 4.9.4 Person

Fullcontact **.person** (*email=None, emailMD5=None, phone=None, twitter=None, facebookUserName=None*)

Return the resource corresponding to a single person

Person **.get** (*queue=None, style=None, prettyPrint=None, countryCode=None*)

Fetch a single object.

#### Variables

- **queue** (*int*) – Using this parameter notifies FullContact that the query in question will be called later.
- **style** (*str*) – The style parameter can be used to control the document structure returned. Only available for email lookups.
- **prettyPrint** (*str*) – Used to disable prettyprint formatting response
- **countryCode** (*str*) – For phone lookups, it must be passed when using non US/Canada based numbers. Use the ISO-3166 two-digit country code. It defaults to US.

## 4.10 GitHub

**class** `github.GitHub` (*token\_or\_username*, *password=None*, *apiroot='https://api.github.com'*)

Create a GitHub service.

### Variables

- **token\_or\_username** (*str*) – Either an OAuth 2.0 token, or the username if you want to use Basic authentication.
- **password** (*str*) – Only used with the Basic authentication, leave this as *None* when using OAuth.
- **apiroot** (*str*) – Only used for GitHub Enterprise, defaults to GitHub api url

### 4.10.1 Authorization

`GitHub.authorization` (*authorization\_id*)

Return the resource corresponding to a single authorization. Authorizations can only be accessed when using Basic authentication.

`Authorization.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Authorization.delete` ()

Delete this resource.

`Authorization.get` (*page=None*, *per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Authorization.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.2 Authorizations

`GitHub.authorizations` ()

Return the resource corresponding to all the authorizations. Authorizations can only be accessed when using Basic authentication.

`Authorizations.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Authorizations.delete()`

Delete this resource.

`Authorizations.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Authorizations.update (obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.3 Gist

`GitHub.gist (gist_id)`

Return the resource corresponding to a single gist.

`Gist.create (obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Gist.delete ()`

Delete this resource.

`Gist.fork ()`

Fork this gist.

`Gist.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Gist.is_starred ()`

Check if this gist is starred.

**Returns** `bool`

`Gist.star ()`

Star this gist.

`Gist.unstar ()`

Unstar this gist.

`Gist.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.4 Gists

`GitHub.gists()`

Return the resource corresponding to all the gists.

#### 4.10.5 GistComment

`Gists.comment(comment_id)`

Return the resource corresponding to a single comment on a gist.

When updating comments, use a simple string as the parameter to `update`, you don't have to use `{“body”:<comment body>}`.

`GistComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`GistComment.delete()`

Delete this resource.

`GistComment.get(format=None, page=None, per_page=None)`

`GistComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Gists.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Gists.delete()`

Delete this resource.

`Gists.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Gists.public(page=None, per_page=None)`

Fetch public gists. The parameters are the same as for `get`.

`Gists.starred(page=None, per_page=None)`

Fetch gists starred by the authenticated user. The parameters are the same as for `get`.

`Gists.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.6 Issues

`GitHub.issues()`

Return the resource corresponding to all the issues of the authenticated user.

`Issues.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Issues.delete()`

Delete this resource.

`Issues.get(filter='assigned', state='open', labels=None, sort='created', direction='desc', since=None, format=None, page=None, per_page=None)`

Fetch the authenticated user's issues based on the filter parameters, and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues>.

`Issues.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.7 Organizations

`GitHub.org(org)`

Return the resource corresponding to a single organization.

## 4.10.8 Member

`Organizations.member(user)`

Return a resource corresponding to a member of this org.

`Member.delete()`

Delete this resource.

`Member.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.



### 4.10.9 Members

`Organizations.members()`

Return a resource corresponding to members of this org.

`Members.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.10 PublicMember

`Organizations.public_member(user)`

Return a resource corresponding to a public member of this org.

`PublicMember.delete()`

Delete this resource.

`PublicMember.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PublicMember.publicize()`

### 4.10.11 PublicMembers

`Organizations.public_members()`

Return a resource corresponding to public members of this org.

`PublicMembers.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.12 OrganizationRepo

`Organizations.repo(repo)`

Return a resource corresponding to single repo for this org.

### 4.10.13 RepoCollaborators

`OrganizationRepo.collaborators()`

Return a resource corresponding to all collaborators in this repo.

`RepoCollaborators.add(user)`

Add a collaborator to this repo.

**Variables** `user` (*str*) – The username of the new collaborator.

`RepoCollaborators.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator(user)`

Check if a user is a collaborator in this repo.

**Variables** `user` (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove(user)`

Remove a collaborator from this repo.

**Variables** `user` (*str*) – The username of the collaborator.

### 4.10.14 RepoCommit

`OrganizationRepo.commit(sha)`

Return a resource corresponding to a single commit in this repo.

### 4.10.15 RepoCommitsComments

`RepoCommit.comments()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables `obj`** – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommit.get()`

Fetch all commits from this repo.

#### 4.10.16 RepoCommits

`OrganizationRepo.commits()`

Return a resource corresponding to all commits in this repo.

#### 4.10.17 RepoCommitsComment

`RepoCommits.comment(comment_id)`

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`

Create a new resource.

**Variables `obj`** – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables `format`** – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables `obj`** – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.18 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables `comment` (`str`)** – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables `format`** – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommits.compare(base, head)`

Fetch the comparison of two commits.

**Variables**

- **base** (*str*) – The commit hash of the first commit.
- **head** (*str*) – The commit hash of the second commit.

`RepoCommits.get(sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- **sha** (*str*) – Optional commit hash or branch to start listing commits from.
- **path** (*str*) – Optional filter to only include commits that include this file path.

### 4.10.19 RepoContents

`OrganizationRepo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink(archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get(path=None, ref=None)`

This method returns the contents of any file or directory in a repository.

**Variables**

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme(ref=None)`

This method returns the preferred README for a repository.

**Variables** `ref` (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.20 Download

`OrganizationRepo.download(download_id)`

Return a resource corresponding to a single download in this repo.

`Download.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Download.delete()`

Delete this resource.

Download.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.21 Downloads

OrganizationRepo.**downloads** ()

Return a resource corresponding to all downloads from this repo.

Downloads.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Downloads.**delete** ()

Delete this resource.

Downloads.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.22 Forks

OrganizationRepo.**forks** ()

Return a resource corresponding to all forks of this repo.

Forks.**create** ()

Fork this repo.

Forks.**get** (*sort='newest', page=None, per\_page=None*)

Fetch this repo's forks.

**Variables**

- **sort** (*str*) – The sort order for the result.
- **page** (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- **per\_page** (*int*) – The amount of results per page.

### 4.10.23 RepoHook

OrganizationRepo.**hook** (*hook\_id*)

Return a resource corresponding to a single hook in this repo.

`RepoHook.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete ()`  
Delete this resource.

`RepoHook.get (page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoHook.ping ()`  
Send a ping event to the hook.

`RepoHook.test ()`  
Trigger the hook with the latest push to the repository.

`RepoHook.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.24 RepoHooks

`OrganizationRepo.hooks ()`  
Return a resource corresponding to all hooks of this repo.

`RepoHooks.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHooks.delete ()`  
Delete this resource.

`RepoHooks.get (page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoHooks.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.25 Repolssue

OrganizationRepo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

### 4.10.26 IssueComments

RepoIssue.**comments** ()

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to *create*, you don't have to use *{"body": <comment body>}*.

IssueComments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComments.**delete** ()

Delete this resource.

IssueComments.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueComments.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.27 IssueEvent

RepoIssue.**event** (*event\_id*)

Return the resource corresponding to a single event of this issue.

IssueEvent.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.28 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.29 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.30 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabels.delete()`

Delete all labels from this issue.



IssueLabels.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueLabels.**replace** (*labels*)

Replace all labels on this issue with new ones.

**Variables** *labels* (*list of str*) – A list of labels to use.

IssueLabels.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssue.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssue.**delete** ()

Delete this resource.

RepoIssue.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoIssue.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.31 RepoIssues

OrganizationRepo.**issues** ()

Return a resource corresponding to all issues from this repo.

### 4.10.32 IssueComment

RepoIssues.**comment** (*comment\_id*)

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to *update*, you don’t have to use *{“body”:* *<comment body>}*.

IssueComment.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComment.**delete** ()

Delete this resource.

IssueComment.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueComment.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.33 IssueEvents

RepoIssues.**events** ()

Return the resource corresponding to all events of this repo's issues.

IssueEvents.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoIssues.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssues.**delete** ()

Delete this resource.

RepoIssues.**get** (*milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per\_page=None*)

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

RepoIssues.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.34 RepoKey

`OrganizationRepo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKey.delete()`

Delete this resource.

`RepoKey.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKey.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.35 RepoKeys

`OrganizationRepo.keys()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`

Delete this resource.

`RepoKeys.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKeys.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.36 RepoLabel

`OrganizationRepo.label (name)`

Return a resource corresponding to a single label from this repo.

`RepoLabel.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete ()`

Delete this resource.

`RepoLabel.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoLabel.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.37 RepoLabels

`OrganizationRepo.labels ()`

Return a resource corresponding to all issues from this repo.

`RepoLabels.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete ()`

Delete this resource.

`RepoLabels.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoLabels.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.38 Milestone

`OrganizationRepo.milestone (milestone_id)`

Return a resource corresponding to a single milestone in this repo.

### 4.10.39 MilestoneLabels

`Milestone.labels ()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MilestoneLabels.delete ()`

Delete this resource.

`MilestoneLabels.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`MilestoneLabels.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestone.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestone.delete ()`

Delete this resource.

`Milestone.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Milestone.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.40 Milestones

`OrganizationRepo.milestones()`

Return a resource corresponding to all milestones in this repo.

`Milestones.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestones.delete()`

Delete this resource.

`Milestones.get(state='open', sort='due_date', direction='desc', page=None, per_page=None)`

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

`Milestones.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.41 PullRequest

`OrganizationRepo.pullrequest(number)`

Return a resource corresponding to a single pull request for this repo.

`PullRequest.commits()`

Fetch commits on this pull request.

`PullRequest.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequest.delete()`

Delete this resource.

`PullRequest.files()`

Fetch files on this pull request.

`PullRequest.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged()`

Check if this pull request has been merged.

`PullRequest.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.42 PullRequests

`OrganizationRepo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.43 Release

`OrganizationRepo.release(release_id)`

Return a resource corresponding to a single release in this repo.

#### 4.10.44 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`

Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.45 ReleaseAssets

Release.**assets** ()

ReleaseAssets.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Release.**delete** ()

Delete this resource.

Release.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.46 Releases

OrganizationRepo.**releases** ()

Return a resource corresponding to all releases from this repo.

Releases.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Releases.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.



OrganizationRepo.**branches** ()  
Fetch the branches for this repo.

OrganizationRepo.**contributors** (*anon=False*)  
Fetch the contributors from this repo.

**Variables** *anon* (*bool*) – Include anonymous contributors.

OrganizationRepo.**create** (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

OrganizationRepo.**delete** ()  
Delete this resource.

OrganizationRepo.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

OrganizationRepo.**languages** ()  
Fetch the languages for this repo.

OrganizationRepo.**tags** ()  
Fetch the tags for this repo.

OrganizationRepo.**teams** ()  
Fetch the teams for this repo.

OrganizationRepo.**update** (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.47 Repo

Organizations.**repos** ()  
Return a resource corresponding to repos for this org.

#### 4.10.48 RepoCollaborators

Repo.**collaborators** ()  
Return a resource corresponding to all collaborators in this repo.

RepoCollaborators.**add** (*user*)  
Add a collaborator to this repo.

**Variables** *user* (*str*) – The username of the new collaborator.

RepoCollaborators.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator (user)`

Check if a user is a collaborator in this repo.

**Variables** `user` (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove (user)`

Remove a collaborator from this repo.

**Variables** `user` (*str*) – The username of the collaborator.

### 4.10.49 RepoCommit

`Repo.commit (sha)`

Return a resource corresponding to a single commit in this repo.

### 4.10.50 RepoCommitsComments

`RepoCommit.comments ()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create (comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete ()`

Delete this resource.

`RepoCommitsComments.get (format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get ()`

Fetch all commits from this repo.

### 4.10.51 RepoCommits

`Repo.commits ()`

Return a resource corresponding to all commits in this repo.

### 4.10.52 RepoCommitsComment

`RepoCommits.comment` (*comment\_id*)

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommitsComment.delete` ()

Delete this resource.

`RepoCommitsComment.get` (*format=None*)

Fetch the comment.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.53 RepoCommitsComments

`RepoCommits.comments` ()

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create` (*comment*)

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete` ()

Delete this resource.

`RepoCommitsComments.get` (*format=None, page=None, per\_page=None*)

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommits.compare` (*base, head*)

Fetch the comparison of two commits.

**Variables**

- `base` (*str*) – The commit hash of the first commit.
- `head` (*str*) – The commit hash of the second commit.

`RepoCommits.get` (*sha=None, path=None, page=None, per\_page=None*)

Fetch commits for this repo.

#### Variables

- **sha** (*str*) – Optional commit hash or branch to start listing commits from.
- **path** (*str*) – Optional filter to only include commits that include this file path.

### 4.10.54 RepoContents

`Repo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink(archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

#### Variables

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get(path=None, ref=None)`

This method returns the contents of any file or directory in a repository.

#### Variables

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme(ref=None)`

This method returns the preferred README for a repository.

**Variables** **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.55 Download

`Repo.download(download_id)`

Return a resource corresponding to a single download in this repo.

`Download.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Download.delete()`

Delete this resource.

`Download.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.56 Downloads

`Repo.downloads()`

Return a resource corresponding to all downloads from this repo.

`Downloads.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Downloads.delete()`

Delete this resource.

`Downloads.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.57 Forks

`Repo.forks()`

Return a resource corresponding to all forks of this repo.

`Forks.create()`

Fork this repo.

`Forks.get(sort='newest', page=None, per_page=None)`

Fetch this repo's forks.

**Variables**

- `sort` (*str*) – The sort order for the result.
- `page` (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- `per_page` (*int*) – The amount of results per page.

### 4.10.58 RepoHook

`Repo.hook(hook_id)`

Return a resource corresponding to a single hook in this repo.

`RepoHook.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete()`

Delete this resource.

`RepoHook.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHook.**ping** ()

Send a ping event to the hook.

RepoHook.**test** ()

Trigger the hook with the latest push to the repository.

RepoHook.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.59 RepoHooks

Repo.**hooks** ()

Return a resource corresponding to all hooks of this repo.

RepoHooks.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoHooks.**delete** ()

Delete this resource.

RepoHooks.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHooks.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.60 RepoIssue

Repo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

### 4.10.61 IssueComments

`RepoIssue.comments()`

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to `create`, you don't have to use `{“body”: <comment body>}`.

`IssueComments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComments.delete()`

Delete this resource.

`IssueComments.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (`int`) – For collections, where should paging start. If left as `None`, the first page is returned.
- **per\_page** (`int`) – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`IssueComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.62 IssueEvent

`RepoIssue.event(event_id)`

Return the resource corresponding to a single event of this issue.

`IssueEvent.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (`int`) – For collections, where should paging start. If left as `None`, the first page is returned.
- **per\_page** (`int`) – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

### 4.10.63 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.64 IssueLabel

`RepoIssue.label` (*name*)

Return the resource corresponding to a single label of this issue.

`IssueLabel.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabel.delete` ()

Delete this resource.

`IssueLabel.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.65 IssueLabels

`RepoIssue.labels` ()

Return the resource corresponding to all labels of this issue.

`IssueLabels.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabels.delete` ()

Delete all labels from this issue.

`IssueLabels.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.



IssueLabels.**replace** (*labels*)

Replace all labels on this issue with new ones.

**Variables** *labels* (*list of str*) – A list of labels to use.

IssueLabels.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssue.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssue.**delete** ()

Delete this resource.

RepoIssue.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoIssue.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.66 RepoIssues

Repo.**issues** ()

Return a resource corresponding to all issues from this repo.

## 4.10.67 IssueComment

RepoIssues.**comment** (*comment\_id*)

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to *update*, you don’t have to use *{“body”:* *<comment body>}*.

IssueComment.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComment.**delete** ()

Delete this resource.

IssueComment.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.68 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.get(milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.69 RepoKey

`Repo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKey.delete()`  
Delete this resource.

`RepoKey.get(page=None, per_page=None)`  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKey.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.70 RepoKeys

`Repo.keys()`  
Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`  
Delete this resource.

`RepoKeys.get(page=None, per_page=None)`  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKeys.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.71 RepoLabel

`Repo.label(name)`  
Return a resource corresponding to a single label from this repo.

`RepoLabel.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete()`

Delete this resource.

`RepoLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.72 RepoLabels

`Repo.labels()`

Return a resource corresponding to all issues from this repo.

`RepoLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete()`

Delete this resource.

`RepoLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.73 Milestone

`Repo.milestone(milestone_id)`

Return a resource corresponding to a single milestone in this repo.

#### 4.10.74 MilestoneLabels

`Milestone.labels()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MilestoneLabels.delete()`

Delete this resource.

`MilestoneLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`MilestoneLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestone.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestone.delete()`

Delete this resource.

`Milestone.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Milestone.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.75 Milestones

`Repo.milestones()`

Return a resource corresponding to all milestones in this repo.

Milestones.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestones.**delete** ()

Delete this resource.

Milestones.**get** (*state='open', sort='due\_date', direction='desc', page=None, per\_page=None*)

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.76 PullRequest

Repo.**pullrequest** (*number*)

Return a resource corresponding to a single pull request for this repo.

PullRequest.**commits** ()

Fetch commits on this pull request.

PullRequest.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

PullRequest.**delete** ()

Delete this resource.

PullRequest.**files** ()

Fetch files on this pull request.

PullRequest.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

PullRequest.**is\_merged** ()

Check if this pull request has been merged.

PullRequest.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.77 PullRequests

`Repo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.78 Release

`Repo.release(release_id)`

Return a resource corresponding to a single release in this repo.

### 4.10.79 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`

Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.80 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Release.delete` ()

Delete this resource.

`Release.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.81 Releases

`Repo.releases` ()

Return a resource corresponding to all releases from this repo.

`Releases.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Releases.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Repo.branches` ()

Fetch the branches for this repo.



Repo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** **anon** (*bool*) – Include anonymous contributors.

Repo.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Repo.**delete** ()

Delete this resource.

Repo.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Repo.**languages** ()

Fetch the languages for this repo.

Repo.**tags** ()

Fetch the tags for this repo.

Repo.**teams** ()

Fetch the teams for this repo.

Repo.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.82 Teams

Organizations.**teams** ()

Return a resource corresponding to this org’s teams.

Teams.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Teams.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Organizations.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Organizations.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.83 Repo

`GitHub.repo` (*user, repo*)

Return the resource corresponding to a single repo.

### 4.10.84 RepoCollaborators

`Repo.collaborators` ()

Return a resource corresponding to all collaborators in this repo.

`RepoCollaborators.add` (*user*)

Add a collaborator to this repo.

**Variables** **user** (*str*) – The username of the new collaborator.

`RepoCollaborators.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator` (*user*)

Check if a user is a collaborator in this repo.

**Variables** **user** (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove` (*user*)

Remove a collaborator from this repo.

**Variables** **user** (*str*) – The username of the collaborator.

### 4.10.85 RepoCommit

`Repo.commit` (*sha*)

Return a resource corresponding to a single commit in this repo.

### 4.10.86 RepoCommitsComments

`RepoCommit.comments()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get()`

Fetch all commits from this repo.

### 4.10.87 RepoCommits

`Repo.commits()`

Return a resource corresponding to all commits in this repo.

### 4.10.88 RepoCommitsComment

`RepoCommits.comment(comment_id)`

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.89 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommits.compare(base, head)`

Fetch the comparison of two commits.

**Variables**

- `base` (*str*) – The commit hash of the first commit.
- `head` (*str*) – The commit hash of the second commit.

`RepoCommits.get(sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- `sha` (*str*) – Optional commit hash or branch to start listing commits from.
- `path` (*str*) – Optional filter to only include commits that include this file path.

## 4.10.90 RepoContents

`Repo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink(archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- `archive_format` – Either tarball or zipball.
- `ref` – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get(path=None, ref=None)`

This method returns the contents of any file or directory in a repository.

**Variables**

- `path` (*str*) – Optional content path.

- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

RepoContents.**readme** (*ref=None*)

This method returns the preferred README for a repository.

**Variables** **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

#### 4.10.91 Download

Repo.**download** (*download\_id*)

Return a resource corresponding to a single download in this repo.

Download.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Download.**delete** ()

Delete this resource.

Download.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.92 Downloads

Repo.**downloads** ()

Return a resource corresponding to all downloads from this repo.

Downloads.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Downloads.**delete** ()

Delete this resource.

Downloads.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.93 Forks

`Repo.forks()`  
Return a resource corresponding to all forks of this repo.

`Forks.create()`  
Fork this repo.

`Forks.get(sort='newest', page=None, per_page=None)`  
Fetch this repo's forks.

#### Variables

- **sort** (*str*) – The sort order for the result.
- **page** (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- **per\_page** (*int*) – The amount of results per page.

### 4.10.94 RepoHook

`Repo.hook(hook_id)`  
Return a resource corresponding to a single hook in this repo.

`RepoHook.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete()`  
Delete this resource.

`RepoHook.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoHook.ping()`  
Send a ping event to the hook.

`RepoHook.test()`  
Trigger the hook with the latest push to the repository.

`RepoHook.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.95 RepoHooks

`Repo.hooks()`  
Return a resource corresponding to all hooks of this repo.

RepoHooks.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoHooks.**delete** ()

Delete this resource.

RepoHooks.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHooks.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.96 RepoIssue

Repo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

## 4.10.97 IssueComments

RepoIssue.**comments** ()

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to *create*, you don’t have to use *{“body”: <comment body>}*.

IssueComments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComments.**delete** ()

Delete this resource.

IssueComments.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueComments.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.98 IssueEvent

`RepoIssue.event(event_id)`

Return the resource corresponding to a single event of this issue.

`IssueEvent.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.99 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.100 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.



`IssueLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.101 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabels.delete()`

Delete all labels from this issue.

`IssueLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace(labels)`

Replace all labels on this issue with new ones.

**Variables** `labels` (*list of str*) – A list of labels to use.

`IssueLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.delete()`

Delete this resource.

`RepoIssue.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.102 RepoIssues

`Repo.issues()`

Return a resource corresponding to all issues from this repo.

### 4.10.103 IssueComment

`RepoIssues.comment(comment_id)`

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to `update`, you don't have to use `{"body": <comment body>}`.

`IssueComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComment.delete()`

Delete this resource.

`IssueComment.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.104 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoIssues.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoIssues.**delete** ()

Delete this resource.

RepoIssues.**get** (*milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per\_page=None*)

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

RepoIssues.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.105 RepoKey

Repo.**key** (*key\_id*)

Return a resource corresponding to a single key in this repo.

RepoKey.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoKey.**delete** ()

Delete this resource.

RepoKey.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoKey.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.106 RepoKeys

Repo.**keys** ()

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete ()`  
Delete this resource.

`RepoKeys.get (page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKeys.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.107 RepoLabel

`Repo.label (name)`  
Return a resource corresponding to a single label from this repo.

`RepoLabel.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete ()`  
Delete this resource.

`RepoLabel.get (page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.108 RepoLabels

`Repo.labels ()`  
Return a resource corresponding to all issues from this repo.

`RepoLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete()`

Delete this resource.

`RepoLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.109 Milestone

`Repo.milestone(milestone_id)`

Return a resource corresponding to a single milestone in this repo.

#### 4.10.110 MilestoneLabels

`Milestone.labels()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MilestoneLabels.delete()`

Delete this resource.

`MilestoneLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`MilestoneLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

Milestone.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**delete** ()

Delete this resource.

Milestone.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Milestone.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.111 Milestones

Repo.**milestones** ()

Return a resource corresponding to all milestones in this repo.

Milestones.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestones.**delete** ()

Delete this resource.

Milestones.**get** (*state='open', sort='due\_date', direction='desc', page=None, per\_page=None*)

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.112 PullRequest

Repo.**pullrequest** (*number*)

Return a resource corresponding to a single pull request for this repo.

PullRequest.**commits** ()

Fetch commits on this pull request.

`PullRequest.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequest.delete()`

Delete this resource.

`PullRequest.files()`

Fetch files on this pull request.

`PullRequest.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged()`

Check if this pull request has been merged.

`PullRequest.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.113 PullRequests

`Repo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.114 Release

`Repo.release(release_id)`

Return a resource corresponding to a single release in this repo.

#### 4.10.115 ReleaseAsset

Release.**asset** (*asset\_id*)

ReleaseAsset.**delete** ()  
Delete this resource.

ReleaseAsset.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

ReleaseAsset.**update** (*obj*)  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.116 ReleaseAssets

Release.**assets** ()

ReleaseAssets.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**create** (*obj*)  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Release.**delete** ()  
Delete this resource.

Release.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**update** (*obj*)  
Update this resource.



**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.117 Releases

`Repo.releases ()`

Return a resource corresponding to all releases from this repo.

`Releases.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Releases.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Repo.branches ()`

Fetch the branches for this repo.

`Repo.contributors (anon=False)`

Fetch the contributors from this repo.

**Variables** `anon` (*bool*) – Include anonymous contributors.

`Repo.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Repo.delete ()`

Delete this resource.

`Repo.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Repo.languages ()`

Fetch the languages for this repo.

`Repo.tags ()`

Fetch the tags for this repo.

`Repo.teams ()`

Fetch the teams for this repo.

Repo.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.118 Repos

GitHub.**repos** ()

Return the resource corresponding to all the repos.

Repos.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Repos.**delete** ()

Delete this resource.

Repos.**get** (*type='all', page=None, per\_page=None*)

Fetch repos for this user.

**Variables** **type** – What type of repos to fetch. For details of allowed values, see <http://developer.github.com/v3/repos/#list-user-repositories>.

Repos.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.119 Team

GitHub.**team** (*team*)

Return the resource corresponding to a single team.

#### 4.10.120 TeamMembership

Team.**member** (*user*)

Return a resource corresponding to a single member of a team.

TeamMembership.**add** ()

TeamMembership.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

TeamMembership.**delete** ()

Delete this resource.

TeamMembership.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

TeamMembership.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.121 Members

Team.**members** ()

Return a resource corresponding to a team’s members.

Members.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.122 TeamRepo

Team.**repo** (*user, repo*)

Return a resource corresponding to a single repo to determine if it is managed by this team.

#### 4.10.123 RepoCollaborators

TeamRepo.**collaborators** ()

Return a resource corresponding to all collaborators in this repo.

RepoCollaborators.**add** (*user*)

Add a collaborator to this repo.

**Variables** *user* (*str*) – The username of the new collaborator.

RepoCollaborators.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoCollaborators.**is\_collaborator** (*user*)

Check if a user is a collaborator in this repo.

**Variables** *user* (*str*) – The username to check.

**Returns** bool

`RepoCollaborators.remove(user)`  
Remove a collaborator from this repo.

**Variables** `user` (*str*) – The username of the collaborator.

#### 4.10.124 RepoCommit

`TeamRepo.commit(sha)`  
Return a resource corresponding to a single commit in this repo.

#### 4.10.125 RepoCommitsComments

`RepoCommit.comments()`  
Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`  
Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`  
Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`  
Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get()`  
Fetch all commits from this repo.

#### 4.10.126 RepoCommits

`TeamRepo.commits()`  
Return a resource corresponding to all commits in this repo.

#### 4.10.127 RepoCommitsComment

`RepoCommits.comment(comment_id)`  
Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.128 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommits.compare(base, head)`

Fetch the comparison of two commits.

**Variables**

- **base** (*str*) – The commit hash of the first commit.
- **head** (*str*) – The commit hash of the second commit.

`RepoCommits.get(sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- **sha** (*str*) – Optional commit hash or branch to start listing commits from.
- **path** (*str*) – Optional filter to only include commits that include this file path.

#### 4.10.129 RepoContents

`TeamRepo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archive_link` (*archive\_format*, *ref=None*)

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get` (*path=None*, *ref=None*)

This method returns the contents of any file or directory in a repository.

**Variables**

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme` (*ref=None*)

This method returns the preferred README for a repository.

**Variables** **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.130 Download

`TeamRepo.download` (*download\_id*)

Return a resource corresponding to a single download in this repo.

`Download.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Download.delete` ()

Delete this resource.

`Download.get` (*page=None*, *per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.131 Downloads

`TeamRepo.downloads` ()

Return a resource corresponding to all downloads from this repo.

`Downloads.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Downloads.delete` ()

Delete this resource.

Downloads **.get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.132 Forks

TeamRepo **.forks** ()

Return a resource corresponding to all forks of this repo.

Forks **.create** ()

Fork this repo.

Forks **.get** (*sort='newest', page=None, per\_page=None*)

Fetch this repo's forks.

**Variables**

- **sort** (*str*) – The sort order for the result.
- **page** (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- **per\_page** (*int*) – The amount of results per page.

### 4.10.133 RepoHook

TeamRepo **.hook** (*hook\_id*)

Return a resource corresponding to a single hook in this repo.

RepoHook **.create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoHook **.delete** ()

Delete this resource.

RepoHook **.get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHook **.ping** ()

Send a ping event to the hook.

RepoHook **.test** ()

Trigger the hook with the latest push to the repository.

RepoHook.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.134 RepoHooks

TeamRepo.**hooks** ()

Return a resource corresponding to all hooks of this repo.

RepoHooks.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoHooks.**delete** ()

Delete this resource.

RepoHooks.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHooks.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.135 RepoIssue

TeamRepo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

#### 4.10.136 IssueComments

RepoIssue.**comments** ()

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to *create*, you don't have to use *{“body”: <comment body>}*.

IssueComments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComments.**delete** ()

Delete this resource.



`IssueComments.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComments.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.137 IssueEvent

`RepoIssue.event` (*event\_id*)

Return the resource corresponding to a single event of this issue.

`IssueEvent.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.138 IssueEvents

`RepoIssue.events` ()

Return the resource corresponding to all the events of this issue.

`IssueEvents.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.139 IssueLabel

`RepoIssue.label` (*name*)

Return the resource corresponding to a single label of this issue.

`IssueLabel.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.140 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabels.delete()`

Delete all labels from this issue.

`IssueLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace(labels)`

Replace all labels on this issue with new ones.

**Variables** **labels** (*list of str*) – A list of labels to use.

`IssueLabels.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.delete()`  
Delete this resource.

`RepoIssue.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update(obj)`  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.141 RepoIssues

`TeamRepo.issues()`  
Return a resource corresponding to all issues from this repo.

### 4.10.142 IssueComment

`RepoIssues.comment(comment_id)`  
Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to *update*, you don't have to use *{“body”:* *<comment body>}*.

`IssueComment.create(obj)`  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueComment.delete()`  
Delete this resource.

`IssueComment.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update(obj)`  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.143 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.get(milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.144 RepoKey

`TeamRepo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoKey.delete()`

Delete this resource.

`RepoKey.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKey.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.145 RepoKeys

`TeamRepo.keys()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`

Delete this resource.

`RepoKeys.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKeys.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.146 RepoLabel

`TeamRepo.label(name)`

Return a resource corresponding to a single label from this repo.

`RepoLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete()`

Delete this resource.

`RepoLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.147 RepoLabels

`TeamRepo.labels()`

Return a resource corresponding to all issues from this repo.

`RepoLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete()`

Delete this resource.

`RepoLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.148 Milestone

`TeamRepo.milestone(milestone_id)`

Return a resource corresponding to a single milestone in this repo.

#### 4.10.149 MilestoneLabels

`Milestone.labels()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MilestoneLabels.delete()`

Delete this resource.

`MilestoneLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

MilestoneLabels.**update** (*obj*)

Update this resource.

**Variables obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**create** (*obj*)

Create a new resource.

**Variables obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**delete** ()

Delete this resource.

Milestone.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Milestone.**update** (*obj*)

Update this resource.

**Variables obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.150 Milestones

TeamRepo.**milestones** ()

Return a resource corresponding to all milestones in this repo.

Milestones.**create** (*obj*)

Create a new resource.

**Variables obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestones.**delete** ()

Delete this resource.

Milestones.**get** (*state='open', sort='due\_date', direction='desc', page=None, per\_page=None*)

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.151 PullRequest

`TeamRepo.pullrequest(number)`

Return a resource corresponding to a single pull request for this repo.

`PullRequest.commits()`

Fetch commits on this pull request.

`PullRequest.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequest.delete()`

Delete this resource.

`PullRequest.files()`

Fetch files on this pull request.

`PullRequest.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged()`

Check if this pull request has been merged.

`PullRequest.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.152 PullRequests

`TeamRepo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)



`PullRequests.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.153 Release

`TeamRepo.release(release_id)`  
Return a resource corresponding to a single release in this repo.

#### 4.10.154 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`  
Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.155 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Release.delete()`  
Delete this resource.

Release.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.156 Releases

TeamRepo.**releases** ()

Return a resource corresponding to all releases from this repo.

Releases.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Releases.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

TeamRepo.**add** ()

TeamRepo.**branches** ()

Fetch the branches for this repo.

TeamRepo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** **anon** (*bool*) – Include anonymous contributors.

TeamRepo.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

TeamRepo.**delete** ()

Delete this resource.

TeamRepo.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

TeamRepo.**languages** ()  
Fetch the languages for this repo.

TeamRepo.**tags** ()  
Fetch the tags for this repo.

TeamRepo.**teams** ()  
Fetch the teams for this repo.

TeamRepo.**update** (*obj*)  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.157 TeamRepos

Team.**repos** ()  
Return a resource corresponding to the repos managed by this team.

TeamRepos.**create** (*obj*)  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

TeamRepos.**delete** ()  
Delete this resource.

TeamRepos.**get** (*type='all', page=None, per\_page=None*)  
Fetch repos for this user.

**Variables** **type** – What type of repos to fetch. For details of allowed values, see <http://developer.github.com/v3/repos/#list-user-repositories>.

TeamRepos.**update** (*obj*)  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Team.**create** (*obj*)  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Team.**delete** ()  
Delete this resource.

Team.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Team.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.158 User, CurrentUser

GitHub.**user** (*name=None*)

Return the resource corresponding to a single user. If *name* is *None* the returned resource is the currently authenticated user, otherwise it is the user with the given name.

#### 4.10.159 Organizations

User.**orgs** ()

Return a resource corresponding to user's organizations.

#### 4.10.160 Member

Organizations.**member** (*user*)

Return a resource corresponding to a member of this org.

Member.**delete** ()

Delete this resource.

Member.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.161 Members

Organizations.**members** ()

Return a resource corresponding to members of this org.

Members.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.162 PublicMember

`Organizations.public_member (user)`

Return a resource corresponding to a public member of this org.

`PublicMember.delete ()`

Delete this resource.

`PublicMember.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PublicMember.publicize ()`

### 4.10.163 PublicMembers

`Organizations.public_members ()`

Return a resource corresponding to public members of this org.

`PublicMembers.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.164 OrganizationRepo

`Organizations.repo (repo)`

Return a resource corresponding to single repo for this org.

### 4.10.165 RepoCollaborators

`OrganizationRepo.collaborators ()`

Return a resource corresponding to all collaborators in this repo.

`RepoCollaborators.add (user)`

Add a collaborator to this repo.

**Variables** `user` (*str*) – The username of the new collaborator.

`RepoCollaborators.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator (user)`

Check if a user is a collaborator in this repo.

**Variables** *user* (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove (user)`

Remove a collaborator from this repo.

**Variables** *user* (*str*) – The username of the collaborator.

#### 4.10.166 RepoCommit

`OrganizationRepo.commit (sha)`

Return a resource corresponding to a single commit in this repo.

#### 4.10.167 RepoCommitsComments

`RepoCommit.comments ()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create (comment)`

Create a comment on this commit.

**Variables** *comment* (*str*) – The comment body.

`RepoCommitsComments.delete ()`

Delete this resource.

`RepoCommitsComments.get (format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** *format* – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update (obj)`

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get ()`

Fetch all commits from this repo.

#### 4.10.168 RepoCommits

`OrganizationRepo.commits ()`

Return a resource corresponding to all commits in this repo.

#### 4.10.169 RepoCommitsComment

`RepoCommits.comment (comment_id)`

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables** `format` – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.170 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (`str`) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommits.compare(base, head)`

Fetch the comparison of two commits.

**Variables**

- `base` (`str`) – The commit hash of the first commit.
- `head` (`str`) – The commit hash of the second commit.

`RepoCommits.get(sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- `sha` (`str`) – Optional commit hash or branch to start listing commits from.
- `path` (`str`) – Optional filter to only include commits that include this file path.

### 4.10.171 RepoContents

`OrganizationRepo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink(archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get(path=None, ref=None)`

This method returns the contents of any file or directory in a repository.

**Variables**

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme(ref=None)`

This method returns the preferred README for a repository.

**Variables** **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.172 Download

`OrganizationRepo.download(download_id)`

Return a resource corresponding to a single download in this repo.

`Download.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Download.delete()`

Delete this resource.

`Download.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.173 Downloads

`OrganizationRepo.downloads()`

Return a resource corresponding to all downloads from this repo.

`Downloads.create(obj)`

Create a new resource.



**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Downloads.delete()`  
Delete this resource.

`Downloads.get(page=None, per_page=None)`  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

### 4.10.174 Forks

`OrganizationRepo.forks()`  
Return a resource corresponding to all forks of this repo.

`Forks.create()`  
Fork this repo.

`Forks.get(sort='newest', page=None, per_page=None)`  
Fetch this repo’s forks.

**Variables**

- `sort (str)` – The sort order for the result.
- `page (int)` – The starting page of the result. If left as `None`, the first page is returned.
- `per_page (int)` – The amount of results per page.

### 4.10.175 RepoHook

`OrganizationRepo.hook(hook_id)`  
Return a resource corresponding to a single hook in this repo.

`RepoHook.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete()`  
Delete this resource.

`RepoHook.get(page=None, per_page=None)`  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoHook.ping()`

Send a ping event to the hook.

`RepoHook.test()`

Trigger the hook with the latest push to the repository.

`RepoHook.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.176 RepoHooks

`OrganizationRepo.hooks()`

Return a resource corresponding to all hooks of this repo.

`RepoHooks.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHooks.delete()`

Delete this resource.

`RepoHooks.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoHooks.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.177 RepoIssue

`OrganizationRepo.issue(issue_id)`

Return a resource corresponding to a single issue from this repo.

#### 4.10.178 IssueComments

`RepoIssue.comments()`

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to `create`, you don't have to use `{"body": <comment body>}`.

`IssueComments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComments.delete()`

Delete this resource.

`IssueComments.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.179 IssueEvent

`RepoIssue.event(event_id)`

Return the resource corresponding to a single event of this issue.

`IssueEvent.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.180 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.181 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.182 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabels.delete()`

Delete all labels from this issue.

`IssueLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace(labels)`

Replace all labels on this issue with new ones.

**Variables** `labels` (*list of str*) – A list of labels to use.

`IssueLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.delete()`

Delete this resource.

`RepoIssue.get(page=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.183 RepoIssues

`OrganizationRepo.issues()`

Return a resource corresponding to all issues from this repo.

### 4.10.184 IssueComment

`RepoIssues.comment(comment_id)`

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to `update`, you don’t have to use `{“body”:` `<comment body>}`.

`IssueComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComment.delete()`

Delete this resource.

`IssueComment.get(page=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.185 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.get(milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.186 RepoKey

`OrganizationRepo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKey.delete()`

Delete this resource.

`RepoKey.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKey.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.187 RepoKeys

`OrganizationRepo.keys()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`

Delete this resource.

`RepoKeys.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKeys.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.188 RepoLabel

`OrganizationRepo.label(name)`

Return a resource corresponding to a single label from this repo.

`RepoLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete()`

Delete this resource.

`RepoLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.189 RepoLabels

`OrganizationRepo.labels()`

Return a resource corresponding to all issues from this repo.

`RepoLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete()`

Delete this resource.

`RepoLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.190 Milestone

`OrganizationRepo.milestone(milestone_id)`

Return a resource corresponding to a single milestone in this repo.

#### 4.10.191 MilestoneLabels

`Milestone.labels()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.



`MilestoneLabels.delete()`

Delete this resource.

`MilestoneLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`MilestoneLabels.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Milestone.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Milestone.delete()`

Delete this resource.

`Milestone.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Milestone.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.192 Milestones

`OrganizationRepo.milestones()`

Return a resource corresponding to all milestones in this repo.

`Milestones.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Milestones.delete()`

Delete this resource.

`Milestones.get(state='open', sort='due_date', direction='desc', page=None, per_page=None)`

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.193 PullRequest

OrganizationRepo.**pullrequest** (*number*)

Return a resource corresponding to a single pull request for this repo.

PullRequest.**commits** ()

Fetch commits on this pull request.

PullRequest.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

PullRequest.**delete** ()

Delete this resource.

PullRequest.**files** ()

Fetch files on this pull request.

PullRequest.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

PullRequest.**is\_merged** ()

Check if this pull request has been merged.

PullRequest.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.194 PullRequests

OrganizationRepo.**pullrequests** ()

Return a resource corresponding to all the pull requests for this repo.

PullRequests.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`PullRequests.delete()`  
Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`  
Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state `state`: open or closed (default is open)

`PullRequests.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.195 Release

`OrganizationRepo.release(release_id)`  
Return a resource corresponding to a single release in this repo.

#### 4.10.196 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`  
Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.197 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Release.**delete** ()

Delete this resource.

Release.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.198 Releases

OrganizationRepo.**releases** ()

Return a resource corresponding to all releases from this repo.

Releases.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Releases.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

OrganizationRepo.**branches** ()

Fetch the branches for this repo.

OrganizationRepo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** *anon* (*bool*) – Include anonymous contributors.

OrganizationRepo.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

OrganizationRepo.**delete**()

Delete this resource.

OrganizationRepo.**get**(*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

OrganizationRepo.**languages**()

Fetch the languages for this repo.

OrganizationRepo.**tags**()

Fetch the tags for this repo.

OrganizationRepo.**teams**()

Fetch the teams for this repo.

OrganizationRepo.**update**(*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.199 Repo

Organizations.**repos**()

Return a resource corresponding to repos for this org.

#### 4.10.200 RepoCollaborators

Repo.**collaborators**()

Return a resource corresponding to all collaborators in this repo.

RepoCollaborators.**add**(*user*)

Add a collaborator to this repo.

**Variables** *user* (*str*) – The username of the new collaborator.

RepoCollaborators.**get**(*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoCollaborators.**is\_collaborator**(*user*)

Check if a user is a collaborator in this repo.

**Variables** *user* (*str*) – The username to check.

**Returns** bool

`RepoCollaborators.remove` (*user*)  
Remove a collaborator from this repo.

**Variables** *user* (*str*) – The username of the collaborator.

#### 4.10.201 RepoCommit

`Repo.commit` (*sha*)  
Return a resource corresponding to a single commit in this repo.

#### 4.10.202 RepoCommitsComments

`RepoCommit.comments` ()  
Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create` (*comment*)  
Create a comment on this commit.

**Variables** *comment* (*str*) – The comment body.

`RepoCommitsComments.delete` ()  
Delete this resource.

`RepoCommitsComments.get` (*format=None, page=None, per\_page=None*)  
Fetch all comments for this commit.

**Variables** *format* – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update` (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get` ()  
Fetch all commits from this repo.

#### 4.10.203 RepoCommits

`Repo.commits` ()  
Return a resource corresponding to all commits in this repo.

#### 4.10.204 RepoCommitsComment

`RepoCommits.comment` (*comment\_id*)  
Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create` (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommitsComment.delete` ()  
Delete this resource.

`RepoCommitsComment.get (format=None)`

Fetch the comment.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.205 RepoCommitsComments

`RepoCommits.comments ()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create (comment)`

Create a comment on this commit.

**Variables** `comment (str)` – The comment body.

`RepoCommitsComments.delete ()`

Delete this resource.

`RepoCommitsComments.get (format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommits.compare (base, head)`

Fetch the comparison of two commits.

**Variables**

- `base (str)` – The commit hash of the first commit.
- `head (str)` – The commit hash of the second commit.

`RepoCommits.get (sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- `sha (str)` – Optional commit hash or branch to start listing commits from.
- `path (str)` – Optional filter to only include commits that include this file path.

## 4.10.206 RepoContents

`Repo.contents ()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink (archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

#### Variables

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get` (*path=None, ref=None*)

This method returns the contents of any file or directory in a repository.

#### Variables

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme` (*ref=None*)

This method returns the preferred README for a repository.

**Variables** **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.207 Download

`Repo.download` (*download\_id*)

Return a resource corresponding to a single download in this repo.

`Download.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Download.delete` ()

Delete this resource.

`Download.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.208 Downloads

`Repo.downloads` ()

Return a resource corresponding to all downloads from this repo.

`Downloads.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Downloads.delete` ()

Delete this resource.

`Downloads.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.



**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

**4.10.209 Forks**

Repo.**forks** ()

Return a resource corresponding to all forks of this repo.

Forks.**create** ()

Fork this repo.

Forks.**get** (*sort='newest', page=None, per\_page=None*)

Fetch this repo's forks.

**Variables**

- **sort** (*str*) – The sort order for the result.
- **page** (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- **per\_page** (*int*) – The amount of results per page.

**4.10.210 RepoHook**

Repo.**hook** (*hook\_id*)

Return a resource corresponding to a single hook in this repo.

RepoHook.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoHook.**delete** ()

Delete this resource.

RepoHook.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoHook.**ping** ()

Send a ping event to the hook.

RepoHook.**test** ()

Trigger the hook with the latest push to the repository.

RepoHook.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.211 RepoHooks

`Repo.hooks()`

Return a resource corresponding to all hooks of this repo.

`RepoHooks.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHooks.delete()`

Delete this resource.

`RepoHooks.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoHooks.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.212 RepoIssue

`Repo.issue(issue_id)`

Return a resource corresponding to a single issue from this repo.

### 4.10.213 IssueComments

`RepoIssue.comments()`

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to `create`, you don't have to use `{"body": <comment body>}`.

`IssueComments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComments.delete()`

Delete this resource.

`IssueComments.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.214 IssueEvent

`RepoIssue.event(event_id)`

Return the resource corresponding to a single event of this issue.

`IssueEvent.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.215 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.216 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.217 IssueLabels

`RepoIssue.labels` ()

Return the resource corresponding to all labels of this issue.

`IssueLabels.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabels.delete` ()

Delete all labels from this issue.

`IssueLabels.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace` (*labels*)

Replace all labels on this issue with new ones.

**Variables** **labels** (*list of str*) – A list of labels to use.

`IssueLabels.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.delete` ()

Delete this resource.

`RepoIssue.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.218 RepoIssues

`Repo.issues` ()

Return a resource corresponding to all issues from this repo.

### 4.10.219 IssueComment

`RepoIssues.comment` (*comment\_id*)

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to *update*, you don't have to use *{“body”:<comment body>}*.

`IssueComment.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueComment.delete` ()

Delete this resource.

`IssueComment.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.220 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.get(milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.221 RepoKey

`Repo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoKey.delete()`

Delete this resource.

`RepoKey.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKey.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.222 RepoKeys

`Repo.keys()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`

Delete this resource.

`RepoKeys.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKeys.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.223 RepoLabel

`Repo.label(name)`

Return a resource corresponding to a single label from this repo.

`RepoLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabel.delete()`

Delete this resource.

`RepoLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.224 RepoLabels

`Repo.labels()`

Return a resource corresponding to all issues from this repo.

`RepoLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoLabels.delete()`

Delete this resource.

`RepoLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.225 Milestone

`Repo.milestone(milestone_id)`

Return a resource corresponding to a single milestone in this repo.

#### 4.10.226 MilestoneLabels

`Milestone.labels()`

Return the resource corresponding to the labels of this milestone.

`MilestoneLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MilestoneLabels.delete()`

Delete this resource.

`MilestoneLabels.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.



### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

MilestoneLabels.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**delete** ()

Delete this resource.

Milestone.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Milestone.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.227 Milestones

Repo.**milestones** ()

Return a resource corresponding to all milestones in this repo.

Milestones.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestones.**delete** ()

Delete this resource.

Milestones.**get** (*state='open', sort='due\_date', direction='desc', page=None, per\_page=None*)

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.228 PullRequest

`Repo.pullrequest (number)`

Return a resource corresponding to a single pull request for this repo.

`PullRequest.commits ()`

Fetch commits on this pull request.

`PullRequest.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequest.delete ()`

Delete this resource.

`PullRequest.files ()`

Fetch files on this pull request.

`PullRequest.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged ()`

Check if this pull request has been merged.

`PullRequest.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.229 PullRequests

`Repo.pullrequests ()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete ()`

Delete this resource.

`PullRequests.get (state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.230 Release

`Repo.release(release_id)`  
Return a resource corresponding to a single release in this repo.

#### 4.10.231 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`  
Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.232 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Release.delete()`  
Delete this resource.

Release.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Release.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.233 Releases

Repo.**releases** ()

Return a resource corresponding to all releases from this repo.

Releases.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Releases.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Repo.**branches** ()

Fetch the branches for this repo.

Repo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** **anon** (*bool*) – Include anonymous contributors.

Repo.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Repo.**delete** ()

Delete this resource.

Repo.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Repo.**languages** ()  
Fetch the languages for this repo.

Repo.**tags** ()  
Fetch the tags for this repo.

Repo.**teams** ()  
Fetch the teams for this repo.

Repo.**update** (*obj*)  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.234 Teams

Organizations.**teams** ()  
Return a resource corresponding to this org's teams.

Teams.**create** (*obj*)  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Teams.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Organizations.**get** (*page=None, per\_page=None*)  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Organizations.**update** (*obj*)  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.235 UserRepos

User.**repos** (*page=None, per\_page=None*)  
Return the resource corresponding to all the repos of this user.

`UserRepos.delete()`  
Delete this resource.

`UserRepos.get(type='all', page=None, per_page=None)`  
Fetch repos for this user.

**Variables** `type` (*str*) – What type of repos to fetch. For details of allowed values, see <http://developer.github.com/v3/repos/#list-user-repositories>.

`UserRepos.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`User.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`User.followers(page=None, per_page=None)`  
Fetch the followers of this user.

`User.following(page=None, per_page=None)`  
Fetch users that this user is following.

`User.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.236 UserEmails

`CurrentUser.emails()`  
Return the resource corresponding to the emails of the authenticated user.

`UserEmails.add(emails)`  
Add emails to the authenticated user.

**Variables** `emails` (*list of str*) – A list of emails to add.

`UserEmails.get()`  
Fetch all emails of the authenticated user.

`UserEmails.remove(emails)`  
Remove emails from the authenticated user.

**Variables** `emails` (*list of str*) – A list of emails to remove.

### 4.10.237 UserMemberships

`CurrentUser.memberships()`  
Return the resource corresponding to the org memberships of the authenticated user.

### 4.10.238 UserMembership

`UserMemberships.org (org)`

Return the resource corresponding to the current user's membership of the specified organization.

`UserMembership.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`UserMembership.delete ()`

Delete this resource.

`UserMembership.get ()`

`UserMembership.update (obj)`

`UserMemberships.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`UserMemberships.delete ()`

Delete this resource.

`UserMemberships.get (state=None)`

List your organization memberships.

**Variables** `state (str)` – Specify whether only active or pending memberships are returned. If left as `None`, all memberships are returned.

`UserMemberships.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.239 Organizations

`CurrentUser.orgs ()`

Return a resource corresponding to user's organizations.

### 4.10.240 Member

`Organizations.member (user)`

Return a resource corresponding to a member of this org.

`Member.delete ()`

Delete this resource.

`Member.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page (int)** – For collections, where should paging start. If left as `None`, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.241 Members

`Organizations.members()`

Return a resource corresponding to members of this org.

`Members.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.242 PublicMember

`Organizations.public_member(user)`

Return a resource corresponding to a public member of this org.

`PublicMember.delete()`

Delete this resource.

`PublicMember.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PublicMember.publicize()`

#### 4.10.243 PublicMembers

`Organizations.public_members()`

Return a resource corresponding to public members of this org.

`PublicMembers.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.



#### 4.10.244 OrganizationRepo

`Organizations.repo(repo)`

Return a resource corresponding to single repo for this org.

#### 4.10.245 RepoCollaborators

`OrganizationRepo.collaborators()`

Return a resource corresponding to all collaborators in this repo.

`RepoCollaborators.add(user)`

Add a collaborator to this repo.

**Variables** `user` (*str*) – The username of the new collaborator.

`RepoCollaborators.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator(user)`

Check if a user is a collaborator in this repo.

**Variables** `user` (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove(user)`

Remove a collaborator from this repo.

**Variables** `user` (*str*) – The username of the collaborator.

#### 4.10.246 RepoCommit

`OrganizationRepo.commit(sha)`

Return a resource corresponding to a single commit in this repo.

#### 4.10.247 RepoCommitsComments

`RepoCommit.comments()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables `format`** – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables `obj`** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommit.get()`

Fetch all commits from this repo.

#### 4.10.248 RepoCommits

`OrganizationRepo.commits()`

Return a resource corresponding to all commits in this repo.

#### 4.10.249 RepoCommitsComment

`RepoCommits.comment(comment_id)`

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`

Create a new resource.

**Variables `obj`** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables `format`** – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables `obj`** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.250 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables `comment`** (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables format** – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoCommits.compare(base, head)`

Fetch the comparison of two commits.

**Variables**

- **base** (*str*) – The commit hash of the first commit.
- **head** (*str*) – The commit hash of the second commit.

`RepoCommits.get(sha=None, path=None, page=None, per_page=None)`

Fetch commits for this repo.

**Variables**

- **sha** (*str*) – Optional commit hash or branch to start listing commits from.
- **path** (*str*) – Optional filter to only include commits that include this file path.

### 4.10.251 RepoContents

`OrganizationRepo.contents()`

Return a resource corresponding to repo contents.

`RepoContents.archivelink(archive_format, ref=None)`

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get(path=None, ref=None)`

This method returns the contents of any file or directory in a repository.

**Variables**

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme(ref=None)`

This method returns the preferred README for a repository.

**Variables ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

### 4.10.252 Download

`OrganizationRepo.download(download_id)`

Return a resource corresponding to a single download in this repo.

`Download.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Download.delete()`  
Delete this resource.

`Download.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

### 4.10.253 Downloads

`OrganizationRepo.downloads()`  
Return a resource corresponding to all downloads from this repo.

`Downloads.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Downloads.delete()`  
Delete this resource.

`Downloads.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

### 4.10.254 Forks

`OrganizationRepo.forks()`  
Return a resource corresponding to all forks of this repo.

`Forks.create()`  
Fork this repo.

`Forks.get(sort='newest', page=None, per_page=None)`  
Fetch this repo's forks.

**Variables**

- `sort (str)` – The sort order for the result.
- `page (int)` – The starting page of the result. If left as `None`, the first page is returned.
- `per_page (int)` – The amount of results per page.

### 4.10.255 RepoHook

`OrganizationRepo.hook(hook_id)`

Return a resource corresponding to a single hook in this repo.

`RepoHook.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete()`

Delete this resource.

`RepoHook.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoHook.ping()`

Send a ping event to the hook.

`RepoHook.test()`

Trigger the hook with the latest push to the repository.

`RepoHook.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.256 RepoHooks

`OrganizationRepo.hooks()`

Return a resource corresponding to all hooks of this repo.

`RepoHooks.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHooks.delete()`

Delete this resource.

`RepoHooks.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

RepoHooks.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.257 RepoIssue

OrganizationRepo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

#### 4.10.258 IssueComments

RepoIssue.**comments** ()

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to *create*, you don't have to use *{“body”: <comment body>}*.

IssueComments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComments.**delete** ()

Delete this resource.

IssueComments.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueComments.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.259 IssueEvent

RepoIssue.**event** (*event\_id*)

Return the resource corresponding to a single event of this issue.

IssueEvent.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.260 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

#### 4.10.261 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.262 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueLabels.delete()`  
Delete all labels from this issue.

`IssueLabels.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace(labels)`  
Replace all labels on this issue with new ones.

**Variables** `labels` (*list of str*) – A list of labels to use.

`IssueLabels.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssue.delete()`  
Delete this resource.

`RepoIssue.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.263 RepoIssues

`OrganizationRepo.issues()`  
Return a resource corresponding to all issues from this repo.



### 4.10.264 IssueComment

`RepoIssues.comment (comment_id)`

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to `update`, you don't have to use `{“body”:`  
`<comment body>}`.

`IssueComment.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComment.delete ()`

Delete this resource.

`IssueComment.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.265 IssueEvents

`RepoIssues.events ()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssues.delete ()`

Delete this resource.

`RepoIssues.get (milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.266 RepoKey

`OrganizationRepo.key(key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKey.delete()`

Delete this resource.

`RepoKey.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoKey.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.267 RepoKeys

`OrganizationRepo.keys()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete()`

Delete this resource.

`RepoKeys.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoKeys.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.268 RepoLabel

OrganizationRepo.**label** (*name*)

Return a resource corresponding to a single label from this repo.

RepoLabel.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoLabel.**delete** ()

Delete this resource.

RepoLabel.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoLabel.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.269 RepoLabels

OrganizationRepo.**labels** ()

Return a resource corresponding to all issues from this repo.

RepoLabels.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

RepoLabels.**delete** ()

Delete this resource.

RepoLabels.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

RepoLabels.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.10.270 Milestone

OrganizationRepo.**milestone** (*milestone\_id*)

Return a resource corresponding to a single milestone in this repo.

#### 4.10.271 MilestoneLabels

Milestone.**labels** ()

Return the resource corresponding to the labels of this milestone.

MilestoneLabels.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

MilestoneLabels.**delete** ()

Delete this resource.

MilestoneLabels.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

MilestoneLabels.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**delete** ()

Delete this resource.

Milestone.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Milestone.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.272 Milestones

OrganizationRepo.**milestones** ()

Return a resource corresponding to all milestones in this repo.

Milestones.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestones.**delete** ()

Delete this resource.

Milestones.**get** (*state='open', sort='due\_date', direction='desc', page=None, per\_page=None*)

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

Milestones.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.273 PullRequest

OrganizationRepo.**pullrequest** (*number*)

Return a resource corresponding to a single pull request for this repo.

PullRequest.**commits** ()

Fetch commits on this pull request.

PullRequest.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

PullRequest.**delete** ()

Delete this resource.

PullRequest.**files** ()

Fetch files on this pull request.

PullRequest.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged()`

Check if this pull request has been merged.

`PullRequest.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.274 PullRequests

`OrganizationRepo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.275 Release

`OrganizationRepo.release(release_id)`

Return a resource corresponding to a single release in this repo.

#### 4.10.276 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`

Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.277 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Release.delete()`

Delete this resource.

`Release.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.278 Releases

`OrganizationRepo.releases()`

Return a resource corresponding to all releases from this repo.

`Releases.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

Releases.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

OrganizationRepo.**branches** ()

Fetch the branches for this repo.

OrganizationRepo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** **anon** (*bool*) – Include anonymous contributors.

OrganizationRepo.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

OrganizationRepo.**delete** ()

Delete this resource.

OrganizationRepo.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

OrganizationRepo.**languages** ()

Fetch the languages for this repo.

OrganizationRepo.**tags** ()

Fetch the tags for this repo.

OrganizationRepo.**teams** ()

Fetch the teams for this repo.

OrganizationRepo.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.279 Repo

Organizations.**repos** ()

Return a resource corresponding to repos for this org.



## 4.10.280 RepoCollaborators

`Repo.collaborators()`

Return a resource corresponding to all collaborators in this repo.

`RepoCollaborators.add(user)`

Add a collaborator to this repo.

**Variables** `user` (*str*) – The username of the new collaborator.

`RepoCollaborators.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoCollaborators.is_collaborator(user)`

Check if a user is a collaborator in this repo.

**Variables** `user` (*str*) – The username to check.

**Returns** `bool`

`RepoCollaborators.remove(user)`

Remove a collaborator from this repo.

**Variables** `user` (*str*) – The username of the collaborator.

## 4.10.281 RepoCommit

`Repo.commit(sha)`

Return a resource corresponding to a single commit in this repo.

## 4.10.282 RepoCommitsComments

`RepoCommit.comments()`

Return a resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (*str*) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either *raw*, *text*, *html* or *full*. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommit.get()`

Fetch all commits from this repo.

#### 4.10.283 RepoCommits

`Repo.commits()`

Return a resource corresponding to all commits in this repo.

#### 4.10.284 RepoCommitsComment

`RepoCommits.comment(comment_id)`

Return the resource corresponding to a single comment of this commit.

`RepoCommitsComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommitsComment.delete()`

Delete this resource.

`RepoCommitsComment.get(format=None)`

Fetch the comment.

**Variables** `format` – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.10.285 RepoCommitsComments

`RepoCommits.comments()`

Return the resource corresponding to all comments of this commit.

`RepoCommitsComments.create(comment)`

Create a comment on this commit.

**Variables** `comment` (`str`) – The comment body.

`RepoCommitsComments.delete()`

Delete this resource.

`RepoCommitsComments.get(format=None, page=None, per_page=None)`

Fetch all comments for this commit.

**Variables** `format` – Which format should be requested, either `raw`, `text`, `html` or `full`. For details on formats, see <http://developer.github.com/v3/mime/#comment-body-properties>.

`RepoCommitsComments.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoCommits.compare` (*base*, *head*)

Fetch the comparison of two commits.

**Variables**

- **base** (*str*) – The commit hash of the first commit.
- **head** (*str*) – The commit hash of the second commit.

`RepoCommits.get` (*sha=None*, *path=None*, *page=None*, *per\_page=None*)

Fetch commits for this repo.

**Variables**

- **sha** (*str*) – Optional commit hash or branch to start listing commits from.
- **path** (*str*) – Optional filter to only include commits that include this file path.

## 4.10.286 RepoContents

`Repo.contents` ()

Return a resource corresponding to repo contents.

`RepoContents.archivelink` (*archive\_format*, *ref=None*)

This method will return a URL to download a tarball or zipball archive for a repository.

**Variables**

- **archive\_format** – Either tarball or zipball.
- **ref** – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.get` (*path=None*, *ref=None*)

This method returns the contents of any file or directory in a repository.

**Variables**

- **path** (*str*) – Optional content path.
- **ref** (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

`RepoContents.readme` (*ref=None*)

This method returns the preferred README for a repository.

**Variables** `ref` (*str*) – Optional string name of the commit/branch/tag. Defaults to master.

## 4.10.287 Download

`Repo.download` (*download\_id*)

Return a resource corresponding to a single download in this repo.

`Download.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Download.delete` ()

Delete this resource.

Download.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.288 Downloads

Repo.**downloads** ()

Return a resource corresponding to all downloads from this repo.

Downloads.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Downloads.**delete** ()

Delete this resource.

Downloads.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

## 4.10.289 Forks

Repo.**forks** ()

Return a resource corresponding to all forks of this repo.

Forks.**create** ()

Fork this repo.

Forks.**get** (*sort='newest', page=None, per\_page=None*)

Fetch this repo's forks.

**Variables**

- **sort** (*str*) – The sort order for the result.
- **page** (*int*) – The starting page of the result. If left as *None*, the first page is returned.
- **per\_page** (*int*) – The amount of results per page.

## 4.10.290 RepoHook

Repo.**hook** (*hook\_id*)

Return a resource corresponding to a single hook in this repo.

`RepoHook.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHook.delete()`  
Delete this resource.

`RepoHook.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoHook.ping()`  
Send a ping event to the hook.

`RepoHook.test()`  
Trigger the hook with the latest push to the repository.

`RepoHook.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.291 RepoHooks

`Repo.hooks()`  
Return a resource corresponding to all hooks of this repo.

`RepoHooks.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoHooks.delete()`  
Delete this resource.

`RepoHooks.get(page=None, per_page=None)`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page(int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page(int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoHooks.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.10.292 RepoIssue

Repo.**issue** (*issue\_id*)

Return a resource corresponding to a single issue from this repo.

## 4.10.293 IssueComments

RepoIssue.**comments** ()

Return the resource corresponding to the comments of this issue.

When creating comments, use a simple string as the parameter to *create*, you don't have to use `{"body": <comment body>}`.

IssueComments.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

IssueComments.**delete** ()

Delete this resource.

IssueComments.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

IssueComments.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.294 IssueEvent

RepoIssue.**event** (*event\_id*)

Return the resource corresponding to a single event of this issue.

IssueEvent.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.295 IssueEvents

`RepoIssue.events()`

Return the resource corresponding to all the events of this issue.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

### 4.10.296 IssueLabel

`RepoIssue.label(name)`

Return the resource corresponding to a single label of this issue.

`IssueLabel.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabel.delete()`

Delete this resource.

`IssueLabel.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabel.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.297 IssueLabels

`RepoIssue.labels()`

Return the resource corresponding to all labels of this issue.

`IssueLabels.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`IssueLabels.delete()`

Delete all labels from this issue.

`IssueLabels.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueLabels.replace` (*labels*)

Replace all labels on this issue with new ones.

**Variables** *labels* (*list of str*) – A list of labels to use.

`IssueLabels.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoIssue.delete` ()

Delete this resource.

`RepoIssue.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssue.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.298 RepoIssues

`Repo.issues` ()

Return a resource corresponding to all issues from this repo.

## 4.10.299 IssueComment

`RepoIssues.comment` (*comment\_id*)

Return the resource corresponding to a single comment of an issue.

When updating comments, use a simple string as the parameter to *update*, you don't have to use *{“body”:* *<comment body>}*.



`IssueComment.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IssueComment.delete()`

Delete this resource.

`IssueComment.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`IssueComment.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.300 IssueEvents

`RepoIssues.events()`

Return the resource corresponding to all events of this repo's issues.

`IssueEvents.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoIssues.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoIssues.delete()`

Delete this resource.

`RepoIssues.get(milestone=None, state='open', assignee=None, mentioned=None, labels=None, sort='created', direction='desc', since=None, page=None, per_page=None)`

Fetch issues for this repository based on the filter parameters and using the specified format.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/#list-issues-for-a-repository>

`RepoIssues.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.301 RepoKey

`Repo.key (key_id)`

Return a resource corresponding to a single key in this repo.

`RepoKey.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKey.delete ()`

Delete this resource.

`RepoKey.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKey.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.302 RepoKeys

`Repo.keys ()`

Return a resource corresponding to all SSH keys of this repo.

`RepoKeys.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`RepoKeys.delete ()`

Delete this resource.

`RepoKeys.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page (int)` – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page (int)` – For collections, how many objects could be returned. The maximum is 100. If left as `None`, 30 objects are returned.

`RepoKeys.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.303 RepoLabel

`Repo.label` (*name*)

Return a resource corresponding to a single label from this repo.

`RepoLabel.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoLabel.delete` ()

Delete this resource.

`RepoLabel.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabel.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.304 RepoLabels

`Repo.labels` ()

Return a resource corresponding to all issues from this repo.

`RepoLabels.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`RepoLabels.delete` ()

Delete this resource.

`RepoLabels.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`RepoLabels.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.305 Milestone

Repo.**milestone** (*milestone\_id*)

Return a resource corresponding to a single milestone in this repo.

### 4.10.306 MilestoneLabels

Milestone.**labels** ()

Return the resource corresponding to the labels of this milestone.

MilestoneLabels.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

MilestoneLabels.**delete** ()

Delete this resource.

MilestoneLabels.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

MilestoneLabels.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Milestone.**delete** ()

Delete this resource.

Milestone.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Milestone.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.307 Milestones

`Repo.milestones()`

Return a resource corresponding to all milestones in this repo.

`Milestones.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Milestones.delete()`

Delete this resource.

`Milestones.get(state='open', sort='due_date', direction='desc', page=None, per_page=None)`

Fetch milestones for this repository, based on the filter parameters.

For details on the meanings and allowed values for each parameter, see <http://developer.github.com/v3/issues/milestones/#list-milestones-for-a-repository>.

`Milestones.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.308 PullRequest

`Repo.pullrequest(number)`

Return a resource corresponding to a single pull request for this repo.

`PullRequest.commits()`

Fetch commits on this pull request.

`PullRequest.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequest.delete()`

Delete this resource.

`PullRequest.files()`

Fetch files on this pull request.

`PullRequest.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`PullRequest.is_merged()`

Check if this pull request has been merged.

`PullRequest.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.309 PullRequests

`Repo.pullrequests()`

Return a resource corresponding to all the pull requests for this repo.

`PullRequests.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PullRequests.delete()`

Delete this resource.

`PullRequests.get(state=None, page=None, per_page=None)`

Fetch pull requests.

**Variables** `state` – Optional filter pull requests by state state: open or closed (default is open)

`PullRequests.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.310 Release

`Repo.release(release_id)`

Return a resource corresponding to a single release in this repo.

### 4.10.311 ReleaseAsset

`Release.asset(asset_id)`

`ReleaseAsset.delete()`

Delete this resource.

`ReleaseAsset.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`ReleaseAsset.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.10.312 ReleaseAssets

`Release.assets()`

`ReleaseAssets.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Release.delete()`

Delete this resource.

`Release.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Release.update(obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.313 Releases

`Repo.releases()`

Return a resource corresponding to all releases from this repo.

`Releases.create(obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Releases.get(page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Repo.**branches** ()

Fetch the branches for this repo.

Repo.**contributors** (*anon=False*)

Fetch the contributors from this repo.

**Variables** **anon** (*bool*) – Include anonymous contributors.

Repo.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Repo.**delete** ()

Delete this resource.

Repo.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

Repo.**languages** ()

Fetch the languages for this repo.

Repo.**tags** ()

Fetch the tags for this repo.

Repo.**teams** ()

Fetch the teams for this repo.

Repo.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.10.314 Teams

Organizations.**teams** ()

Return a resource corresponding to this org's teams.

Teams.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Teams.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.



- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Organizations.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`Organizations.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.10.315 Teams

`CurrentUser.teams` ()

Return the resource corresponding to the teams that the current user belongs to.

`Teams.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Teams.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`CurrentUser.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`CurrentUser.follow` (*name*)

Start following the given user.

`CurrentUser.followers` (*page=None, per\_page=None*)

Fetch the followers of this user.

`CurrentUser.following` (*page=None, per\_page=None*)

Fetch users that this user is following.

`CurrentUser.follows` (*name*)

Check if the authenticated user follows the given user.

**Returns** bool

`CurrentUser.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 30 objects are returned.

`CurrentUser.unfollow` (*name*)

Stop following the given user.

`CurrentUser.update` (*obj*)

## 4.11 GoogleAnalytics

**class** `googleanalytics.GoogleAnalytics` (*access\_token=None*)

Create a Google Analytics service.

**Variables** *access\_token* –

### 4.11.1 Management

`GoogleAnalytics.management` ()

Return the resource corresponding to the management API

### 4.11.2 Account

`Management.account` (*account\_id*)

Return the resource corresponding to a single account

### 4.11.3 WebProperties

`Account.webproperties` ()

Return the resource corresponding to all web properties

`WebProperties.get` (*max\_results=None, start\_index=None, userIp=None, quotaUser=None*)

List resource

**Variables**

- **max-results** (*int*) – The maximum number of rows to include in the response
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user's IP address is unknown.

#### 4.11.4 WebProperty

`Account.webproperty` (*webproperty\_id*)  
Return the resource corresponding to a single property

#### 4.11.5 View

`WebProperty.view` (*profile\_id*)  
Return the resource corresponding to a single view

#### 4.11.6 Goal

`View.goal` (*goal\_id*)  
Return the resource corresponding to a single goal

`Goal.get` (*userIp=None, quotaUser=None*)  
Get resource

##### Variables

- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

#### 4.11.7 Goals

`View.goals` ()  
Return the resource corresponding to all goals

`Goals.get` (*max\_results=None, start\_index=None, userIp=None, quotaUser=None*)  
List resource

##### Variables

- **max-results** (*int*) – The maximum number of rows to include in the response
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

`View.get` (*userIp=None, quotaUser=None*)  
Get resource

##### Variables

- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

### 4.11.8 Views

WebProperty.**views** ()

Return the resource corresponding to all views

Views.**get** (*max\_results=None, start\_index=None, userIp=None, quotaUser=None*)

List resource

#### Variables

- **max-results** (*int*) – The maximum number of rows to include in the response
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

WebProperty.**get** (*userIp=None, quotaUser=None*)

Get resource

#### Variables

- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

Account.**get** (*userIp=None, quotaUser=None*)

Get resource

#### Variables

- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

### 4.11.9 Accounts

Management.**accounts** ()

Return the resource corresponding to all accounts

Accounts.**get** (*max\_results=None, start\_index=None, userIp=None, quotaUser=None*)

List resource

#### Variables

- **max-results** (*int*) – The maximum number of rows to include in the response
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

### 4.11.10 Segments

Management.**segments** ()

Return the resource corresponding to all segments

Segments.**get** (*max\_results=None, start\_index=None, userIp=None, quotaUser=None*)

List resource

#### Variables

- **max-results** (*int*) – The maximum number of rows to include in the response
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

### 4.11.11 Reporting

GoogleAnalytics.**reporting** ()

Return the resource corresponding to the reporting API

Reporting.**core** (*ids, start\_date, end\_date, metrics, dimensions=None, sort=None, filters=None, segment=None, start\_index=None, max\_results=None, fields=None, prettyPrint=None, userIp=None, quotaUser=None, access\_token=None, key=None*)

Query the Core Reporting API for Google Analytics report data.

#### Variables

- **ids** (*str*) – The unique table ID of the form ga:XXXX, where XXXX is the Analytics view (profile) ID for which the query will retrieve the data.
- **start-date** (*str*) – The first date of the date range for which you are requesting the data.
- **end-date** (*str*) – The first last of the date range for which you are requesting the data.
- **metrics** (*str*) – A list of comma-separated metrics, such as ga:visits,ga:bounces.
- **dimensions** (*str*) – A list of comma-separated dimensions for your Analytics data, such as ga:browser,ga:city.

**:var sort** A list of comma-separated dimensions and metrics indicating the sorting order and sorting direction for the returned data.

#### Variables

- **filters** (*str*) – Dimension or metric filters that restrict the data returned for your request.
- **segment** (*str*) – Segments the data returned for your request.
- **start-index** (*int*) – The first row of data to retrieve, starting at 1. Use this parameter as a pagination mechanism along with the max-results parameter.
- **max-results** (*int*) – The maximum number of rows to include in the response
- **fields** – Selector specifying a subset of fields to include in the response.

- **prettyPrint** (*bool*) – Returns response with indentations and line breaks. Default false.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.
- **access\_token** (*str*) – One possible way to provide an OAuth 2.0 token.
- **key** (*str*) – Used for OAuth 1.0a authorization to specify your application to get quota. For example: key=AldefliuhSFADSfasdfasdfASdf.

Reporting.**realtime** (*ids, metrics, dimensions=None, sort=None, filters=None, max\_results=None, fields=None, prettyPrint=None, userIp=None, quotaUser=None, access\_token=None, key=None*)

Returns real-time data for a view (profile)

**Variables**

- **ids** (*str*) – The unique table ID of the form ga:XXXX, where XXXX is the Analytics view (profile) ID for which the query will retrieve the data.
- **metrics** (*str*) – A list of comma-separated metrics, such as ga:visits,ga:bounces.
- **dimensions** (*str*) – A list of comma-separated dimensions for your Analytics data, such as ga:browser,ga:city.

**:var sort** A list of comma-separated dimensions and metrics indicating the sorting order and sorting direction for the returned data.

**Variables**

- **filters** (*str*) – Dimension or metric filters that restrict the data returned for your request.
- **max-results** (*int*) – The maximum number of rows to include in the response
- **fields** – Selector specifying a subset of fields to include in the response.
- **prettyPrint** (*bool*) – Returns response with indentations and line breaks. Default false.
- **userIp** (*str*) – Specifies IP address of the end user for whom the API call is being made. Used to cap usage per IP.
- **quotaUser** (*str*) – Alternative to userIp in cases when the user’s IP address is unknown.

## 4.12 GoogleCalendar

**class** googlecalendar.**GoogleCalendar** (*access\_token=None*)  
 Create a Google Calendar service.

**Variables** *access\_token* –

### 4.12.1 Calendar

GoogleCalendar.**calendar** (*calendar\_id*)  
 Return the resource corresponding to a single calendar

## 4.12.2 Event

Calendar.**event** (*event\_id*)

Return the resource corresponding to a single event

Event.**delete** (*sendNotifications=None*)

Delete this resource.

**Variables** **sendNotifications** (*str*) – Whether to send notifications. The default is False.

Event.**get** (*alwaysIncludeEmail=None, maxAttendees=None, timeZone=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### Variables

- **alwaysIncludeEmail** (*str*) – Whether to always include a value in the “email” field for the organizer, creator and attendees, even if no real email is available. The default is False.
- **maxAttendees** (*int*) – The maximum number of attendees to include in the response. If none is indicated, only the participant is returned.
- **timeZone** (*str*) – Time zone used in the response. The default is the time zone of the calendar.

Event.**instances** (*alwaysIncludeEmail=None, maxAttendees=None, maxResults=None, originalStart=None, pageToken=None, showDeleted=None, timeZone=None*)

Fetch all instances of the recurring event.

### Variables

- **alwaysIncludeEmail** (*str*) – Whether to always include a value in the “email” field for the organizer, creator and attendees, even if no real email is available. The default is False.
- **maxAttendees** (*int*) – The maximum number of attendees to include in the response. If none is indicated, only the participant is returned.
- **maxResults** (*int*) – Maximum number of instances returned.
- **originalStart** (*str*) – The original start time of the instance in the result.
- **pageToken** (*str*) – Token specifying which result page to return.
- **showDeleted** (*str*) – Whether to include deleted instances. The default is False.
- **timeZone** (*str*) – Time zone used in the response. The default is the time zone of the calendar.

Event.**move** (*destination, sendNotifications=None*)

Move an event to another calendar.

### Variables

- **destination** (*str*) – Calendar identifier of the target calendar where the event is to be moved to.
- **sendNotifications** (*str*) – Whether to send notifications. The default is False.

Event.**patch** (*obj, alwaysIncludeEmail=None, sendNotifications=None*)

Patch this resource.

### Variables

- **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

- **alwaysIncludeEmail** (*str*) – Whether to always include a value in the “email” field for the organizer, creator and attendees, even if no real email is available. The default is False.
- **sendNotifications** (*str*) – Whether to send notifications. The default is False.

`Event.update` (*obj*, *alwaysIncludeEmail=None*, *sendNotifications=None*)

Update this resource.

#### Variables

- **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.
- **alwaysIncludeEmail** (*str*) – Whether to always include a value in the “email” field for the organizer, creator and attendees, even if no real email is available. The default is False.
- **sendNotifications** (*str*) – Whether to send notifications. The default is False.

### 4.12.3 Events

`Calendar.events` ()

Return the resource corresponding to all the events

`Events.create` (*obj*, *sendNotifications=None*)

Create a new resource.

#### Variables

- **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.
- **sendNotifications** (*str*) – Whether to send notifications. The default is False.

`Events.get` (*alwaysIncludeEmail=None*, *iCalUID=None*, *maxAttendees=None*, *maxResults=None*, *orderBy=None*, *pageToken=None*, *q=None*, *showDeleted=None*, *showHiddenInvitations=None*, *singleEvents=None*, *timeMax=None*, *timeMin=None*, *timeZone=None*, *updateMin=None*)

Fetch all events on the calendar.

#### Variables

- **alwaysIncludeEmail** (*str*) – Whether to always include a value in the “email” field for the organizer, creator and attendees, even if no real email is available. The default is False.
- **iCalUID** (*str*) – Specifies iCalendar UID of events to be included.
- **maxAttendees** (*int*) – The maximum number of attendees to include in the response. If none is indicated, only the participant is returned.
- **maxResults** (*int*) – Maximum number of events returned.
- **orderBy** (*str*) – The order of the events returned in the result. The default is an unspecified, stable order.
- **pageToken** (*str*) – Token specifying which result page to return.
- **q** (*str*) – Free text search terms to find events that match these terms.
- **showDeleted** (*str*) – Whether to include deleted events. The default is False.
- **showHiddenInvitations** (*str*) – Whether to include hidden invitations. The default is False.



- **singleEvents** (*str*) – Whether to expand recurring events into instances and only return single one-off events and instances of recurring events, but not the underlying recurring events themselves. The default is False.

. :var **timeMax**: Upper bound for an event’s start time to filter by. The default is not to filter by start time.

#### Variables

- **timeMin** (*str*) – Lower bound for an event’s end time to filter by. The default is not to filter by end time.
- **timeZone** (*str*) – Time zone used in the response. The default is the time zone of the calendar.
- **updatedMin** – Lower bound for an event’s last modification timestamp to filter by. Optional. The default is not to filter by last modification time.

Events.**importing** (*obj*)

Import an event.

**Variables** *obj* – a Python object representing the imported event.

Events.**quick\_add** (*text*, *sendNotifications=None*)

Import an event.

#### Variables

- **text** (*str*) – The text describing the event to be created.
- **sendNotifications** (*str*) – Whether to send notifications. The default is False.

### 4.12.4 Acl

Calendar.**rule** (*rule\_id*)

Return the resource corresponding to a single rule

Acl.**delete** ()

Delete this resource.

Acl.**get** ()

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

Acl.**patch** (*obj*)

Update this resource’s metadata.

**Variables** *obj* – a Python object representing the updated resource. Refer to the upstream documentation for details.

Acl.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.12.5 Acls

Calendar.**rules** ()

Return the resource corresponding to all the rules

`AcIs.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`AcIs.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Calendar.clear()`

Clear this calendar.

`Calendar.delete()`

Delete this resource.

`Calendar.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Calendar.patch(obj)`

Update this resource's metadata.

**Variables** `obj` – a Python object representing the updated resource. Refer to the upstream documentation for details.

`Calendar.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.12.6 Calendars

`GoogleCalendar.calendars()`

Return the resource corresponding to all the calendars

`Calendars.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.12.7 ColorsResource

`GoogleCalendar.colors()`

Return the resource corresponding to all the colors

`ColorsResource.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.12.8 FreeBusyResource

`GoogleCalendar.freebusy()`

Return the resource corresponding to all the free/busy info

`FreeBusyResource.query(obj)`

Return free/busy info for a set of calendars.

**Variables** `obj` – a Python object representing the query.

### 4.12.9 User

`GoogleCalendar.me()`  
Return the resource corresponding to the current user

### 4.12.10 CalendarList

`User.calendar_list(calendar_id)`  
Return the resource corresponding to a single calendar list

`CalendarList.delete()`  
Delete this resource.

`CalendarList.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`CalendarList.patch(obj)`  
Update this resource's metadata.

**Variables** `obj` – a Python object representing the updated resource. Refer to the upstream documentation for details.

`CalendarList.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.12.11 CalendarLists

`User.calendar_lists()`  
Return the resource corresponding to all the calendar lists

`CalendarLists.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CalendarLists.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.12.12 SettingResource

`User.setting(setting)`  
Return the resource corresponding to a single setting

`SettingResource.get()`  
For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.12.13 SettingResource

`User.settings()`  
Return the resource corresponding to all the settings

`SettingResource.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.13 GoogleOAuth2

**class** `googleoauth2.GoogleOAuth2` (*client\_id, client\_secret*)

Create a Google Analytics service.

### Variables

- **client\_id** – The `client_id` obtained from the APIs Console. Indicates the client that is making the request. The value passed in this parameter must exactly match the value shown in the APIs Console.
- **client\_id** – str

### 4.13.1 Service methods

`GoogleOAuth2.access_token` (*code, redirect\_uri*)

Get the access and/or refresh token

### Variables

- **code** (*str*) – The authorization code returned from the initial request
- **redirect\_uri** (*str*) – The URI registered with the application

`GoogleOAuth2.refresh_token` (*refresh\_token*)

Refresh the access token

**Variables** *refresh\_token* – The refresh token returned from the authorization code exchange

## 4.14 GoogleSpreadsheets

**class** `googlespreadsheets.GoogleSpreadsheets` (*access\_token=None*)

Create a Google Spreadsheets service.

**Variables** *access\_token* –

### 4.14.1 Spreadsheet

`GoogleSpreadsheets.spreadsheet` (*key*)

Return the resource corresponding to a single spreadsheet

### 4.14.2 Worksheet

`Spreadsheet.worksheet` (*worksheet\_id, visibility, projection*)

Return the resource corresponding to a single worksheet

### 4.14.3 Cell

Worksheet.**cell** (*cell\_id*)

Return the resource corresponding to a single cell

Cell.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Cell.**update** (*version*, *obj*)

Update this resource.

#### Variables

- **version** (*str*) – the resource version you want to update.
- **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.14.4 Cells

Worksheet.**cells** ()

Return the resource corresponding to all the cells

Cells.**get** (*min\_row=None*, *max\_row=None*, *min\_col=None*, *max\_col=None*)

Fetch cells for the worksheet.

#### Variables

- **min\_row** (*int*) – To get cells above the indicated row.
- **max\_row** (*int*) – To get cells below the given row.
- **min\_col** (*int*) – To get cells from the indicated column.
- **max\_col** (*int*) – To get cells to the given column.

### 4.14.5 Row

Worksheet.**row** (*row\_id*)

Return the resource corresponding to a single row

Row.**delete** (*version*)

Delete this resource.

**Variables** **version** (*str*) – the resource version you want to delete.

Row.**update** (*version*, *obj*)

Update this resource.

#### Variables

- **version** (*str*) – the resource version you want to update.
- **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.14.6 Rows

Worksheet.**rows** ()

Return the resource corresponding to all the rows

Rows.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

Rows.**get** (*reverse=None, orderby=None, sq=None*)

Fetch rows for the worksheet.

**Variables**

- **reverse** (*bool*) – To get rows in reverse order
- **orderby** (*str*) – To sort the values in ascending order by a particular column.
- **sq** (*str*) – Use it to produce a feed with entries that meet the specified criteria.

Worksheet.**delete** (*version*)

Delete this resource.

**Variables** *version* (*str*) – the resource version you want to delete.

Worksheet.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Worksheet.**update** (*version, obj*)

Update this resource.

**Variables**

- **version** (*str*) – the resource version you want to update.
- **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.14.7 Worksheets

Spreadsheet.**worksheets** (*visibility, projection*)

Return the resource corresponding to all the worksheets

Worksheets.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

Worksheets.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.14.8 Spreadsheets

GoogleSpreadsheets.**spreadsheets** ()

Return the resource corresponding to all the spreadsheets

Spreadsheets.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.15 Instagram

**class** `instagram.Instagram` (*client\_id=None, access\_token=None*)

Create an Instagram service.

### Variables

- **client\_id** (*str*) – Associates your script with a specific application. Required if no `access_token`.
- **access\_token** (*str*) – For some requests, specifically those made on behalf of a user, authentication is needed. Required if no `client_id`.

### 4.15.1 AuthenticatedUser

`Instagram.authenticated_user()`

Return the resource corresponding to the authenticated user.

### 4.15.2 Feed

`AuthenticatedUser.feed()`

Return the resource corresponding to all entries for the user.

`Feed.get` (*count=None, min\_id=None, max\_id=None*)

Fetch all of the objects.

### Variables

- **count** (*int*) – Count of media to return.
- **min\_id** (*int*) – Return media later than this `min_id`.
- **max\_id** (*int*) – Return media earlier than this `max_id`.

### 4.15.3 LikedMedia

`AuthenticatedUser.liked_media()`

Return the resource corresponding to all liked media for the user.

`LikedMedia.get` (*count=None, max\_liked\_id=None*)

Fetch all of the objects.

### Variables

- **count** (*int*) – Count of media to return.
- **max\_like\_id** (*int*) – Return media liked before this id.

### 4.15.4 RequestedBy

`AuthenticatedUser.requested_by()`

Return the resource corresponding to all requests for the user.

`RequestedBy.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`AuthenticatedUser.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.15.5 Geography

`Instagram.geography(geo_id)`

Return the resource corresponding to a single geography

### 4.15.6 RecentMedia

`Geography.recent_media()`

Return the resource corresponding to all recent media for the geography.

`RecentMedia.get(count=None, max_timestamp=None, min_timestamp=None, min_id=None, max_id=None)`

Fetch all of the objects.

#### Variables

- `count` (*int*) – Count of media to return.
- `max_timestamp` (*int*) – Return media before this UNIX timestamp.
- `min_timestamp` (*int*) – Return media after this UNIX timestamp.
- `min_id` (*int*) – Return media later than this `min_id`.
- `max_id` (*int*) – Return media earlier than this `max_id`.

### 4.15.7 Location

`Instagram.location(location_id)`

Return the resource corresponding to a single location.

### 4.15.8 RecentMedia

`Location.recent_media()`

Return the resource corresponding to all recent media for the location.

`RecentMedia.get(count=None, max_timestamp=None, min_timestamp=None, min_id=None, max_id=None)`

Fetch all of the objects.

#### Variables

- `count` (*int*) – Count of media to return.
- `max_timestamp` (*int*) – Return media before this UNIX timestamp.
- `min_timestamp` (*int*) – Return media after this UNIX timestamp.
- `min_id` (*int*) – Return media later than this `min_id`.
- `max_id` (*int*) – Return media earlier than this `max_id`.

`Location.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.



### 4.15.9 Locations

`Instagram.locations()`

Return the resource corresponding to all locations.

`Locations.get(lat=None, distance=None, lng=None, foursquare_v2_id=None, foursquare_id=None)`

fetch all locations by geographic coordinate.

#### Variables

- **lat** (*float*) – Latitude of the center search coordinate. If used, lng is required.
- **distance** (*int*) – Default is 1km (distance=1000), max distance is 5km.
- **lng** (*float*) – Longitude of the center search coordinate. If used, lat is required.
- **foursquare\_v2\_id** (*str*) – A foursquare v2 api location id. If used, you are not required to use lat and lng.
- **foursquare\_id** (*str*) – A foursquare v1 api location id. If used, you are not required to use lat and lng. Note that this method is deprecated; you should use the new foursquare IDs with V2 of their API.

### 4.15.10 Media

`Instagram.media(media_id)`

Return the resource corresponding to a single media.

### 4.15.11 Comment

`Media.comment(comment_id)`

Return the resource corresponding to a single comment for the media.

`Comment.delete()`

Delete this resource.

### 4.15.12 Comments

`Media.comments()`

Return the resource corresponding to all comments for the media.

`Comments.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.15.13 Likes

`Media.likes()`

Return the resource corresponding to all likes for the media.

Likes.**create**()

Set a like on this media by the currently authenticated user.

Likes.**delete**()

Remove a like on this media by the currently authenticated user.

Likes.**get**()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Media.**get**()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.15.14 Medias

Instagram.**medias**()

Return the resource corresponding to all medias.

Medias.**get**(*lat=None, max\_timestamp=None, min\_timestamp=None, lng=None, distance=None*)

Fetch all of the objects.

##### Variables

- **lat** (*float*) – Latitude of the center search coordinate. If used, lng is required.
- **max\_timestamp** (*int*) – A unix timestamp. All media returned will be taken later than this timestamp.
- **min\_timestamp** (*int*) – A unix timestamp. All media returned will be taken earlier than this timestamp.
- **lng** (*float*) – Longitude of the center search coordinate. If used, lat is required.
- **distance** (*int*) – Default is 1km (distance=1000), max distance is 5km.

#### 4.15.15 PopularMedia

Instagram.**popular\_media**()

Return the resource corresponding to all most popular media.

PopularMedia.**get**()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.15.16 Tag

Instagram.**tag**(*tag\_name*)

Return the resource corresponding to a single tag.

#### 4.15.17 RecentMedia

Tag.**recent\_media**()

Return the resource corresponding to all recent media for the tag.

RecentMedia.**get**(*count=None, max\_timestamp=None, min\_timestamp=None, min\_id=None, max\_id=None*)

Fetch all of the objects.

##### Variables

- `count` (*int*) – Count of media to return.
- `max_timestamp` (*int*) – Return media before this UNIX timestamp.
- `min_timestamp` (*int*) – Return media after this UNIX timestamp.
- `min_id` (*int*) – Return media later than this `min_id`.
- `max_id` (*int*) – Return media earlier than this `max_id`.

`Tag.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.15.18 Tags

`Instagram.tags()`

Return the resource corresponding to all tags.

`Tags.get(query)`

fetch all tags by name.

**Variables** `query` (*str*) – A valid tag name without a leading #. (eg. snow, nofilter).

#### 4.15.19 User

`Instagram.user(user_id)`

Return the resource corresponding to a single user.

#### 4.15.20 FollowedBy

`User.followed_by()`

Return the resource corresponding to all followers for the user.

`FollowedBy.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.15.21 Follows

`User.follows()`

Return the resource corresponding to all follows for the user.

`Follows.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.15.22 RecentMedia

`User.recent_media()`

Return the resource corresponding to all recent media for the user.

`RecentMedia.get(count=None, max_timestamp=None, min_timestamp=None, min_id=None, max_id=None)`

Fetch all of the objects.

**Variables**

- `count` (*int*) – Count of media to return.

- `max_timestamp` (*int*) – Return media before this UNIX timestamp.
- `min_timestamp` (*int*) – Return media after this UNIX timestamp.
- `min_id` (*int*) – Return media later than this `min_id`.
- `max_id` (*int*) – Return media earlier than this `max_id`.

### 4.15.23 Relationship

`User.relationship` ()

Return the resource corresponding to all relationships for the user.

`Relationship.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Relationship.update` (*action*)

Modifies the relationship between the current user and the target user.

**Variables** `action` (*str*) – One of follow/unfollow/block/unblock/approve/deny.

`User.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.15.24 Users

`Instagram.users` ()

Return the resource corresponding to all users.

`Users.get` (*query*, *count=None*)

Fetch all users by name.

**Variables**

- `query` (*str*) – A query string.
- `count` (*int*) – Number of users to return.

## 4.16 Intercom

`class intercom.Intercom` (*app\_id*, *api\_key*)

Create a Intercom service.

**Variables**

- `app_id` (*str*) – The APP identifier.
- `api_key` (*str*) – The API key.

### 4.16.1 Companies

`Intercom.companies` ()

Return the resource corresponding to all companies.

`Companies.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Companies.get` (*page=None, per\_page=None*)

Fetch all of the objects.

**Variables**

- **page** (*int*) – The page that should be returned. If left as *None*, first page are returned.
- **per\_page** (*int*) – How many objects should be returned. The maximum is 500. If left as *None*, 500 objects are returned.

## 4.16.2 Company

`Intercom.company` (*id*)

Return the resource corresponding to a single company.

`Company.get` ()

Fetch the company’s data.

`Company.users` ()

Fetch the company’s users.

## 4.16.3 Counts

`Intercom.counts` ()

Return the resource corresponding to all counts.

`Counts.get` (*type=None, count=None*)

Get counts of users and companies filtered by certain criteria.

**Variables**

- **type** (*str*) – The count’s type
- **count** (*str*) – The count’s filter criteria

## 4.16.4 Events

`Intercom.events` ()

Return the resource corresponding to all events.

`Events.create` (*event\_name, created\_at, user\_id=None, email=None, metadata=None*)

Create a new Event object.

**Variables**

- **event\_name** (*str*) – The name of the event that occurred.
- **created\_at** (*int*) – The time the event occurred as a UTC Unix timestamp.
- **user\_id** (*int*) – The `user_id` of the user which messages should be returned. Required if no email.
- **email** (*str*) – The email of the user which messages that should be returned. Required if no `user_id`.
- **metadata** (*dict*) – Optional metadata about the event.

### 4.16.5 Impressions

`Intercom.impressions()`

Return the resource corresponding to all impressions.

`Impressions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.16.6 MessageThread

`Intercom.message_thread()`

Return the resource corresponding to a single message thread.

`MessageThread.get(thread_id, user_id=None, email=None)`

Fetch all a single object.

**Variables**

- `thread_id(int)` – The `thread_id` of the message that should be returned.
- `user_id(int)` – The `user_id` of the user which message should be returned. Required if no email.
- `email(str)` – The email of the user which message that should be returned. Required if no `user_id`.

### 4.16.7 MessageThreads

`Intercom.message_threads()`

Return the resource corresponding to all message threads.

`MessageThreads.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MessageThreads.get(user_id=None, email=None)`

Fetch all of the objects for the user.

**Variables**

- `user_id(int)` – The `user_id` of the user which messages should be returned. Required if no email.
- `email(str)` – The email of the user which messages that should be returned. Required if no `user_id`.

`MessageThreads.reply(obj)`

Reply to a message thread from an admin from a user

### 4.16.8 User

`Intercom.user()`

Return the resource corresponding to a single user.

`User.get (user_id=None, email=None)`  
Fetch the object's data.

**Variables**

- **user\_id** (*int*) – The user\_id of the user that should be returned. Required if no email.
- **email** (*str*) – The email of the user that should be returned. Required if no user\_id.

## 4.16.9 Users

`Intercom.users ()`  
Return the resource corresponding to all users.

`Users.create (obj)`  
Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Users.get (page=None, per_page=None)`  
Fetch all of the objects.

**Variables**

- **page** (*int*) – The page that should be returned. If left as *None*, first page are returned.
- **per\_page** (*int*) – How many objects should be returned. The maximum is 500. If left as *None*, 500 objects are returned.

`Users.update (obj)`  
Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.17 Mailchimp

`class mailchimp.Mailchimp (api_key)`  
Create a Mailchimp service.

**Variables** **api\_key** (*str*) – The API key including the region, for instance *8ac789caf98879caf897a678fa76daf-us2*.

### 4.17.1 Service methods

`Mailchimp.apikeyAdd (username, password)`  
Call Mailchimp's *apikeyAdd* method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/apikeyadd.func.php>

`Mailchimp.apikeyExpire (username, password)`  
Call Mailchimp's *apikeyExpire* method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/apikeyexpire.func.php>

Mailchimp.**apikey** (*username, password, apikey, expired=False*)

Call Mailchimp's apikeys method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/apikeys.func.php>

Mailchimp.**campaignAbuseReports** (*cid, since=None, start=None, limit=None*)

Call Mailchimp's campaignAbuseReports method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignabusereports.func.php>

Mailchimp.**campaignAdvice** (*cid*)

Call Mailchimp's campaignAdvice method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignadvice.func.php>

Mailchimp.**campaignAnalytics** (*cid*)

Call Mailchimp's campaignAnalytics method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignanalytics.func.php>

Mailchimp.**campaignBounceMessage** (*cid, email*)

Call Mailchimp's campaignBounceMessage method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignbouncemessage.func.php>

Mailchimp.**campaignBounceMessages** (*cid, start=None, limit=None, since=None*)

Call Mailchimp's campaignBounceMessages method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignbouncemessages.func.php>

Mailchimp.**campaignClickDetailAIM** (*cid, url, start=None, limit=None*)

Call Mailchimp's campaignClickDetailAIM method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignclickdetailaim.func.php>

Mailchimp.**campaignClickStats** (*cid*)

Call Mailchimp's campaignClickStats method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignclickstats.func.php>

Mailchimp.**campaignContent** (*cid, for\_archive=True*)

Call Mailchimp's campaignContent method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigncontent.func.php>

Mailchimp.**campaignCreate** (*type, options, content, segment\_opts={}, type\_opts={}*)

Call Mailchimp's campaignCreate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigncreate.func.php>

Mailchimp.**campaignDelete** (*cid*)

Call Mailchimp's campaignDelete method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigndelete.func.php>

Mailchimp.**campaignEcommOrderAdd** (*order*)

Call Mailchimp's campaignEcommOrderAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignecommorderadd.func.php>

Mailchimp.**campaignEcommOrders** (*cid, start=None, limit=None, since=None*)

Call Mailchimp's campaignEcommOrders method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignecommorders.func.php>



Mailchimp.**campaignEepUrlStats** (*cid*)

Call Mailchimp's campaignEepUrlStats method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigneepurlstats.func.php>

Mailchimp.**campaignEmailStatsAIM** (*cid*, *email\_address=[]*)

Call Mailchimp's campaignEmailStatsAIM method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignemailstatsaim.func.php>

Mailchimp.**campaignGeoOpens** (*cid*)

Call Mailchimp's campaignGeoOpens method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigngeoopens.func.php>

Mailchimp.**campaignGeoOpensForCountry** (*cid*, *code*)

Call Mailchimp's campaignGeoOpensForCountry method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigngeoopensforcountry.func.php>

Mailchimp.**campaignMembers** (*cid*, *status=None*, *start=None*, *limit=None*)

Call Mailchimp's campaignMembers method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignmembers.func.php>

Mailchimp.**campaignNotOpenedAIM** (*cid*, *start=None*, *limit=None*)

Call Mailchimp's campaignNotOpenedAIM method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignnotopenedaim.func.php>

Mailchimp.**campaignOpenedAIM** (*cid*, *start=None*, *limit=None*)

Call Mailchimp's campaignOpenedAIM method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignopenedaim.func.php>

Mailchimp.**campaignPause** (*cid*)

Call Mailchimp's campaignPause method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignpause.func.php>

Mailchimp.**campaignReplicate** (*cid*)

Call Mailchimp's campaignReplicate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignreplicate.func.php>

Mailchimp.**campaignResume** (*cid*)

Call Mailchimp's campaignResume method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignresume.func.php>

Mailchimp.**campaignSchedule** (*cid*, *schedule\_time*, *schedule\_time\_b=None*)

Call Mailchimp's campaignSchedule method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignschedule.func.php>

Mailchimp.**campaignSegmentTest** (*list\_id*, *options={}*)

Call Mailchimp's campaignSegmentTest method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignsegmenttest.func.php>

Mailchimp.**campaignSendNow** (*cid*)

Call Mailchimp's campaignSendNow method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignsendnow.func.php>

Mailchimp.**campaignSendTest** (*cid*, *test\_emails=[]*, *send\_type=None*)

Call Mailchimp's campaignSendTest method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignsendtest.func.php>

Mailchimp.**campaignShareReport** (*cid*, *opts={}*)

Call Mailchimp's campaignShareReport method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignsharereport.func.php>

Mailchimp.**campaignStats** (*cid*)

Call Mailchimp's campaignStats method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignstats.func.php>

Mailchimp.**campaignTemplateContent** (*cid*)

Call Mailchimp's campaignTemplateContent method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigntemplatecontent.func.php>

Mailchimp.**campaignUnschedule** (*cid*)

Call Mailchimp's campaignUnschedule method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignunschedule.func.php>

Mailchimp.**campaignUnsubscribes** (*cid*, *start=None*, *limit=None*)

Call Mailchimp's campaignUnsubscribes method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignunsubscribes.func.php>

Mailchimp.**campaignUpdate** (*cid*, *name*, *value*)

Call Mailchimp's campaignUpdate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignupdate.func.php>

Mailchimp.**campaigns** (*filters={}*, *start=None*, *limit=None*)

Call Mailchimp's campaigns method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaigns.func.php>

Mailchimp.**campaignsForEmail** (*email\_address*, *options={}*)

Call Mailchimp's campaignsForEmail method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/campaignsforemail.func.php>

Mailchimp.**chimpChatter** ()

Call Mailchimp's chimpChatter method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/chimpchatter.func.php>

Mailchimp.**ecommOrderAdd** (*order={}*)

Call Mailchimp's ecommOrderAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/ecommorderadd.func.php>

Mailchimp.**ecommOrderDel** (*store\_id*, *order\_id*)

Call Mailchimp's ecommOrderDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/ecommorderdel.func.php>

Mailchimp.**ecommOrders** (*start=None*, *limit=None*, *since=None*)

Call Mailchimp's ecommOrders method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/ecommorders.func.php>

Mailchimp.**folderAdd** (*name*, *type=None*)

Call Mailchimp's folderAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/folderadd.func.php>

Mailchimp.**folderDel** (*fid*, *type=None*)

Call Mailchimp's folderDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/folderdel.func.php>

Mailchimp.**folderUpdate** (*fid*, *name*, *type=None*)

Call Mailchimp's folderUpdate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/folderupdate.func.php>

Mailchimp.**folders** (*type=None*)

Call Mailchimp's folders method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/folders.func.php>

Mailchimp.**generateText** (*type*, *content*)

Call Mailchimp's generateText method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/generatetext.func.php>

Mailchimp.**getAccountDetails** ()

Call Mailchimp's getAccountDetails method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/getaccountdetails.func.php>

Mailchimp.**gmonkeyActivity** ()

Call Mailchimp's gmonkeyActivity method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/gmonkeyactivity.func.php>

Mailchimp.**gmonkeyAdd** (*cid*, *id*, *email\_address=[]*)

Call Mailchimp's gmonkeyAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/gmonkeyadd.func.php>

Mailchimp.**gmonkeyDel** (*cid*, *id*, *email\_address=[]*)

Call Mailchimp's gmonkeyDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/gmonkeydel.func.php>

Mailchimp.**gmonkeyMembers** ()

Call Mailchimp's gmonkeyMembers method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/gmonkeymembers.func.php>

Mailchimp.**inlineCss** (*html*, *strip\_css=False*)

Call Mailchimp's inlineCss method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/inlinecss.func.php>

Mailchimp.**listAbuseReports** (*id*, *start=None*, *limit=None*, *since=None*)

Call Mailchimp's listAbuseReports method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listabusereports.func.php>

Mailchimp.**listActivity** (*id*)

Call Mailchimp's listActivity method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listactivity.func.php>

`Mailchimp.listBatchSubscribe` (*id*, *batch=[]*, *double\_optin=True*, *update\_existing=False*, *replace\_interests=True*)

Call Mailchimp's listBatchSubscribe method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listbatchsubscribe.func.php>

`Mailchimp.listBatchUnsubscribe` (*id*, *emails=[]*, *delete\_member=False*, *send\_goodbye=True*, *send\_notify=False*)

Call Mailchimp's listBatchUnsubscribe method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listbatchunsubscribe.func.php>

`Mailchimp.listClients` (*id*)

Call Mailchimp's listClients method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listclients.func.php>

`Mailchimp.listGrowthHistory` (*id*)

Call Mailchimp's listGrowthHistory method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listgrowthhistory.func.php>

`Mailchimp.listInterestGroupAdd` (*id*, *group\_name*, *grouping\_id=None*)

Call Mailchimp's listInterestGroupAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupadd.func.php>

`Mailchimp.listInterestGroupDel` (*id*, *group\_name*, *grouping\_id=None*)

Call Mailchimp's listInterestGroupDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupdel.func.php>

`Mailchimp.listInterestGroupUpdate` (*id*, *old\_name*, *new\_name*, *grouping\_id=None*)

Call Mailchimp's listInterestGroupUpdate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupupdate.func.php>

`Mailchimp.listInterestGroupingAdd` (*id*, *name*, *type*, *groups=[]*)

Call Mailchimp's listInterestGroupingAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupingadd.func.php>

`Mailchimp.listInterestGroupingDel` (*grouping\_id*)

Call Mailchimp's listInterestGroupingDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupingdel.func.php>

`Mailchimp.listInterestGroupingUpdate` (*grouping\_id*, *name*, *value*)

Call Mailchimp's listInterestGroupingUpdate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupingupdate.func.php>

`Mailchimp.listInterestGroupings` (*id*)

Call Mailchimp's listInterestGroupings method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listinterestgroupings.func.php>

`Mailchimp.listLocations` (*id*)

Call Mailchimp's listLocations method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listlocations.func.php>

`Mailchimp.listMemberActivity` (*id*, *email\_address=[]*)

Call Mailchimp's listMemberActivity method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmemberactivity.func.php>

`Mailchimp.listMemberInfo` (*id*, *email\_address*=[])

Call Mailchimp's `listMemberInfo` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmemberinfo.func.php>

`Mailchimp.listMembers` (*id*, *status*='subscribed', *since*=None, *start*=None, *limit*=None)

Call Mailchimp's `listMembers` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmembers.func.php>

`Mailchimp.listMergeVarAdd` (*id*, *tag*, *name*, *options*={})

Call Mailchimp's `listMergeVarAdd` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmergevaradd.func.php>

`Mailchimp.listMergeVarDel` (*id*, *tag*)

Call Mailchimp's `listMergeVarDel` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmergevardel.func.php>

`Mailchimp.listMergeVarUpdate` (*id*, *tag*, *options*={})

Call Mailchimp's `listMergeVarUpdate` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmergevarupdate.func.php>

`Mailchimp.listMergeVars` (*id*)

Call Mailchimp's `listMergeVars` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listmergevars.func.php>

`Mailchimp.listStaticSegmentAdd` (*id*, *name*)

Call Mailchimp's `listStaticSegmentAdd` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegmentadd.func.php>

`Mailchimp.listStaticSegmentDel` (*id*, *seg\_id*)

Call Mailchimp's `listStaticSegmentDel` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegmentdel.func.php>

`Mailchimp.listStaticSegmentMembersAdd` (*id*, *seg\_id*, *batch*=[])

Call Mailchimp's `listStaticSegmentMembersAdd` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegmentmembersadd.func.php>

`Mailchimp.listStaticSegmentMembersDel` (*id*, *seg\_id*, *batch*=[])

Call Mailchimp's `listStaticSegmentMembersDel` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegmentmembersdel.func.php>

`Mailchimp.listStaticSegmentReset` (*id*, *seg\_id*)

Call Mailchimp's `listStaticSegmentReset` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegmentreset.func.php>

`Mailchimp.listStaticSegments` (*id*)

Call Mailchimp's `listStaticSegments` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/liststaticsegments.func.php>

`Mailchimp.listSubscribe` (*id*, *email\_address*, *merge\_vars*={}, *email\_type*='html', *double\_optin*=True, *update\_existing*=False, *replace\_interests*=True, *send\_welcome*=False)

Call Mailchimp's `listSubscribe` method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listssubscribe.func.php>

`Mailchimp.listUnsubscribe` (*id*, *email\_address*, *delete\_member=False*, *send\_goodbye=True*,  
*send\_notify=True*)

Call Mailchimp's listUnsubscribe method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listunsubscribe.func.php>

`Mailchimp.listUpdateMember` (*id*, *email\_address*, *merge\_vars={}*, *email\_type=None*, *re-*  
*place\_interests=True*)

Call Mailchimp's listUpdateMember method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listupdatemember.func.php>

`Mailchimp.listWebhookAdd` (*id*, *url*, *actions={}*, *sources={}*)

Call Mailchimp's listWebhookAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listwebhookadd.func.php>

`Mailchimp.listWebhookDel` (*id*, *url*)

Call Mailchimp's listWebhookDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listwebhookdel.func.php>

`Mailchimp.listWebhooks` (*id*)

Call Mailchimp's listWebhooks method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listwebhooks.func.php>

`Mailchimp.lists` (*filters={}*, *start=None*, *limit=None*)

Call Mailchimp's lists method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/lists.func.php>

`Mailchimp.listsForEmail` (*email\_address*)

Call Mailchimp's listsForEmail method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/listsforemail.func.php>

`Mailchimp.ping` ()

Call Mailchimp's ping method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/ping.func.php>

`Mailchimp.templateAdd` (*name*, *html*)

Call Mailchimp's templateAdd method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templateadd.func.php>

`Mailchimp.templateDel` (*id*)

Call Mailchimp's templateDel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templatedel.func.php>

`Mailchimp.templateInfo` (*tid*, *type='user'*)

Call Mailchimp's templateInfo method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templateinfo.func.php>

`Mailchimp.templateUndel` (*id*)

Call Mailchimp's templateUndel method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templateundel.func.php>

`Mailchimp.templateUpdate` (*id*, *values={}*)

Call Mailchimp's templateUpdate method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templateupdate.func.php>

`Mailchimp.templates` (*types=None, category=None, inactives=None*)

Call Mailchimp's templates method.

Upstream documentation: <http://apidocs.mailchimp.com/api/rtfm/templates.func.php>

## 4.18 Mixpanel

**class** `mixpanel.Mixpanel` (*token=None, api\_key=None, api\_secret=None*)

Create a Mixpanel service.

### Variables

- **token** (*str or None*) – Optional token used for tracking events. If you leave this as `None`, you won't be able to track events through this service object.
- **api\_key** (*str or None*) – Optional API key. If you leave this as `None`, you won't be able to export data through this service object.
- **api\_secret** (*str or None*) – Optional API secret. If you leave this as `None`, you won't be able to export data through this service object.

### 4.18.1 Events

`Mixpanel.events` ()

Return the resource corresponding to events.

`Events.get` (*event, type, unit, interval*)

Fetch event data.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#events-default>

`Events.names` (*type, limit=None*)

Fetch the most common events over the last 31 days.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#events-names>

`Events.top` (*type, limit=None*)

Fetch the top events for today.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#events-top>

### 4.18.2 Funnels

`Mixpanel.funnels` ()

Return the resource corresponding to funnels.

`Funnels.get` (*funnel\_id, from\_date=None, to\_date=None, length=None, interval=None, unit=None, on=None, where=None, limit=None*)

Fetch data for a funnel.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#funnels-default>

`Funnels.list` ()

Fetch the list of all funnels.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#funnels-list>

### 4.18.3 Properties

`Mixpanel.properties()`

Return the resource corresponding to events properties.

`Properties.get(event, name, type, unit, interval, values=None, limit=None)`

Fetch data of a single event.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#event-properties-default>

`Properties.top(event, limit=None)`

Fetch top property names for an event.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#event-properties-top>

`Properties.values(event, name, limit=None, bucket=None)`

Fetch top values for a property.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#event-properties-values>

### 4.18.4 Retention

`Mixpanel.retention()`

Return the resource corresponding to retention (cohort analysis).

`Retention.get(from_date, to_date, retention_type=None, born_event=None, event=None, born_where=None, where=None, interval=None, interval_count=None, unit=None, on=None, limit=None)`

Fetch cohort analysis.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#retention-default>

### 4.18.5 Segmentation

`Mixpanel.segmentation()`

Return the resource corresponding to segmentation.

`Segmentation.average(event, from_date, to_date, on, unit=None, where=None)`

Fetch the average of an expression for an event per time unit.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#segmentation-average>

`Segmentation.get(event, from_date, to_date, on=None, unit=None, where=None, limit=None, type=None)`

Fetch segmented and filtered data for an event.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#segmentation-default>

`Segmentation.multiseg(event, type, from_date, to_date, inner, outer, limit=None, unit=None)`

Fetch the average of an expression for an event per time unit.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#segmentation-multiseg>



`Segmentation.numeric` (*event, from\_date, to\_date, on, buckets, unit=None, where=None, type=None*)

Fetch segmented and filtered data for an event, sorted into numeric buckets.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#segmentation-numeric>

`Segmentation.sum` (*event, from\_date, to\_date, on, unit=None, where=None*)

Fetch the sum of an expression for an event per time unit.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/data-export-api#segmentation-sum>

#### 4.18.6 Service methods

`Mixpanel.engage` (*distinct\_id, data*)

Store people properties

Upstream documentation: <https://mixpanel.com/docs/people-analytics/people-http-specification-insert-data>

**Variables** *properties* (*dict*) – The user properties, your access token will be inserted into it automatically.

**Returns** A boolean that tells if the event has been logged.

`Mixpanel.export` (*from\_date, to\_date, event=None, where=None, bucket=None*)

Export raw data from your account.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/exporting-raw-data-you-inserted-into-mixpanel#export>

**Variables**

- **from\_date** (*str*) – Query start date, in yyyy-mm-dd format.
- **to\_date** (*str*) – Query finish date, in yyyy-mm-dd format.
- **event** (*list of str*) – Optional list of events to export.
- **where** (*str*) – A filter expression.
- **bucket** (*str*) – Data bucket to query.

`Mixpanel.track` (*event, properties=None, ip=False, test=False*)

Track an event.

Upstream documentation: <https://mixpanel.com/docs/api-documentation/http-specification-insert-data>

**Variables**

- **event** (*str*) – The name of the event.
- **properties** (*dict*) – The event’s properties, your access token will be inserted into it automatically.
- **ip** (*bool*) – Should Mixpanel automatically use the incoming request IP.
- **test** (*bool*) – Use a high priority rate limited queue for testing.

**Returns** A boolean that tells if the event has been logged.

## 4.19 MixRank

**class** `mixrank.MixRank` (*api\_key*)

Create a MixRank service.

**Variables** `api_key` (*str*) – The MixRank API key.

### 4.19.1 Advertiser

`MixRank.advertiser` (*advertiser*)

Return the resource corresponding to an advertiser.

**Variables** `advertiser` (*str*) – The advertiser’s domain name. Use the root domain name; in particular, do not prefix with “www.” or any other subdomain.

### 4.19.2 DisplayAd

`Advertiser.displayad` (*hash*)

Return a resource corresponding to a single display ad.

**Variables** `hash` (*str*) – A unique hash identifying this ad.

`DisplayAd.destinations` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network destinations for this ad.

Upstream documentation: [http://mixrank.com/api/documentation#displayad\\_destinations](http://mixrank.com/api/documentation#displayad_destinations)

`DisplayAd.publishers` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network publishers for this ad.

Upstream documentation: [http://mixrank.com/api/documentation#displayad\\_publishers](http://mixrank.com/api/documentation#displayad_publishers)

### 4.19.3 TextAd

`Advertiser.textad` (*hash*)

Return a resource corresponding to a single text ad.

**Variables** `hash` (*str*) – A unique hash identifying this ad.

`TextAd.destinations` (*offset=None, page\_size=None, min\_avg\_position=None, max\_avg\_position=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network destinations for this ad.

Upstream documentation: [http://mixrank.com/api/documentation#textad\\_destinations](http://mixrank.com/api/documentation#textad_destinations)

`TextAd.publishers` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network publishers for this ad.

Upstream documentation: [http://mixrank.com/api/documentation#textad\\_publishers](http://mixrank.com/api/documentation#textad_publishers)

`Advertiser.displayads` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network display ads for this advertiser.

Upstream documentation: [http://mixrank.com/api/documentation#advertiser\\_displayads](http://mixrank.com/api/documentation#advertiser_displayads)

`Advertiser.keywords` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network keywords for this advertiser.

Upstream documentation: [http://mixrank.com/api/documentation#advertiser\\_keywords](http://mixrank.com/api/documentation#advertiser_keywords)

`Advertiser.publishers` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, min\_monthly\_uniques=None, max\_monthly\_uniques=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network publishers for this advertiser.

Upstream documentation: [http://mixrank.com/api/documentation#advertiser\\_publishers](http://mixrank.com/api/documentation#advertiser_publishers)

`Advertiser.summary` ()

Fetch the advertiser's summary.

Upstream documentation: <http://mixrank.com/api/documentation#advertiser>

`Advertiser.textads` (*offset=None, page\_size=None, min\_avg\_position=None, max\_avg\_position=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network text ads for this advertiser.

Upstream documentation: [http://mixrank.com/api/documentation#advertiser\\_textads](http://mixrank.com/api/documentation#advertiser_textads)

#### 4.19.4 Keyword

`MixRank.keyword` (*keyword*)

Return the resource corresponding to a keyword.

**Variables** `keyword` (*str*) – The keyword, can contain spaces.

`Keyword.advertisers` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the advertisers that show ads for this keyword.

Upstream documentation: [http://mixrank.com/api/documentation#keyword\\_advertisers](http://mixrank.com/api/documentation#keyword_advertisers)

`Keyword.displayads` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network display ads targeting at this keyword.

Upstream documentation: [http://mixrank.com/api/documentation#keyword\\_displayads](http://mixrank.com/api/documentation#keyword_displayads)

`Keyword.summary` ()

Fetch the keyword summary.

Upstream documentation: <http://mixrank.com/api/documentation#keyword>

`Keyword.textads` (*offset=None, page\_size=None, min\_avg\_position=None, max\_avg\_position=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network text ads targeting this keyword.

Upstream documentation: [http://mixrank.com/api/documentation#keyword\\_textads](http://mixrank.com/api/documentation#keyword_textads)

## 4.19.5 Publisher

`MixRank.publisher` (*publisher*)

Return the resource corresponding to a publisher.

**Variables** `publisher` (*str*) – The publisher’s domain name. Use the root domain name; in particular, do not prefix with “www.” or any other subdomain.

`Publisher.advertisers` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the advertisers that run ads on this publisher.

Upstream documentation: [http://mixrank.com/api/documentation#publisher\\_advertisers](http://mixrank.com/api/documentation#publisher_advertisers)

`Publisher.displayads` (*offset=None, page\_size=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network display ads for this publisher.

Upstream documentation: [http://mixrank.com/api/documentation#publisher\\_displayads](http://mixrank.com/api/documentation#publisher_displayads)

`Publisher.summary` ()

Fetch the publisher’s summary.

Upstream documentation: <http://mixrank.com/api/documentation#publisher>

`Publisher.textads` (*offset=None, page\_size=None, min\_avg\_position=None, max\_avg\_position=None, min\_times\_seen=None, max\_times\_seen=None, first\_seen\_before=None, first\_seen\_after=None, last\_seen\_before=None, last\_seen\_after=None, sort\_field=None, sort\_order=None*)

Fetch the Google Display Network text ads for this publisher.

Upstream documentation: [http://mixrank.com/api/documentation#advertiser\\_textads](http://mixrank.com/api/documentation#advertiser_textads)

## 4.19.6 Service methods

`MixRank.echo` ()

Verify that the API key is valid.

## 4.20 Mozscape

**class** `mozscape.Mozscape` (*access\_id, secret\_key*)

Create a Mozscape service.

### Variables

- `access_id` (*str*) – Your Mozscape AccessID.

- **secret\_key** (*str*) – Your Mozscape Secret Key.

### 4.20.1 Metadata

Mozscape.**metadata** ()

Return the resource responsible for Mozscape Index metadata.

Metadata.**index\_stats** ()

Fetch data about the volume of information in the Mozscape Index.

Metadata.**last\_update** ()

Fetch the Unix timestamp of the last Mozscape Index update.

Metadata.**next\_update** ()

Fetch the Unix timestamp of the next Mozscape Index update.

### 4.20.2 Service methods

Mozscape.**urlmetrics** (*urls, cols*)

Fetch URL metrics for one or more URLs.

#### Variables

- **urls** (*str or list of str*) – The URLs you're interested in.
- **cols** – The sum of column constants for metrics you want to have fetched, taken from *libsaas.services.mozscape.constants*.

## 4.21 Insights

**class** newrelic.**Insights** (*account\_id, query\_key=None, insert\_key=None*)

Create a New Relic Insights service.

#### Variables

- **account\_id** (*str*) – The account id
- **query\_key** (*str*) – The query key.
- **insert\_key** (*str*) – The insert key.

### 4.21.1 Service methods

Insights.**insert** (*events*)

Submit event or events to rubicon

**Variables** *events* – Event data

Upstream documentation: <http://docs.newrelic.com/docs/rubicon/inserting-events>

Insights.**query** (*nrql*)

NRQL query

**Variables** *nrql* (*str*) – The nrql query

Upstream documentation: <http://docs.newrelic.com/docs/rubicon/using-nrql>

## 4.22 Pingdom

**class** pingdom.**Pingdom**(*username, password, app\_key*)

Create a Pingdom service.

### Variables

- **username** (*str*) – The username for the authenticated user.
- **password** (*str*) – The password for the authenticated user.
- **app\_key** (*str*) – The app\_key for the application.

### 4.22.1 Actions

Pingdom.**actions**()

Return the resource corresponding to all actions

Actions.**get**(*from\_=None, to=None, limit=None, offset=None, checkids=None, contactids=None, status=None, via=None*)

Returns a list of actions (alerts) that have been generated for your account.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceActions>

### 4.22.2 Analysis

Pingdom.**analysis**(*checkid*)

Return the resource corresponding to the analysis for specified check

**Variables** **checkid** (*str*) – The check id

Analysis.**get**(*from\_=None, to=None, limit=None, offset=None*)

Returns a list of the latest root cause analysis results for a specified check.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceAnalysis>

Analysis.**get\_raw\_analysis**(*analysisid*)

Get Raw Analysis Results

**Variables** **analysisid** (*str*) – The specified error analysis id

### 4.22.3 Check

Pingdom.**check**(*checkid*)

Return the resource corresponding to a single check

**Variables** **checkid** (*str*) – The check id

### 4.22.4 Analysis

Check.**analysis**()

Return the resource corresponding to the analysis for the check.

Analysis.**get**(*from\_=None, to=None, limit=None, offset=None*)

Returns a list of the latest root cause analysis results for a specified check.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceAnalysis>

`Analysis.get_raw_analysis` (*analysisid*)

Get Raw Analysis Results

**Variables** `analysisid` (*str*) – The specified error analysis id

### 4.22.5 Results

`Check.results` ()

Return the resource corresponding to the results for the check.

`Results.get` (*from\_=None, to=None, limit=None, offset=None, probes=None, status=None, includeanalysis=None, maxresponse=None, minresponse=None*)

Return a list of raw test results for a specified check

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceResults>

### 4.22.6 Summary

`Check.summary` ()

Return the resource corresponding to the summary for the check.

`Summary.average` (*from\_=None, to=None, probes=None, includeuptime=None, bycountry=None, byprobe=None*)

Get the average time / uptime value for a specified check and time period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.average>

`Summary.hoursofday` (*from\_=None, to=None, probes=None, uselocaltime=None*)

Returns the average response time for each hour of the day (0-23) for a specific check over a selected time period. I.e. it shows you what an average day looks like during that time period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.hoursofday>

`Summary.outage` (*from\_=None, to=None, order=None*)

Get a list of status changes for a specified check and time period. If order is specified to descending, the list is ordered by newest first. (Default is ordered by oldest first.)

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.outage>

`Summary.performance` (*from\_=None, to=None, resolution=None, includeuptime=None, probes=None, order=None*)

For a given interval in time, return a list of sub intervals with the given resolution. Useful for generating graphs. A sub interval may be a week, a day or an hour depending on the chosen resolution.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.performance>

`Summary.probes` (*from\_=None, to=None*)

Get a list of probes that performed tests for a specified check during a specified period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.probes>

`Check.delete` ()

Deletes a check. You will lose all collected data.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodDelete+Check>

`Check.get()`

Returns a detailed description of a specified check.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodGet+Detailed+Check+Information>

`Check.update(obj)`

Modify settings for a check. The provided settings will overwrite previous values. Settings not provided will stay the same as before the update. To clear an existing value, provide an empty value. Please note that you cannot change the type of a check once it has been created.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/MethodModify+Check>

## 4.22.7 Checks

`Pingdom.checks()`

Return the resource corresponding to all checks

`Checks.create(obj)`

Creates a new check with settings specified by provided parameters.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodCreate+New+Check>

`Checks.delete()`

Deletes a list of checks. You will lose all collected data.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodDelete+Multiple+Checks>

`Checks.get(limit=None, offset=None)`

Returns a list overview of all checks.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceChecks>

`Checks.update(obj)`

Pause or change resolution for multiple checks in one bulk call.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodModify+Multiple+Checks>

## 4.22.8 Contact

`Pingdom.contact(contactid)`

Return the resource corresponding to a single contact

**Variables** `contactid` (*str*) – The contact id

`Contact.delete()`

Deletes a contact.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodDelete+Contact>

`Contact.update(obj)`

Modify a contact.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/MethodModify+Contact>



### 4.22.9 Contacts

`Pingdom.contacts()`

Return the resource corresponding to all contacts

`Contacts.create(obj)`

Creates a new contact with settings specified by provided parameters.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodCreate+Contact>

`Contacts.delete()`

Deletes a list of contacts.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodDelete+Multiple+Contacts>

`Contacts.get(limit=None, offset=None)`

Returns a list of all contacts.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceContacts>

`Contacts.update(obj)`

Modifies a list of contacts.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodModify+Multiple+Contacts>

### 4.22.10 Credits

`Pingdom.credits()`

Return the resource corresponding to all credits

`Credits.get()`

Returns information about remaining checks, SMS credits and SMS auto-refill status.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceCredits>

### 4.22.11 Probes

`Pingdom.probes()`

Return the resource corresponding to all probes

`Probes.get(limit=None, offset=None, onlyactive=None, includedeleted=None)`

Returns a list of all Pingdom probe servers.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceProbes>

### 4.22.12 Reference

`Pingdom.reference()`

Return the resource corresponding to the reference of regions

`Reference.get()`

Get a reference of regions, timezones and date/time/number formats and their identifiers.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceReference>

### 4.22.13 ReportsEmail

Pingdom.**report\_email** (*reportid*)

Return the resource corresponding to a single email report

**Variables** **reportid** (*str*) – The report id

ReportsEmail.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

ReportsEmail.**delete** ()

Delete this resource.

ReportsEmail.**get** ()

ReportsEmail.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.22.14 ReportsPublic

Pingdom.**report\_public** (*reportid*)

Return the resource corresponding to a single public report

**Variables** **reportid** (*str*) – The report id

ReportsPublic.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

ReportsPublic.**delete** ()

Delete this resource.

ReportsPublic.**get** ()

ReportsPublic.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.22.15 ReportsShared

Pingdom.**report\_shared** (*reportid*)

Return the resource corresponding to a single shared report

**Variables** **reportid** (*str*) – The report id

ReportsShared.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

ReportsShared.**delete**()  
Delete this resource.

ReportsShared.**get**()

ReportsShared.**update**(*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.22.16 ReportsEmail

Pingdom.**reports\_email**()  
Return the resource corresponding to the email reports

ReportsEmail.**create**(*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

ReportsEmail.**delete**()  
Delete this resource.

ReportsEmail.**get**()

ReportsEmail.**update**(*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.22.17 ReportsPublic

Pingdom.**reports\_public**()  
Return the resource corresponding to the public reports

ReportsPublic.**create**(*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

ReportsPublic.**delete**()  
Delete this resource.

ReportsPublic.**get**()

ReportsPublic.**update**(*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.22.18 ReportsShared

Pingdom.**reports\_shared**()  
Return the resource corresponding to the shared reports

`ReportsShared.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`ReportsShared.delete()`

Delete this resource.

`ReportsShared.get()`

`ReportsShared.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.22.19 Results

`Pingdom.results(checkid)`

Return the resource corresponding to the raw test results for a specified check

**Variables** `checkid` (*str*) – The check id

`Results.get(from_=None, to=None, limit=None, offset=None, probes=None, status=None, includeanalysis=None, maxresponse=None, minresponse=None)`

Return a list of raw test results for a specified check

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceResults>

### 4.22.20 Servertime

`Pingdom.servertime()`

Return the resource corresponding to the servertime

`Servertime.get()`

Get the current time of the API server.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceServertime>

### 4.22.21 Settings

`Pingdom.settings()`

Return the resource corresponding to the settings

`Settings.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Settings.update(obj)`

Modify account-specific settings.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#MethodModify+Account+Settings>

### 4.22.22 Single

`Pingdom.single()`

Return the resource corresponding to the single test

`Single.get(**params)`

Performs a single test using a specified Pingdom probe against a specified target. Please note that this method is meant to be used sparingly, not to set up your own monitoring solution.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSingle>

### 4.22.23 Summary

`Pingdom.summary(checkid)`

Return the resource corresponding to the summary for a specified check.

**Variables** `checkid` (*str*) – The check id

`Summary.average` (*from\_=None, to=None, probes=None, includeuptime=None, bycountry=None, byprobe=None*)

Get the average time / uptime value for a specified check and time period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.average>

`Summary.hoursofday` (*from\_=None, to=None, probes=None, uselocaltime=None*)

Returns the average response time for each hour of the day (0-23) for a specific check over a selected time period. I.e. it shows you what an average day looks like during that time period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.hoursofday>

`Summary.outage` (*from\_=None, to=None, order=None*)

Get a list of status changes for a specified check and time period. If order is specified to descending, the list is ordered by newest first. (Default is ordered by oldest first.)

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.outage>

`Summary.performance` (*from\_=None, to=None, resolution=None, includeuptime=None, probes=None, order=None*)

For a given interval in time, return a list of sub intervals with the given resolution. Useful for generating graphs. A sub interval may be a week, a day or an hour depending on the chosen resolution.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.performance>

`Summary.probes` (*from\_=None, to=None*)

Get a list of probes that performed tests for a specified check during a specified period.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceSummary.probes>

### 4.22.24 Traceroute

`Pingdom.traceroute()`

Return the resource corresponding to the traceroute test

Traceroute.**get** (*host=None, probeid=None*)

Perform a traceroute to a specified target from a specified Pingdom probe.

Upstream documentation: <https://www.pingdom.com/services/api-documentation-rest/#ResourceTraceroute>

## 4.23 Pipedrive

**class** pipedrive.**Pipedrive** (*api\_token*)

Create a Pipedrive service.

**Variables** **api\_token** (*str*) – The API token

### 4.23.1 Activities

Pipedrive.**activities** ()

Return the resource corresponding to all activities

Activities.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Activities.**delete** (*ids*)

Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Activities>

Activities.**get** (*user\_id=None, start=None, limit=None, start\_date=None, end\_date=None*)

Returns all activities assigned to a particular user

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Activities>

Activities.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.23.2 Activity

Pipedrive.**activity** (*activity\_id*)

Return the resource corresponding to a single activity

**Variables** **activity\_id** (*str*) – The activity id

Activity.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Activity.**delete** ()

Delete this resource.

Activity.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Activity.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.3 ActivityType

`Pipedrive.activity_type(type_id)`

Return the resource corresponding to a single activity type

**Variables** `type_id(str)` – The activity type id

`ActivityType.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`ActivityType.delete()`

Delete this resource.

`ActivityType.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`ActivityType.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.4 ActivityTypes

`Pipedrive.activity_types()`

Return the resource corresponding to all activity types

`ActivityTypes.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`ActivityTypes.delete(ids)`

Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Activities>

`ActivityTypes.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`ActivityTypes.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.5 Authorizations

`Pipedrive.authorizations()`

Return the resource corresponding to the user authorizations

`Authorizations.get(email, password)`

Returns all authorizations for a particular user. Authorization objects contain the API tokens the user has with different company accounts in Pipedrive.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Authorizations>

### 4.23.6 Filter

`Pipedrive.condition_filter(filter_id)`

Return the resource corresponding to a single filter

**Variables** `filter_id` (*str*) – The filter id

`Filter.delete()`

Delete this resource.

`Filter.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.7 Filters

`Pipedrive.condition_filters()`

Return the resource corresponding to all filters

`Filters.delete(ids)`

Marks multiple filters as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Filters>

`Filters.get(type=None)`

Returns all filters.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Filters>

### 4.23.8 Currencies

`Pipedrive.currencies()`

Return the resource corresponding to the deals currencies

`Currencies.get(term=None)`

Returns all supported currencies which should be used when saving monetary values with other objects. The 'code' parameter of the returning objects is the currency code according to ISO 4217.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Currencies>

### 4.23.9 Deal

`Pipedrive.deal(deal_id)`

Return the resource corresponding to a single deal

**Variables** `deal_id` (*str*) – The deal id



## 4.23.10 Products

Deal.**products** ()

Returns the resource corresponding to the deal products

Products.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Products.**delete** (*product\_attachment\_id*)

Deletes a product attachment from a deal, using the *product\_attachment\_id*.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Products.**get** (*start=None, limit=None*)

Lists products attached to a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Products.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Deal.**activities** (*start=None, limit=None, done=None, exclude=None*)

Lists activities associated with a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deal.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Deal.**delete** ()

Delete this resource.

Deal.**files** (*start=None, limit=None*)

Lists files associated with a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deal.**followers** ()

Lists the followers of a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deal.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Deal.**participants** (*start=None, limit=None*)

Lists participants associated with a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deal.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Deal.update` (*start=None, limit=None*)

Lists updates about a deal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

#### 4.23.11 DealField

`Pipedrive.deal_field` (*field\_id*)

Return the resource corresponding to a single deal field

**Variables** `field_id` (*str*) – The deal field id

`DealField.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`DealField.delete` ()

Delete this resource.

`DealField.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.23.12 DealFields

`Pipedrive.deal_fields` ()

Return the resource corresponding to all deal fields

`DealFields.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`DealFields.delete` (*ids*)

Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-DealFields>

`DealFields.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.23.13 Deals

`Pipedrive.deals` ()

Return the resource corresponding to all deals

`Deals.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Deals.delete` (*ids*)

Marks multiple deals as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deals.**find** (*term*)

Searches all deals by their title.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deals.**get** (*filter\_id=None, start=None, limit=None, sort\_by=None, sort\_mode=None, owned\_by\_you=None*)

Returns all deals

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deals.**timeline** (*start\_date, interval, amount, field\_key, user\_id=None, pipeline\_id=None, filter\_id=None*)

Returns open and won deals, grouped by defined interval of time set in a date-type dealField (*field\_key*) - e.g. when month is the chosen interval, and 3 months are asked starting from January 1st, 2012, deals are returned grouped into 3 groups - January, February and March - based on the value of the given *field\_key*.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Deals>

Deals.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.23.14 File

Pipedrive.**file** (*file\_id*)

Return the resource corresponding to a single file

**Variables** *file\_id* (*str*) – The file id

File.**delete** ()

Delete this resource.

File.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

File.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.23.15 Files

Pipedrive.**files** ()

Return the resource corresponding to all files

Files.**delete** ()

Delete this resource.

Files.**get** (*start=None, limit=None*)

Returns data about all files.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Files>

Files.**update** (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.16 Goal

`Pipedrive.goal(goal_id)`

Return the resource corresponding to a single goal

**Variables** `goal_id` (*str*) – The goal id

`Goal.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Goal.delete()`

Delete this resource.

`Goal.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Goal.results(period_start=None, period_end=None)`

Lists results of a specific goal.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Goals>

`Goal.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.17 Goals

`Pipedrive.goals()`

Return the resource corresponding to all goals

`Goals.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Goals.delete()`

Delete this resource.

`Goals.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Goals.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.18 Note

`Pipedrive.note(note_id)`

Return the resource corresponding to a single note

**Variables** `note_id` (*str*) – The note id

`Note.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Note.delete()`

Delete this resource.

`Note.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Note.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.19 Notes

`Pipedrive.notes()`

Return the resource corresponding to all notes

`Notes.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Notes.delete()`

Delete this resource.

`Notes.get(user_id=None, deal_id=None, person_id=None, org_id=None, start=None, limit=None, sort_by=None, sort_mode=None, start_date=None, end_date=None)`

Returns all notes.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Notes>

`Notes.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.20 Organization

`Pipedrive.organization(organization_id)`

Return the resource corresponding to a single organization

**Variables** `organization_id` (*str*) – The organization id

`Organization.activities` (*start=None, limit=None, done=None, exclude=None*)

Lists activities associated with an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Organization.deals` (*start=None, limit=None*)

Lists deals associated with an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.delete` ()

Delete this resource.

`Organization.files` (*start=None, limit=None*)

Lists files associated with an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.followers` ()

Lists the followers of an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Organization.merge` (*merge\_with\_id*)

Merges an organization with another organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.persons` (*start=None, limit=None*)

Lists the persons of an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organization.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Organization.updates` (*start=None, limit=None*)

Lists updates about an organization.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

### 4.23.21 OrganizationField

`Pipedrive.organization_field` (*field\_id*)

Return the resource corresponding to a single organization field

**Variables** `field_id` (*str*) – The organization field id

`OrganizationField.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`OrganizationField.delete()`

Delete this resource.

`OrganizationField.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.22 OrganizationFields

`Pipedrive.organization_fields()`

Return the resource corresponding to all organization fields

`OrganizationFields.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`OrganizationFields.delete(ids)`

Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-OrganizationFields>

`OrganizationFields.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.23 Organizations

`Pipedrive.organizations()`

Return the resource corresponding to all organizations

`Organizations.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Organizations.delete(ids)`

Marks multiple organizations as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organizations.find(term, start=None, limit=None)`

Searches all organizations by their name.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organizations.get(filter_id=None, start=None, limit=None, sort_by=None, sort_mode=None)`

Returns all organizations

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Organizations>

`Organizations.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.23.24 Person

Pipedrive.**person** (*person\_id*)

Return the resource corresponding to a single person

**Variables** **person\_id** (*str*) – The person id

Person.**activities** (*start=None, limit=None, done=None, exclude=None*)

Lists activities associated with a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

Person.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Person.**deals** (*start=None, limit=None*)

Lists deals associated with a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

Person.**delete** ()

Delete this resource.

Person.**files** (*start=None, limit=None*)

Lists files associated with a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

Person.**followers** ()

Lists the followers of a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

Person.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Person.**merge** (*merge\_with\_id*)

Merges a person with another person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

Person.**products** (*start=None, limit=None*)

Lists the products of a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-products>

Person.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Person.**updates** (*start=None, limit=None*)

Lists updates about a person.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

#### 4.23.25 PersonField

Pipedrive.**person\_field** (*field\_id*)

Return the resource corresponding to a single person field



**Variables** `field_id` (*str*) – The person field id

`PersonField.create` (*obj*)  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PersonField.delete` ()  
Delete this resource.

`PersonField.get` ()  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### 4.23.26 PersonFields

`Pipedrive.person_fields` ()  
Return the resource corresponding to all person fields

`PersonFields.create` (*obj*)  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`PersonFields.delete` (*ids*)  
Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-PersonFields>

`PersonFields.get` ()  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### 4.23.27 Persons

`Pipedrive.persons` ()  
Return the resource corresponding to all persons

`Persons.create` (*obj*)  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Persons.delete` (*ids*)  
Marks multiple persons as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

`Persons.find` (*term*, *org\_id=None*, *start=None*, *limit=None*)  
Searches all persons by their name.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

`Persons.get` (*filter\_id=None*, *start=None*, *limit=None*, *sort\_by=None*, *sort\_mode=None*)  
Returns all persons

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Persons>

`Persons.update` (*obj*)  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.28 Pipeline

`Pipedrive.pipeline` (*pipeline\_id*)

Return the resource corresponding to a single pipeline

**Variables** `pipeline_id` (*str*) – The pipeline id

`Pipeline.conversion_rates` (*start\_date, end\_date, user\_id=None*)

Returns all stage-to-stage conversion and pipeline-to-close rates for given time period.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Pipelines>

`Pipeline.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Pipeline.deals` (*filter\_id=None, user\_id=None, everyone=None, stage\_id=None, start=None, limit=None*)

Lists deals in a specific pipeline across all its stages.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Pipelines>

`Pipeline.delete` ()

Delete this resource.

`Pipeline.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Pipeline.movements` (*start\_date, end\_date, user\_id=None*)

Returns statistics for deals movements for given time period.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Pipelines>

`Pipeline.update` (*obj*)

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.29 Pipelines

`Pipedrive.pipelines` ()

Return the resource corresponding to all pipelines

`Pipelines.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Pipelines.delete` ()

Delete this resource.

`Pipelines.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Pipelines.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.30 Product

`Pipedrive.product(product_id)`

Return the resource corresponding to a single product

**Variables** `product_id(str)` – The product id

`Product.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Product.deals(start=None, limit=None)`

Returns data about a deals that have a product attached to.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Products>

`Product.delete()`

Delete this resource.

`Product.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Product.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.31 ProductField

`Pipedrive.product_field(field_id)`

Return the resource corresponding to a single product field

**Variables** `field_id(str)` – The product field id

`ProductField.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`ProductField.delete()`

Delete this resource.

`ProductField.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.32 ProductFields

`Pipedrive.product_fields()`

Return the resource corresponding to all product fields

`ProductFields.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`ProductFields.delete(ids)`

Marks multiple activities as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-ProductFields>

`ProductFields.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.33 Products

`Pipedrive.products()`

Return the resource corresponding to all products

`Products.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Products.delete()`

Delete this resource.

`Products.find(term, currency=None, start=None, limit=None)`

Returns data about the products that were found. If `currency` was set in request, prices in that currency are served back.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Products>

`Products.get(start=None, limit=None)`

Returns all products

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Products>

`Products.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.23.34 Stage

`Pipedrive.stage(stage_id)`

Return the resource corresponding to a single stage

**Variables** `stage_id(str)` – The stage id

`Stage.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Stage.deals(filter_id=None, user_id=None, everyone=None, start=None, limit=None)`

Lists deals in a specific stage

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Stages>

Stage.**delete** ()

Delete this resource.

Stage.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

Stage.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.23.35 Stages

Pipedrive.**stages** ()

Return the resource corresponding to all stages

Stages.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Stages.**delete** (*ids*)

Marks multiple stages as deleted.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Stages>

Stages.**get** (*pipeline\_id=None*)

Returns data about all stages

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Stages>

Stages.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.23.36 User

Pipedrive.**user** (*user\_id*)

Return the resource corresponding to a single user

**Variables** *user\_id* (*str*) – The user id

User.**activities** (*start=None, limit=None, done=None, exclude=None*)

Lists activities associated with a user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Users>

User.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

User.**followers** ()

Lists the followers of a user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Users>

User.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

User.**merge** (*merge\_with\_id*)

Merges a user with another user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Users>

User.**updates** (*start=None, limit=None*)

Lists updates about a user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Users>

### 4.23.37 UserConnections

Pipedrive.**user\_connections** ()

Return the resource corresponding to the user connections

UserConnections.**get** ()

Returns data about all connections for the authorized user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-UserConnections>

### 4.23.38 Users

Pipedrive.**users** ()

Return the resource corresponding to all users

Users.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Users.**find** (*term*)

Searches all users by their name.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-Users>

Users.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.23.39 Service methods

Pipedrive.**search** (*term, start=None, limit=None*)

Performs a search across the account and returns SearchResults.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-SearchResults>

Pipedrive.**settings** ()

Lists settings of authorized user.

Upstream documentation: <https://developers.pipedrive.com/v1#methods-UserSettings>

## 4.24 Recurly

**class** `recurly.Recurly` (*api\_key*)

Create a Recurly service.

**Variables** `api_key` (*str*) – The API key including.

### 4.24.1 Account

`Recurly.account` (*account\_code*)

Return the resource corresponding to a single account.

### 4.24.2 AccountAdjustments

`Account.adjustments` ()

Return the resource corresponding to all charges and credits issued for the account.

`AccountAdjustments.count` (*\*args*, *\*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`AccountAdjustments.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

`AccountAdjustments.delete` ()

Delete this resource.

`AccountAdjustments.get` (*type=None*, *state=None*, *cursor=None*, *per\_page=None*)

Fetch credits and charges for an account.

**Variables**

- **type** (*str*) – The type of adjustments: ‘charge’ or ‘credit’.
- **state** (*str*) – The state of the adjustments to return: ‘pending’ or ‘invoiced’.

### 4.24.3 BillingInfo

`Account.billing_info` ()

Return the resource corresponding to the account’s current billing information.

`BillingInfo.count` (*\*args*, *\*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`BillingInfo.delete` ()

Delete this resource.

`BillingInfo.get` (*cursor=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`BillingInfo.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.24.4 AccountInvoices

`Account.invoices` ()

Return the resource corresponding to all invoices for the account.

`AccountInvoices.count` (*\*args, \*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`AccountInvoices.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

`AccountInvoices.get` (*cursor=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

#### 4.24.5 CouponRedemption

`Account.redemption` ()

Return the resource corresponding to the coupon redeemed by the account.

`CouponRedemption.count` (*\*args, \*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.



`CouponRedemption.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from `get`. Refer to the upstream documentation for details.

`CouponRedemption.delete()`

Delete this resource.

`CouponRedemption.get(cursor=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

### 4.24.6 AccountSubscriptions

`Account.subscriptions()`

Return the resource corresponding to all subscriptions for the account.

`AccountSubscriptions.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`AccountSubscriptions.get(state='live', cursor=None, per_page=None)`

Fetch all your subscription.

**Variables** `state` (*str*) – The state of subscriptions to return: “active”, “canceled”, “expired”, “future”, “in\_trial”, “live”, or “past\_due”. A subscription will belong to more than one state.

### 4.24.7 AccountTransactions

`Account.transactions()`

Return the resource corresponding to all transactions for the account.

`AccountTransactions.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`AccountTransactions.get(cursor=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`Account.count` (\*args, \*\*kwargs)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`Account.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

`Account.delete` ()

Delete this resource.

`Account.get` (*cursor=None*, *per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`Account.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.24.8 Accounts

`Recurly.accounts` ()

Return the resource corresponding to all accounts.

`Accounts.count` (\*args, \*\*kwargs)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`Accounts.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

`Accounts.get` (*state='active'*, *cursor=None*, *per\_page=None*)

Fetch accounts for your site.

**Variables** *state* (*str*) – The state of the accounts to return: 'active', 'closed', 'past\_due'. Defaults to 'active'.

### 4.24.9 Adjustment

`Recurly.adjustment (uuid)`

Return the resource corresponding to a single adjustment.

`Adjustment.count (*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Adjustment.delete ()`

Delete this resource.

`Adjustment.get (cursor=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

### 4.24.10 Coupon

`Recurly.coupon (coupon_code)`

Return the resource corresponding to a single coupon.

`Coupon.count (*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Coupon.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from `get`. Refer to the upstream documentation for details.

`Coupon.delete ()`

Delete this resource.

`Coupon.get (cursor=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

### 4.24.11 Coupons

`Recurly.coupons()`

Return the resource corresponding to all coupons.

`Coupons.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Coupons.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from `get`. Refer to the upstream documentation for details.

`Coupons.get(state=None, cursor=None, per_page=None)`

Fetch all your coupons.

**Variables** `state` (`str`) – The state of coupons to return: “redeemable”, “expired” or “maxed\_out”.

### 4.24.12 Invoice

`Recurly.invoice(invoice_number)`

Return the resource corresponding to a single invoice.

`Invoice.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Invoice.get(cursor=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (`int`) – For collections, where should paging start. If left as `None`, the first page is returned.
- **per\_page** (`int`) – For collections, how many objects could be returned. The maximum is 200. If left as `None`, 50 objects are returned.

`Invoice.get_pdf(language='en-US')`

Fetch a PDF blob for the invoice.

**Variables** `language` (`str`) – The language for the invoice, defaults to “en-US”.

`Invoice.mark_failed()`

Mark an invoice as failed collection

`Invoice.mark_successful()`

Mark an invoice as paid successfully

### 4.24.13 Invoices

Recurly.**invoices** ()

Return the resource corresponding to all invoices.

Invoices.**count** (\*args, \*\*kwargs)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

Invoices.**get** (state=None, cursor=None, per\_page=None)

Fetch all your invoices.

**Variables state** (*str*) – The state of invoices to return: “open”, “collected”, “failed”, or “past\_due”.

### 4.24.14 Plan

Recurly.**plan** (plan\_code)

Return the resource corresponding to a single plan.

### 4.24.15 Addon

Plan.**addon** (add\_on\_code)

Return the resource corresponding to a single plan’s add-on.

Addon.**count** (\*args, \*\*kwargs)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

Addon.**delete** ()

Delete this resource.

Addon.**get** (cursor=None, per\_page=None)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

#### Variables

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

Addon.**update** (obj)

Update this resource.

**Variables obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.24.16 Addons

`Plan.addons()`

Return the resource corresponding to all the add-ons for the plan.

`Addons.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`Addons.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

`Addons.get(cursor=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`Plan.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

`Plan.delete()`

Delete this resource.

`Plan.get(cursor=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`Plan.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.24.17 Plans

`Recurly.plans()`

Return the resource corresponding to all plans.

Plans.**count** (*\*args, \*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

Plans.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same as returned from *get*. Refer to the upstream documentation for details.

Plans.**get** (*cursor=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

## 4.24.18 Subscription

Recurly.**subscription** (*uuid*)

Return the resource corresponding to a single subscription.

Subscription.**cancel** ()

Cancel a subscription, remaining it as active until next billing cycle.

Subscription.**count** (*\*args, \*\*kwargs*)

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as *get*.

Subscription.**get** (*cursor=None, per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

Subscription.**postpone** (*next\_renewal\_date*)

Postpone a subscription

**Variables** *next\_renewal\_date* (*str*) – The next renewal date that will be applied

Subscription.**reactivate** ()

Reactivating a canceled subscription.

Subscription.**terminate** (*refund=None*)

Terminate a subscription, removing any stored billing information.

**Variables** *refund* (*str*) – The type of the refund to perform: ‘full’ or ‘partial’ Defaults to ‘none’.

`Subscription.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.24.19 Subscriptions

`Recurly.subscriptions()`

Return the resource corresponding to all subscriptions.

`Subscriptions.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Subscriptions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from `get`. Refer to the upstream documentation for details.

`Subscriptions.get(state='live', cursor=None, per_page=None)`

Fetch all your subscription.

**Variables** `state` (*str*) – The state of subscriptions to return: “active”, “canceled”, “expired”, “future”, “in\_trial”, “live”, or “past\_due”. A subscription will belong to more than one state.

### 4.24.20 Transaction

`Recurly.transaction(uuid)`

Return the resource corresponding to a single transaction.

`Transaction.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Transaction.get(cursor=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **cursor** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 200. If left as *None*, 50 objects are returned.

`Transaction.refund(amount_in_cents=None)`

Refund or void a previous, successful transaction.



## 4.24.21 Transactions

`Recurly.transactions()`

Return the resource corresponding to all transactions.

`Transactions.count(*args, **kwargs)`

Fetch an integer count of the number of objects of a collection. This is an absolute number, regardless of paging limits, so use this if you want to tally up a collection instead of iterating through all of its objects.

For single-object resources, returns one.

Accepts the same arguments as `get`.

`Transactions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same as returned from `get`. Refer to the upstream documentation for details.

`Transactions.get(state=None, type=None, cursor=None, per_page=None)`

Fetch all your transactions.

### Variables

- **state** (`str`) – The state of transactions to return: “successful”, “failed”, or “voided”.
- **type** (`str`) – The type of transactions to return: “authorization”, “refund”, or “purchase”.

## 4.25 SegmentIO

`class segmentio.SegmentIO(api_secret)`

Create a SegmentIO service

**Variables** `api_secret` (`str`) – Your project’s API secret.

### 4.25.1 User

`SegmentIO.user(user_id)`

Return the resource corresponding to a single user

`User.identify(traits=None, context=None, timestamp=None)`

Identify an user.

### Variables

- **traits** (`dict`) – A dictionary of traits you know about the user.
- **context** (`dict`) – A dictionary of provider specific options.
- **timestamp** – An ISO 8601 date string representing

when the identify took place. :vartype timestamp: str

`User.track(event, properties=None, context=None, timestamp=None)`

Track an event.

### Variables

- **event** (`str`) – The name of the event you’re tracking.
- **properties** (`dict`) – A dictionary of properties for the event.

- **context** (*dict*) – A dictionary of provider specific options.
- **timestamp** – An ISO 8601 date string representing

when the event took place. :vartype timestamp: str

## 4.25.2 Service methods

SegmentIO.**alias** (*from\_user\_id, to\_user\_id, context=None, timestamp=None*)

Identify an user.

### Variables

- **from\_user\_id** (*str*) – The anonymous user’s id before they are logged in.
- **to\_user\_id** (*str*) – The identified user’s id after they’re logged in.
- **context** (*dict*) – A dictionary of provider specific options.
- **timestamp** – An ISO 8601 date string representing

when the identify took place. :vartype timestamp: str

SegmentIO.**batch\_import** (*batch, context=None*)

The import method lets you send a series of identify, group, track, page and screen requests in a single batch

### Variables

- **batch** (*dict*) – List of actions.
- **context** (*dict*) – A dictionary of provider specific options.

## 4.26 Spotify

**class** spotify.**Spotify**

Create a Spotify service.

### 4.26.1 Lookup

Spotify.**lookup**()

Return the resource corresponding to the lookup service

Lookup.**get** (*uri, extras=None*)

Lookup for an artist, album or track in the Spotify’s music catalogue

### Variables

- **uri** (*str*) – Spotify valid uri
- **extras** (*str*) – A comma-separated list of words that defines the detail level expected in the response.

See <https://developer.spotify.com/technologies/web-api/lookup/>

## 4.26.2 Search

`Spotify.search()`

Return the resource corresponding to the search service

`Search.get(type, q, page=None)`

Search in the Spotify's music catalogue.

See <https://developer.spotify.com/technologies/web-api/search/> and <http://www.spotify.com/es/about/features/advanced-search-syntax/>

### Variables

- **type** (*str*) – What to search for, artist, album or track.
- **q** (*str*) – Search string.
- **page** (*int*) – The page of the result set to return. defaults to 1

## 4.27 Stripe

`class stripe.Stripe(api_key)`

Create a Stripe service.

**Variables** `api_key` (*str*) – The API key.

### 4.27.1 Account

`Stripe.account()`

Return the resource corresponding to the logged account.

`Account.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.27.2 BalanceHistory

`Stripe.balance_history()`

Return the resource corresponding to the balance history.

`BalanceHistory.get(limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

### Variables

- **limit** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **ending\_before** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

### 4.27.3 Charge

`Stripe.charge(id)`

Return the resource corresponding to a single charge.

`Charge.dispute(obj)`

Update a dispute

**Variables** `obj` – a Python object representing the updated dispute.

`Charge.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Charge.refund(amount=None)`

Refunding a charge

**Variables** `amount` (*int*) – A positive integer in cents representing how much of this charge to refund. Can only refund up to the unrefunded amount remaining of the charge. Default is entire charge.

### 4.27.4 Charges

`Stripe.charges()`

Return the resource corresponding to all charges.

`Charges.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Charges.get(customer=None, limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

**Variables**

- **`customer`** (*str*) – Only return charges for the customer specified by this customer ID.
- **`limit`** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **`ending_before`** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **`starting_after`** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

### 4.27.5 Coupon

`Stripe.coupon(id)`

Return the resource corresponding to a single coupon.

`Coupon.delete()`

Delete this resource.

`Coupon.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.27.6 Coupons

`Stripe.coupons()`

Return the resource corresponding to all coupons.

`Coupons.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Coupons.get(limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

**Variables**

- **limit** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **ending\_before** (`str`) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (`str`) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

## 4.27.7 Customer

`Stripe.customer(id)`

Return the resource corresponding to a single customer.

## 4.27.8 DiscountResource

`Customer.discount()`

Return the resource corresponding to a single discount.

`DiscountResource.delete()`

Delete this resource.

## 4.27.9 SubscriptionResource

`Customer.subscription(subscription_id)`

Return the resource corresponding to a single customer's subscription.

**Variables** `subscription_id` (`str`) – The subscription's id.

`SubscriptionResource.delete()`

Delete this resource.

`SubscriptionResource.get()`

Fetch the object's data.

`SubscriptionResource.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.27.10 SubscriptionsResource

`Customer.subscriptions()`

Return the resource corresponding to the customer's subscriptions.

`SubscriptionsResource.get()`

Fetch the object's data.

`Customer.delete()`

Delete this resource.

`Customer.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Customer.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.27.11 Customers

`Stripe.customers()`

Return the resource corresponding to all customers.

`Customers.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Customers.get(total_count=False, limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

**Variables**

- **total\_count** (*bool*) – Include the total count of all customers.
- **limit** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **ending\_before** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

### 4.27.12 Event

`Stripe.event(id)`

Return the resource corresponding to a single event.

`Event.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.27.13 Events

`Stripe.events()`

Return the resource corresponding to all events.

`Events.get` (*type=None, limit=None, ending\_before=None, starting\_after=None*)

Fetch all of the objects.

#### Variables

- **type** (*str*) – A string containing a specific event name, or group of events using \* as a wildcard. The list will be filtered to include only events with a matching event property.
- **limit** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **ending\_before** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

### 4.27.14 Invoice

`Stripe.invoice(id)`

Return the resource corresponding to a single invoice.

### 4.27.15 LineItems

`Invoice.lines()`

Return the resource corresponding to all invoice's lines.

`LineItems.get` (*customer=None, limit=None, ending\_before=None, starting\_after=None*)

Fetch all of the objects.

#### Variables

- **customer** (*str*) – In the case of upcoming invoices, the customer of the upcoming invoice is required. In other cases it is ignored.
- **limit** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **ending\_before** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

`Invoice.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Invoice.pay()`

Paying an invoice

`Invoice.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.27.16 InvoiceItem

`Stripe.invoiceitem(id)`

Return the resource corresponding to a single invoiceitem.

`InvoiceItem.delete()`

Delete this resource.

`InvoiceItem.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.27.17 InvoiceItems

`Stripe.invoiceitems()`

Return the resource corresponding to all invoiceitems.

`InvoiceItems.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`InvoiceItems.get(customer=None, limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

#### Variables

- **`customer`** (*str*) – The identifier of the customer whose invoice items to return. If none is provided, all invoices will be returned.
- **`limit`** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.
- **`ending_before`** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **`starting_after`** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

### 4.27.18 Invoices

`Stripe.invoices()`

Return the resource corresponding to all invoices.

`Invoices.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Invoices.get(customer=None, limit=None, ending_before=None, starting_after=None)`

Fetch all of the objects.

#### Variables

- **`customer`** (*str*) – The identifier of the customer whose invoices to return. If none is provided, all invoices will be returned.
- **`limit`** – A limit on the number of objects to be returned. Count can range between 1 and 100 objects.



- **ending\_before** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be newer than the given object.
- **starting\_after** (*str*) – A cursor (object ID) for use in pagination. Fetched objects will be older than the given object.

`Invoices.upcoming(customer)`

Fetch a customer's upcoming invoice.

**Variables** *customer* (*str*) – The identifier of the customer whose invoices to return. If none is provided, all invoices will be returned.

#### 4.27.19 Plan

`Stripe.plan(id)`

Return the resource corresponding to a single plan.

`Plan.delete()`

Delete this resource.

`Plan.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Plan.update(obj)`

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.27.20 Plans

`Stripe.plans()`

Return the resource corresponding to all plans.

`Plans.create(obj)`

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Plans.get(limit=10)`

Fetch all plans.

**Variables** *limit* (*int*) – A limit on the number of objects to be returned. Limit can range between 1 and 100 items.

#### 4.27.21 Token

`Stripe.token(id)`

Return the resource corresponding to a single token.

`Token.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

## 4.27.22 Tokens

`Stripe.tokens()`

Return the resource corresponding to all tokens.

`Tokens.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.28 Trello

`class trello.Trello(key, token=None)`

Create a Trello service.

**Variables**

- `key` (*str*) – Your application key
- `token` (*str*) – The authorization token from the user (optional).

### 4.28.1 Action

`Trello.action(action_id)`

Return the resource corresponding to a single action.

### 4.28.2 Board

`Action.board()`

Returns a single board

`Board.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource's field.

`Board.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

### 4.28.3 Card

`Action.card()`

Returns a single card

`Card.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource's field.

`Card.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

#### 4.28.4 MemberCreator

Action.**creator**()

Returns a single creator

MemberCreator.**field**(*field*)

Returns a single resource field.

**Variables** *field*(*str*) – a valid resource’s field.

MemberCreator.**get**(*fields=None*)

Fetch a single object.

**Variables** *fields*(*list*) – all or comma-separated list of fields.

#### 4.28.5 List

Action.**list**()

Returns a single list

List.**field**(*field*)

Returns a single resource field.

**Variables** *field*(*str*) – a valid resource’s field.

List.**get**(*fields=None*)

Fetch a single object.

**Variables** *fields*(*list*) – all or comma-separated list of fields.

#### 4.28.6 Member

Action.**member**()

Returns a single member

Member.**field**(*field*)

Returns a single resource field.

**Variables** *field*(*str*) – a valid resource’s field.

Member.**get**(*fields=None*)

Fetch a single object.

**Variables** *fields*(*list*) – all or comma-separated list of fields.

#### 4.28.7 Organization

Action.**organization**()

Returns a single organization

Organization.**field**(*field*)

Returns a single resource field.

**Variables** *field*(*str*) – a valid resource’s field.

Organization.**get**(*fields=None*)

Fetch a single object.

**Variables** *fields*(*list*) – all or comma-separated list of fields.

Action.**delete**()

Delete this resource.

Action.**field**(*field*)

Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

Action.**get**(*fields=None, entities=None, member=None, member\_fields=None, memberCreator=None, memberCreator\_fields=None*)

Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/action/index.html#get-1-actions-idaction>

Action.**update**(*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.28.8 Board

Trello.**board**(*board\_id*)

Return the resource corresponding to a single board

### 4.28.9 Actions

Board.**actions**()

Returns all actions

Actions.**get**(*\*\*kwargs*)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.10 Card

Board.**card**(*card\_id*)

Returns a single card

Card.**get**(*fields=None*)

Fetch a single object.

**Variables** *fields* (*list*) – all or comma-separated list of fields.

### 4.28.11 Cards

Board.**cards**()

Returns all cards

Cards.**filter**(*filter\_id*)

Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

`Cards.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.12 Checklists

`Board.checklists ()`  
Returns all checklists

`Checklists.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Checklists.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.13 Lists

`Board.lists ()`  
Returns all lists

`Lists.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Lists.filter (filter_id)`  
Fetch a collection filtered.

**Variables** `filter` (`str`) – a valid resource’s filter.

`Lists.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.14 Member

`Board.member (member_id)`  
Returns a single member

`Member.delete ()`  
Delete this resource.

`Member.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.15 Members

`Board.members()`

Returns all members

`Members.filter(filter_id)`

Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Members.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.16 MembersInvited

`Board.members_invited()`

Returns all invited members

`MembersInvited.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`MembersInvited.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

### 4.28.17 Membership

`Board.membership(membership_id)`

Returns a single membership

`Membership.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

`Membership.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.18 Memberships

`Board.memberships()`

Returns all memberships

`Memberships.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.19 Organization

`Board.organization()`

Returns a single organization

`Organization.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Organization.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

`Board.calendar_key()`

Generates a calendar key.

`Board.email_key()`

Generates a email key.

`Board.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Board.get(**kwargs)`

Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/board/index.html#get-1-boards-board-id>

`Board.mark_as_viewed()`

Marks board as viewed.

`Board.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.20 Boards

`Trello.boards()`

Return the resource corresponding to all boards

`Boards.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.21 Card

`Trello.card(card_id_or_shortlink)`

Return the resource corresponding to a single card

#### 4.28.22 Actions

`Card.actions()`  
Returns all actions

#### 4.28.23 Comments

`Actions.comments()`  
Returns all comments

`Comments.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Comments.delete()`

`Comments.update(obj)`

`Actions.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.24 Attachment

`Card.attachment(attachment_id)`  
Returns a single checklist

`Attachment.delete()`  
Delete this resource.

`Attachment.get(fields=None)`  
Fetch a single object.

**Variables** `fields` (`list`) – all or comma-separated list of fields.

#### 4.28.25 Attachments

`Card.attachments()`  
Returns all attachments

`Attachments.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Attachments.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>



### 4.28.26 Board

`Card.board()`

Returns a single board

`Board.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Board.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

### 4.28.27 CheckItemStates

`Card.checkitem_states()`

Returns all checkitem states

`CheckItemStates.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.28 Checklist

`Card.checklist(checklist_id)`

Returns a single checklist

### 4.28.29 CheckItem

`Checklist.checkitem(checkitem_id)`

Returns a single checkitem

`CheckItem.convert_to_card()`

Converts checkitem to card.

`CheckItem.delete()`

Delete this resource.

`CheckItem.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.30 CheckItems

`Checklist.checkitems()`

Returns all checkitems

`CheckItems.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.31 Checklists

`Card.checklists()`  
Returns all checklists

`Checklists.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Checklists.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.32 Label

`Card.label(color)`  
Returns a single label

`Label.delete()`  
Delete this resource.

### 4.28.33 Labels

`Card.labels()`  
Returns all labels

`Labels.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.34 List

`Card.list()`  
Returns a single list

`List.field(field)`  
Returns a single resource field.

**Variables** `field` (`str`) – a valid resource's field.

`List.get(fields=None)`  
Fetch a single object.

**Variables** `fields` (`list`) – all or comma-separated list of fields.

### 4.28.35 Members

`Card.members()`

Returns all members

`Members.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.36 MembersVoted

`Card.members_voted()`

Returns all voted members

`MembersVoted.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`MembersVoted.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.37 Sticker

`Card.sticker(sticker_id)`

Returns a single sticker

`Sticker.delete()`

Delete this resource.

`Sticker.get(fields=None)`

Fetch a single object.

**Variables** `fields` (`list`) – all or comma-separated list of fields.

`Sticker.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.38 Stickers

`Card.stickers()`

Returns all stickers

`Stickers.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Stickers.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`Card.field(field)`  
Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Card.get(**kwargs)`  
Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/card/index.html#get-1-cards-card-id-or-shortlink>

`Card.mark_as_read()`  
Marks associated notification as read.

`Card.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.39 Cards

`Trello.cards()`  
Return the resource corresponding to all cards

`Cards.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.40 Checklist

`Trello.checklist(checklist_id)`  
Return the resource corresponding to a single checklist

### 4.28.41 Board

`Checklist.board()`  
Returns a single board

`Board.field(field)`  
Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Board.get(fields=None)`  
Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

#### 4.28.42 Cards

`Checklist.cards()`

Returns all cards

`Cards.filter(filter_id)`

Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Cards.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.43 CheckItem

`Checklist.checkitem(checkitem_id)`

Returns a single checkitem

`CheckItem.delete()`

Delete this resource.

`CheckItem.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

#### 4.28.44 CheckItems

`Checklist.checkitems()`

Returns all checkitems

`CheckItems.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CheckItems.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`Checklist.delete()`

Delete this resource.

`Checklist.field(field)`

Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Checklist.get(**kwargs)`

Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/checklist/index.html>

`Checklist.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.45 Checklists

`Trello.checklists()`

Return the resource corresponding to all checklists

`Checklists.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.46 List

`Trello.list(list_id)`

Return the resource corresponding to a single list

#### 4.28.47 Actions

`List.actions()`

Returns all actions

`Actions.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.48 Board

`List.board()`

Returns a single board

`Board.field(field)`

Returns a single resource field.

**Variables** `field` (`str`) – a valid resource’s field.

`Board.get(fields=None)`

Fetch a single object.

**Variables** `fields` (`list`) – all or comma-separated list of fields.

#### 4.28.49 Cards

`List.cards()`

Returns all cards

`Cards.filter(filter_id)`

Fetch a collection filtered.

**Variables** `filter` (`str`) – a valid resource’s filter.

`Cards.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`List.archive_all_cards ()`  
Archive all list cards.

`List.field (field)`  
Returns a single resource field.

**Variables** `field (str)` – a valid resource’s field.

`List.get (**kwargs)`  
Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/list/index.html#get-1-lists-idlist>

`List.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.50 Lists

`Trello.lists ()`  
Return the resource corresponding to all lists

`Lists.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.51 Member

`Trello.me ()`  
Return the resource corresponding to the current member

#### 4.28.52 Actions

`Member.actions ()`  
Returns all actions

`Actions.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.53 BoardBackgrounds

`Member.board_background (board_background_id)`  
Returns a single board background

`BoardBackgrounds.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`BoardBackgrounds.get (**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.54 BoardBackgrounds

`Member.board_backgrounds ()`

Returns all board backgrounds

`BoardBackgrounds.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`BoardBackgrounds.get (**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.55 BoardStar

`Member.board_star (board_star_id)`

Returns a single board star

`BoardStar.delete ()`

Delete this resource.

`BoardStar.update (obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.56 BoardStars

`Member.board_stars ()`

Returns all board stars

`BoardStars.create (obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`BoardStars.get ()`

#### 4.28.57 Boards

`Member.boards ()`

Returns all boards



`Boards.filter (filter_id)`  
Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Boards.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.58 Cards

`Member.cards ()`  
Returns all cards

`Cards.filter (filter_id)`  
Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Cards.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.59 CustomBoardBackground

`Member.custom_board_background (board_background_id)`  
Returns a single custom board background

`CustomBoardBackground.delete ()`  
Delete this resource.

`CustomBoardBackground.get (fields=None)`  
Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

`CustomBoardBackground.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

#### 4.28.60 CustomBoardBackgrounds

`Member.custom_board_backgrounds ()`  
Returns all custom board backgrounds

`CustomBoardBackgrounds.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`CustomBoardBackgrounds.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.61 CustomSticker

Member.**custom\_sticker** (*sticker\_id*)  
Returns a single custom stickers

CustomSticker.**delete** ()  
Delete this resource.

CustomSticker.**get** (*fields=None*)  
Fetch a single object.

**Variables** *fields* (*list*) – all or comma-separated list of fields.

### 4.28.62 CustomStickers

Member.**custom\_stickers** ()  
Returns all custom stickers

CustomStickers.**create** (*obj*)  
Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

CustomStickers.**get** (\*\**kwargs*)  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.63 Notifications

Member.**notifications** ()  
Returns all notifications

Notifications.**filter** (*filter\_id*)  
Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

Notifications.**get** (\*\**kwargs*)  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.64 Organizations

Member.**organizations** ()  
Returns all organizations

Organizations.**filter** (*filter\_id*)  
Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

Organizations.**get** (\*\**kwargs*)  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.65 Sessions

`Member.sessions()`  
Returns all sessions

`Sessions.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.66 Tokens

`Member.tokens()`  
Returns all tokens

`Tokens.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`Member.field(field)`  
Returns a single resource field.

**Variables** `field` (*str*) – a valid resource’s field.

`Member.get(**kwargs)`  
Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/member/index.html>

`Member.update(obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.28.67 Member

`Trello.member(member_id_or_username)`  
Return the resource corresponding to a single member

### 4.28.68 Actions

`Member.actions()`  
Returns all actions

`Actions.get(**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.69 BoardBackgrounds

`Member.board_background(board_background_id)`  
Returns a single board background

BoardBackgrounds.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

BoardBackgrounds.**get** (\*\**kwargs*)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.70 BoardBackgrounds

Member.**board\_backgrounds** ()

Returns all board backgrounds

BoardBackgrounds.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

BoardBackgrounds.**get** (\*\**kwargs*)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.71 BoardStar

Member.**board\_star** (*board\_star\_id*)

Returns a single board star

BoardStar.**delete** ()

Delete this resource.

BoardStar.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.28.72 BoardStars

Member.**board\_stars** ()

Returns all board stars

BoardStars.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

BoardStars.**get** ()

### 4.28.73 Boards

`Member.boards()`

Returns all boards

`Boards.filter(filter_id)`

Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Boards.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.74 Cards

`Member.cards()`

Returns all cards

`Cards.filter(filter_id)`

Fetch a collection filtered.

**Variables** `filter` (*str*) – a valid resource’s filter.

`Cards.get(**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.75 CustomBoardBackground

`Member.custom_board_background(board_background_id)`

Returns a single custom board background

`CustomBoardBackground.delete()`

Delete this resource.

`CustomBoardBackground.get(fields=None)`

Fetch a single object.

**Variables** `fields` (*list*) – all or comma-separated list of fields.

`CustomBoardBackground.update(obj)`

Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.28.76 CustomBoardBackgrounds

`Member.custom_board_backgrounds()`

Returns all custom board backgrounds

`CustomBoardBackgrounds.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

CustomBoardBackgrounds.**get** (\*\*kwargs)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.77 CustomSticker

Member.**custom\_sticker** (sticker\_id)

Returns a single custom stickers

CustomSticker.**delete** ()

Delete this resource.

CustomSticker.**get** (fields=None)

Fetch a single object.

**Variables** **fields** (*list*) – all or comma-separated list of fields.

#### 4.28.78 CustomStickers

Member.**custom\_stickers** ()

Returns all custom stickers

CustomStickers.**create** (obj)

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

CustomStickers.**get** (\*\*kwargs)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.79 Notifications

Member.**notifications** ()

Returns all notifications

Notifications.**filter** (filter\_id)

Fetch a collection filtered.

**Variables** **filter** (*str*) – a valid resource's filter.

Notifications.**get** (\*\*kwargs)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.80 Organizations

Member.**organizations** ()

Returns all organizations

Organizations.**filter** (filter\_id)

Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

`Organizations.get (**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.81 Sessions

`Member.sessions ()`

Returns all sessions

`Sessions.get (**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

### 4.28.82 Tokens

`Member.tokens ()`

Returns all tokens

`Tokens.get (**kwargs)`

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`Member.field (field)`

Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

`Member.get (**kwargs)`

Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/member/index.html>

`Member.update (obj)`

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.28.83 Notification

`Trello.notification (notification_id)`

Return the resource corresponding to a single notification

### 4.28.84 Board

`Notification.board ()`

Returns a single board

`Board.field (field)`

Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

`Board.get (fields=None)`

Fetch a single object.

**Variables** `fields (list)` – all or comma-separated list of fields.

#### 4.28.85 Card

`Notification.card ()`

Returns a single card

`Card.field (field)`

Returns a single resource field.

**Variables** `field (str)` – a valid resource's field.

`Card.get (fields=None)`

Fetch a single object.

**Variables** `fields (list)` – all or comma-separated list of fields.

#### 4.28.86 MemberCreator

`Notification.creator ()`

Returns a single creator

`MemberCreator.field (field)`

Returns a single resource field.

**Variables** `field (str)` – a valid resource's field.

`MemberCreator.get (fields=None)`

Fetch a single object.

**Variables** `fields (list)` – all or comma-separated list of fields.

#### 4.28.87 List

`Notification.list ()`

Returns a single list

`List.field (field)`

Returns a single resource field.

**Variables** `field (str)` – a valid resource's field.

`List.get (fields=None)`

Fetch a single object.

**Variables** `fields (list)` – all or comma-separated list of fields.

#### 4.28.88 Member

`Notification.member ()`

Returns a single member

`Member.field (field)`

Returns a single resource field.



**Variables** *field* (*str*) – a valid resource’s field.

Member.**get** (*fields=None*)

Fetch a single object.

**Variables** *fields* (*list*) – all or comma-separated list of fields.

#### 4.28.89 Organization

Notification.**organization** ()

Returns a single organization

Organization.**field** (*field*)

Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

Organization.**get** (*fields=None*)

Fetch a single object.

**Variables** *fields* (*list*) – all or comma-separated list of fields.

Notification.**field** (*field*)

Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

Notification.**get** (*\*\*kwargs*)

Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/notification/index.html>

Notification.**update** (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.28.90 Organization

Trello.**organization** (*organization\_id\_or\_name*)

Return the resource corresponding to a single organization

#### 4.28.91 Actions

Organization.**actions** ()

Returns all actions

Actions.**get** (*\*\*kwargs*)

Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.92 Boards

Organization.**boards** ()

Returns all boards

Boards.**filter** (*filter\_id*)  
Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

Boards.**get** (\*\**kwargs*)  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.93 Member

Organization.**member** (*member\_id*)  
Returns a single member

Member.**delete** ()  
Delete this resource.

Member.**update** (*obj*)  
Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.28.94 Members

Organization.**members** ()  
Returns all members

Members.**filter** (*filter\_id*)  
Fetch a collection filtered.

**Variables** *filter* (*str*) – a valid resource’s filter.

Members.**get** (\*\**kwargs*)  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

#### 4.28.95 MembersInvited

Organization.**members\_invited** ()  
Returns all invited members

MembersInvited.**field** (*field*)  
Returns a single resource field.

**Variables** *field* (*str*) – a valid resource’s field.

MembersInvited.**get** (*fields=None*)  
Fetch a single object.

**Variables** *fields* (*list*) – all or comma-separated list of fields.

## 4.28.96 Membership

`Organization.membership (membership_id)`  
Returns a single membership

`Membership.delete ()`  
Delete this resource.

`Membership.get (fields=None)`  
Fetch a single object.

**Variables** `fields (list)` – all or comma-separated list of fields.

`Membership.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.28.97 Memberships

`Organization.memberships ()`  
Returns all memberships

`Memberships.get (**kwargs)`  
Fetch a collection.

Upstream documentation: <https://trello.com/docs/api/>

`Organization.delete ()`  
Delete this resource.

`Organization.field (field)`  
Returns a single resource field.

**Variables** `field (str)` – a valid resource’s field.

`Organization.get (**kwargs)`  
Fetch a single object.

Upstream documentation: <https://trello.com/docs/api/organization/index.html>

`Organization.update (obj)`  
Update this resource.

**Variables** `obj` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.28.98 Organizations

`Trello.organizations ()`  
Return the resource corresponding to all organizations

`Organizations.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

## 4.29 Twilio

**class** `twilio.Twilio` (*account\_sid*, *auth\_token*)

Create a Twilio service.

### Variables

- **account\_sid** (*str*) – The users’s account SID
- **auth\_token** (*str*) – THe account’s API token

### 4.29.1 Account

`Twilio.account` (*sid*)

Return the representation of an Account or SubAccount.

### 4.29.2 Application

`Account.application` (*sid*)

Return a Application resource representation, representing an application within this account.

`Application.delete` ()

Delete this resource.

`Application.get` ()

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

`Application.update` (*obj*)

Update this resource.

**Variables** *obj* (*dict*) – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.29.3 Applications

`Account.applications` ()

Return a list of Application resource representations, each representing an application within this account.

`Applications.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Applications.get` (*FriendlyName=None*, *Page=None*, *PageSize=None*, *AfterSid=None*)

Fetch the Applications belonging to an account.

### Variables

- **FriendlyName** (*str*) – Only return the Account resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.

- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.4 AuthorizedConnectApp

`Account.authorized_connect_app` (*sid*)

Return a Connect App resource representation, representing a Connect App you've authorized to access this account.

`AuthorizedConnectApp.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.29.5 AuthorizedConnectApps

`Account.authorized_connect_apps` ()

Return a list of Connect App resource representations, each representing a Connect App you've authorized to access this account.

`AuthorizedConnectApps.get` (*Page=None, PageSize=None, AfterSid=None*)

Fetch the Authorized Connect Apps belonging to an account.

##### Variables

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.6 AvailablePhoneNumbers

`Account.available_phone_numbers` ()

Return an AvailablePhoneNumbers resource that allows querying local and toll-free available for this account.

#### 4.29.7 AvailablePhoneNumbersLocal

`AvailablePhoneNumbers.local` (*country\_code*)

Return a list of local AvailablePhoneNumber resource representations that match the specified filters, each representing a phone number that is currently available for provisioning within this account.

`AvailablePhoneNumbersLocal.get` (*AreaCode=None, Contains=None, InRegion=None, InPostalCode=None, NearLatLong=None, NearNumber=None, InLata=None, InRateCenter=None, Distance=None*)

Fetch available local phone numbers for an account.

##### Variables

- **AreaCode** (*str*) – Find phone numbers in the specified area code.
- **Contains** (*str*) – A pattern to match phone numbers on. Valid characters are \* and [0-9a-zA-Z]. The \* character will match any single digit.
- **InRegion** (*str*) – Limit results to a particular region (State/Province). Given a phone number, search within the same Region as that number. (US and Canada only)

- **InPostalCode** (*str*) – Limit results to a particular postal code. Given a phone number, search within the same postal code as that number. (US and Canada only)
- **NearLatLong** (*str*) – Given a latitude/longitude pair lat,long find geographically close numbers within Distance miles. (US and Canada only)
- **NearNumber** (*str*) – Given a phone number, find a geographically close number within Distance miles. Distance defaults to 25 miles. (US and Canada only)
- **InLata** (*str*) – Limit results to a specific Local access and transport area (LATA). Given a phone number, search within the same LATA as that number. (US and Canada only)
- **InRateCenter** (*str*) – Limit results to a specific rate center, or given a phone number search within the same rate center as that number. Requires InLata to be set as well. (US and Canada only)
- **InDistance** (*int*) – Specifies the search radius for a Near- query in miles. If not specified this defaults to 25 miles. (US and Canada only)

#### 4.29.8 AvailablePhoneNumbersTollFree

AvailablePhoneNumbers.**toll\_free** (*country\_code*)

Return a list of toll-free AvailablePhoneNumber resource representations that match the specified filters, each representing a phone number that is currently available for provisioning within this account.

AvailablePhoneNumbersTollFree.**get** (*AreaCode=None, Contains=None*)

Fetch available toll-free phone numbers for an account.

##### Variables

- **AreaCode** (*str*) – Find phone numbers in the specified area code.
- **Contains** (*str*) – A pattern to match phone numbers on. Valid characters are \* and [0-9a-zA-Z]. The \* character will match any single digit.

#### 4.29.9 Call

Account.**call** (*sid*)

Return a phone call made to and from this account.

#### 4.29.10 Notifications

Call.**notifications** ()

Return a list of notifications generated for this call.

Notifications.**get** (*Log=None, MessageDate=None, MessageDateGT=None, MessageDateLT=None, Page=None, PageSize=None, AfterSid=None*)

Fetch notifications for an account or call.

##### Variables

- **Log** (*int*) – Only show notifications for this log, using the integer log values.
- **MessageDate** (*str*) – Only show notifications for this date, given as YYYY-MM-DD.
- **MessageDateGT** (*str*) – Greater than inequality for MessageDate, use it for messages logged at or after midnight on a date (generates MessageDate>=YYYY-MM-DD).

- **MessageDateLT** – Lower than inequality for MessageDate, use it for messages logged at or before midnight on a date (generates MessageDate<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.11 Recordings

Call.**recordings** ()

Return a list of Recording resource representations, each representing a recording generated during the course of this phone call.

Recordings.**get** (*CallSid=None, DateCreated=None, DateCreatedGT=None, DateCreatedLT=None, Page=None, PageSize=None, AfterSid=None*)

Fetch the list of transcriptions for an account or call.

#### Variables

- **CallSid** (*str*) – Show only recordings made during the call given by this sid.
- **DateCreated** (*str*) – Only show recordings created on this date, given as YYYY-MM-DD.
- **DateCreatedGT** (*str*) – Greater than inequality for DateCreated, use it for recordings created at or after midnight on a date (generates DateCreated>=YYYY-MM-DD).
- **DateCreatedLT** – Lower than inequality for DateCreated, use it for recordings created at or before midnight on a date (generates DateCreated<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

Call.**get** ()

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

Call.**update** (*obj*)

Update this resource.

**Variables** *obj* (*dict*) – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.29.12 Calls

Account.**calls** ()

Return a list of phone calls made to and from this account.

Calls.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Calls.get` (*To=None, From=None, Status=None, StartTime=None, StartTimeGT=None, StartTimeLT=None, ParentCallSid=None, Page=None, PageSize=None, AfterSid=None*)  
 Fetch the calls made to or from an account.

#### Variables

- **To** (*str*) – Only show calls to this phone number or Client identifier.
- **From** (*str*) – Only show calls from this phone number or Client identifier.
- **Status** (*str*) – Only show calls currently in this status. May be *queued, ringing, in-progress, completed, failed, busy* or *no-answer*.
- **StartTime** (*str*) – Only show calls that started on this date, given as YYYY-MM-DD.
- **StartTimeGT** (*str*) – Greater than inequality for StartTime, use it for calls that started at or after midnight on a date (generates `StartTime>=YYYY-MM-DD`).
- **StartTimeLT** – Lower than inequality for StartTime, use it for calls that started at or before midnight on a date (generates `StartTime<=YYYY-MM-DD`).
- **ParentCallSid** (*str*) – Only show calls spawned by the call with this Sid.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.13 Conference

`Account.conference` (*sid*)  
 Return a conference within this account.

### 4.29.14 Participant

`Conference.participant` (*sid*)  
 Return a participant in this conference.

`Participant.delete` ()  
 Delete this resource.

`Participant.get` ()  
 For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Participant.update` (*obj*)  
 Update this resource.

**Variables** *obj* (*dict*) – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.29.15 Participants

`Conference.participants` ()  
 Return the list of participants in this conference.

`Participants.get` (*Muted=None, Page=None, PageSize=None, AfterSid=None*)  
 Fetch the participants of a conference.



### Variables

- **Muted** (*bool*) – Only show participants that are muted or unmuted. Either *True* or *False*.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`Conference.get()`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

## 4.29.16 Conferences

`Account.conferences()`

Return a list of conferences within this account.

`Conferences.get(Status=None, FriendlyName=None, DateCreated=None, DateCreatedGT=None, DateCreatedLT=None, DateUpdated=None, DateUpdatedGT=None, DateUpdatedLT=None, Page=None, PageSize=None, AfterSid=None)`

Fetch the calls made to or from an account.

### Variables

- **Status** (*str*) – Only show conferences currently in with this status. May be *init*, *in-progress*, or *completed*.
- **FriendlyName** (*str*) – List conferences who’s *FriendlyName* is the exact match of this string.
- **DateCreated** (*str*) – Only show conferences that started on this date, given as YYYY-MM-DD.
- **DateCreatedGT** (*str*) – Greater than inequality for *DateCreated*, use it for conferences that started at or after midnight on a date (generates *DateCreated*>=YYYY-MM-DD).
- **DateCreatedLT** – Lower than inequality for *DateCreated*, use it for conferences that started at or before midnight on a date (generates *DateCreated*<=YYYY-MM-DD).
- **DateUpdated** (*str*) – Only show conferences that were last updated on this date, given as YYYY-MM-DD.
- **DateUpdatedGT** (*str*) – Greater than inequality for *DateUpdated*, use it for conferences that were last updated at or after midnight on a date (generates *DateUpdated*>=YYYY-MM-DD).
- **DateUpdatedLT** – Lower than inequality for *DateUpdated*, use it for conferences that were last updated at or before midnight on a date (generates *DateUpdated*<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.17 ConnectApp

`Account.connect_app (sid)`

Return a Connect App resource representations, representing a Connect App in this account.

`ConnectApp.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`ConnectApp.update (obj)`

Update this resource.

**Variables** `obj (dict)` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.18 ConnectApps

`Account.connect_apps ()`

Return a list of Connect App resource representations, each representing a Connect App in this account.

`ConnectApps.get (Page=None, PageSize=None, AfterSid=None)`

Fetch the Connect Apps belonging to an account.

**Variables**

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.19 IncomingPhoneNumber

`Account.incoming_phone_number (sid)`

Return an IncomingPhoneNumber resource representation, representing a phone number given to this account.

`IncomingPhoneNumber.delete ()`

Delete this resource.

`IncomingPhoneNumber.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`IncomingPhoneNumber.update (obj)`

Update this resource.

**Variables** `obj (dict)` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.20 IncomingPhoneNumbers

`Account.incoming_phone_numbers ()`

Return a list of IncomingPhoneNumber resource representations, each representing a phone number given to this account.

### 4.29.21 IncomingPhoneNumbersLocal

`IncomingPhoneNumbers.local()`

`IncomingPhoneNumbersLocal.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IncomingPhoneNumbersLocal.get(PhoneNumber=None, FriendlyName=None, Page=None, PageSize=None, AfterSid=None)`  
Fetch incoming phone numbers list for an account.

#### Variables

- **PhoneNumber** (*str*) – Only show the incoming phone number resources that match this pattern. You can specify partial numbers and use `*` as a wildcard for any digit.
- **FriendlyName** (*str*) – Only show the incoming phone number resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.22 IncomingPhoneNumbersTollFree

`IncomingPhoneNumbers.toll_free()`

`IncomingPhoneNumbersTollFree.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IncomingPhoneNumbersTollFree.get(PhoneNumber=None, FriendlyName=None, Page=None, PageSize=None, AfterSid=None)`  
Fetch incoming phone numbers list for an account.

#### Variables

- **PhoneNumber** (*str*) – Only show the incoming phone number resources that match this pattern. You can specify partial numbers and use `*` as a wildcard for any digit.
- **FriendlyName** (*str*) – Only show the incoming phone number resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`IncomingPhoneNumbers.create(obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`IncomingPhoneNumbers.get(PhoneNumber=None, FriendlyName=None, Page=None, PageSize=None, AfterSid=None)`  
 Fetch incoming phone numbers list for an account.

**Variables**

- **PhoneNumber** (*str*) – Only show the incoming phone number resources that match this pattern. You can specify partial numbers and use `*` as a wildcard for any digit.
- **FriendlyName** (*str*) – Only show the incoming phone number resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.23 Notification

`Account.notification(sid)`  
 Return a notification generated for this account.

`Notification.delete()`  
 Delete this resource.

`Notification.get()`  
 For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

### 4.29.24 Notifications

`Account.notifications()`  
 Return a list of notifications generated for this account.

`Notifications.get(Log=None, MessageDate=None, MessageDateGT=None, MessageDateLT=None, Page=None, PageSize=None, AfterSid=None)`  
 Fetch notifications for an account or call.

**Variables**

- **Log** (*int*) – Only show notifications for this log, using the integer log values.
- **MessageDate** (*str*) – Only show notifications for this date, given as YYYY-MM-DD.
- **MessageDateGT** (*str*) – Greater than inequality for MessageDate, use it for messages logged at or after midnight on a date (generates MessageDate>=YYYY-MM-DD).
- **MessageDateLT** – Lower than inequality for MessageDate, use it for messages logged at or before midnight on a date (generates MessageDate<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.25 OutgoingCallerId

`Account.outgoing_caller_id(sid)`

Return an `OutgoingCallerId` resource representation, representing a Caller ID number valid for this account.

`OutgoingCallerId.delete()`

Delete this resource.

`OutgoingCallerId.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`OutgoingCallerId.update(obj)`

Update this resource.

**Variables** `obj` (*dict*) – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.26 OutgoingCallerIds

`Account.outgoing_caller_ids()`

Return a list of `OutgoingCallerId` resource representations, each representing a Caller ID number valid for this account.

`OutgoingCallerIds.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`OutgoingCallerIds.get(PhoneNumber=None, FriendlyName=None, Page=None, PageSize=None, AfterSid=None)`

Fetch outgoing caller ids for an account.

#### Variables

- **PhoneNumber** (*str*) – Only show the incoming phone number resources that match this pattern. You can specify partial numbers and use `*` as a wildcard for any digit.
- **FriendlyName** (*str*) – Only show the incoming phone number resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.27 Queue

`Account.queue(sid)`

Return a queue within this account.

### 4.29.28 Member

`Queue.member(sid)`

Return a member in this queue.

`Member.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Member.update(obj)`

Update this resource.

**Variables** `obj` (*dict*) – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.29.29 Members

`Queue.members()`

Return the list of members in this queue.

`Members.get(Page=None, PageSize=None, AfterSid=None)`

Fetch the list of members for a conference.

### Variables

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`Queue.delete()`

Delete this resource.

`Queue.get()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`Queue.update(obj)`

Update this resource.

**Variables** `obj` (*dict*) – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

## 4.29.30 Queues

`Account.queues()`

Return a list of queues within this account.

`Queues.create(obj)`

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Queues.get(Page=None, PageSize=None, AfterSid=None)`

Fetch the list of conferences of an account.

### Variables

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.

- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.31 Recording

`Account.recording` (*sid*)

Return a Recording resource representation, representing a recording generated during the course of a phone call made to or from this account.

### 4.29.32 Transcriptions

`Recording.transcriptions` ()

Return a set of Transcription resource representations for this recording.

`Transcriptions.get` (*Page=None, PageSize=None, AfterSid=None*)

Fetch the list of transcriptions for an account or call.

#### Variables

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`Recording.delete` ()

Delete this resource.

`Recording.get` ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.29.33 Recordings

`Account.recordings` ()

Return a list of Recording resource representations, each representing a recording generated during the course of a phone call made to or from this account.

`Recordings.get` (*CallSid=None, DateCreated=None, DateCreatedGT=None, DateCreatedLT=None, Page=None, PageSize=None, AfterSid=None*)

Fetch the list of transcriptions for an account or call.

#### Variables

- **CallSid** (*str*) – Show only recordings made during the call given by this sid.
- **DateCreated** (*str*) – Only show recordings created on this date, given as YYYY-MM-DD.
- **DateCreatedGT** (*str*) – Greater than inequality for DateCreated, use it for recordings created at or after midnight on a date (generates DateCreated>=YYYY-MM-DD).
- **DateCreatedLT** – Lower than inequality for DateCreated, use it for recordings created at or before midnight on a date (generates DateCreated<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.

- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.34 SMS

Account.**sms** ()

Return a SMS resource to query messages and short codes resources.

#### 4.29.35 Message

SMS.**message** (*sid*)

Return a SMS message associated with this account.

Message.**get** ()

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### 4.29.36 Messages

SMS.**messages** ()

Return a list of SMS messages associated with this account.

Messages.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

Messages.**get** (*To=None, From=None, DateSent=None, DateSentGT=None, DateSentLT=None, Page=None, PageSize=None, AfterSid=None*)

Fetch the list of SMS messages associated with an account.

**Variables**

- **To** (*str*) – Only show SMS messages to this phone number.
- **From** (*str*) – Only show SMS messages from this phone number.
- **DateSent** (*str*) – Only show SMS messages on this date, given as YYYY-MM-DD.
- **DateSentGT** (*str*) – Greater than inequality for DateSent, use it for message sent at or after midnight on a date (generates DateSent>=YYYY-MM-DD).
- **DateSentLT** – Lower than inequality for DateSent, use it for messages sent at or before midnight on a date (generates DateSent<=YYYY-MM-DD).
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.



### 4.29.37 ShortCode

`SMS.short_code (sid)`

Return a ShortCode resource representation, representing a short code within this account.

`ShortCode.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

`ShortCode.update (obj)`

Update this resource.

**Variables** `obj (dict)` – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.38 ShortCodes

`SMS.short_codes ()`

Return a list of ShortCode resource representations, each representing a short code within this account.

`ShortCodes.get (ShortCode=None, FriendlyName=None, Page=None, PageSize=None, AfterSid=None)`

Fetch the list of short codes for an account.

#### Variables

- **ShortCode** (*str*) – Only show the ShortCode resources that match this pattern. You can specify partial numbers and use `*` as a wildcard for any digit.
- **FriendlyName** (*str*) – Only show the ShortCode resources with friendly names that exactly match this name.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.39 Transcription

`Account.transcription (sid)`

Return a Transcription resource representation for call made to of from this account.

`Transcription.get ()`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### 4.29.40 Transcriptions

`Account.transcriptions ()`

Return a set of Transcription resource representations for this account.

`Transcriptions.get (Page=None, PageSize=None, AfterSid=None)`

Fetch the list of transcriptions for an account or call.

#### Variables

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.

- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.41 Usage

`Account.usage()`

Return a usage resource to query records and triggers resources.

#### 4.29.42 Records

`Usage.records()`

Return a list of usage records.

#### 4.29.43 RecordsAllTime

`Records.all_time()`

Return a single usage record for each usage category, each representing usage over the date-range specified. This is the same as the root `.usage().records()`.

`RecordsAllTime.get(Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None)`

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, `StartDate=-30days` will make `StartDate` be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, `EndDate=+30days` will make `EndDate` be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.44 RecordsDaily

`Records.daily()`

Return multiple usage records for each usage category, each representing usage over a daily time-interval.

`RecordsDaily.get(Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None)`

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.45 RecordsLastMonth

Records.**last\_month** ()

Return a single usage record per usage category, for last month's usage only.

RecordsLastMonth.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.46 RecordsMonthly

Records.**monthly** ()

Return multiple usage records for each usage category, each representing usage over a monthly time-interval.

RecordsMonthly.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.

- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.47 RecordsThisMonth

Records.**this\_month** ()

Return a single usage record per usage category, for this month's usage only.

RecordsThisMonth.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.48 RecordsToday

Records.**today** ()

Return a single usage record per usage category, for today's usage only.

RecordsToday.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.

- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.49 RecordsYearly

Records.**yearly** ()

Return multiple usage records for each usage category, each representing usage over a yearly time-interval.

RecordsYearly.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

#### 4.29.50 RecordsYesterday

Records.**yesterday** ()

Return a single usage record per usage category, for yesterday's usage only.

RecordsYesterday.**get** (*Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage records.

##### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, StartDate=-30days will make StartDate be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, EndDate=+30days will make EndDate be 30 days from today.

- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`Records.get (Category=None, StartDate=None, EndDate=None, Page=None, PageSize=None, AfterSid=None)`  
Fetch a list of usage records.

#### Variables

- **Category** (*str*) – Only include usage records of this usage category.
- **StartDate** (*str*) – Only include usage that has occurred on or after this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, `StartDate=-30days` will make `StartDate` be 30 days before today.
- **EndDate** (*str*) – Only include usage that has occurred on or before this date. Format is YYYY-MM-DD. All dates are in GMT. As a convenience, you can also specify offsets to today. For example, `EndDate=+30days` will make `EndDate` be 30 days from today.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

### 4.29.51 Trigger

`Usage.trigger (sid)`  
Return an usage trigger set on this account.

`Trigger.delete ()`  
Delete this resource.

`Trigger.get ()`  
For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

`Trigger.update (obj)`  
Update this resource.

**Variables** `obj` (*dict*) – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.52 Triggers

`Usage.triggers ()`  
Return a list of usage triggers set on this account.

`Triggers.create (obj)`  
Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Triggers.get` (*Recurring=None, UsageCategory=None, TriggerBy=None, Page=None, PageSize=None, AfterSid=None*)

Fetch a list of usage triggers resource representations.

**Variables**

- **Recurring** (*str*) – Only show usage triggers that count over this interval. One of daily, monthly, or yearly. To retrieve non-recurring triggers, leave this empty or use alltime.
- **UsageCategory** (*str*) – Only include usage triggers that watch this usage category.
- **TriggerBy** (*str*) – Only show usage triggers that trigger by this field in the usage record. Must be one of: count, usage, or price.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.
- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

`Account.get` ()

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

`Account.update` (*obj*)

Update this resource.

**Variables** `obj` (*dict*) – a Python object representing the updated resource, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

### 4.29.53 Accounts

`Twilio.accounts` ()

Return the set of Accounts resources belonging to the Account used to make the API request. This list includes that account, along with any subaccounts belonging to it.

You can use the Accounts list resource to create subaccounts and retrieve the subaccounts that exist under your main account.

`Accounts.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python object representing the resource to be created, usually in the same format as returned from `get`. Refer to the upstream documentation for details.

`Accounts.get` (*FriendlyName=None, Status=None, Page=None, PageSize=None, AfterSid=None*)

Fetch the (sub)accounts belonging to this account.

**Variables**

- **FriendlyName** (*str*) – Only return the Account resources with friendly names that exactly match this name.
- **Status** (*str*) – Only return Account resources with the given status. Can be *closed*, *suspended* or *active*.
- **Page** (*int*) – The current page number. Zero-indexed, so the first page is 0.
- **PageSize** (*int*) – How many resources to return in each list page. The default is 50, and the maximum is 1000.

- **AfterSid** (*str*) – The last Sid returned in the previous page, used to avoid listing duplicated resources if new ones are created while paging.

## 4.30 UserVoice

**class** `uservoice.UserVoice` (*subdomain*, *api\_key*, *api\_secret=None*, *access\_token=None*, *access\_token\_secret=None*)

Create a UserVoice service.

### Variables

- **subdomain** (*str*) – The account-specific part of the UserVoice domain, for instance use *mycompany* if your UserVoice domain is *mycompany.uservoice.com*.
- **api\_key** (*str*) – The API key.
- **api\_secret** (*str or None*) – Optional API secret. If you leave this as *None*, all requests will be made as unauthenticated requests.
- **access\_token** (*str or None*) – Optional OAuth 1.0a access token. If you leave this as *None*, all requests be made as unauthenticated requests.
- **access\_token\_secret** (*str or None*) – Optional OAuth 1.0a access token secret. If you leave this as *None*, all requests be made as unauthenticated requests.

### 4.30.1 Article

`UserVoice.article` (*article\_id*)

Return the resource corresponding to a single article.

`Article.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Article.delete` ()

Delete this resource.

`Article.get` (*page=None*, *per\_page=None*, *sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Article.update` (*obj*)

Update this resource.



**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Article.useful()`

Mark the article as useful.

## 4.30.2 Articles

`UserVoice.articles()`

Return the resource corresponding to all the articles.

`Articles.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Articles.delete()`

Delete this resource.

`Articles.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Articles.search(page=None, per_page=None, query=None)`

Search for articles.

### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

`Articles.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.3 Comments

`UserVoice.comments()`

Return the resource corresponding to all the comments.

`Comments.delete()`

Delete this resource.

`Comments.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Comments.update(text)`

Update this resource.

**Variables** **text** (*str*) – the new text of the resource.

### 4.30.4 CustomFields

`UserVoice.custom_fields()`

Return the resource corresponding to custom fields.

`CustomFields.delete()`

Delete this resource.

`CustomFields.get(page=None, per_page=None, filter=None, sort=None)`

Fetch all custom fields.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects should be returned. If left as *None*, 10 objects are returned.
- **filter** (*str*) – The kind of fields to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`CustomFields.public(page=None, per_page=None, sort=None)`

Fetch public custom fields.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects should be returned. If left as *None*, 10 objects are returned.
- **filter** (*str*) – The kind of fields to return, see upstream documentation for possible values.

- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

CustomFields.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.5 Faq

UserVoice.**faq** (*faq\_id*)

Return the resource corresponding to a single FAQ.

### 4.30.6 FaqFlags

Faq.**flags** ()

Return the resource corresponding to all the flags of this FAQ.

FaqFlags.**create** (*flag*)

Create a new flag.

**Variables** **flag** (*str*) – The flag name. Refer to the upstream documentation for details.

FaqFlags.**delete** ()

Delete this resource.

FaqFlags.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

Faq.**delete** ()

Delete this resource.

Faq.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.7 Forum

`UserVoice.forum(forum_id)`

Return the resource corresponding to a single forum.

### 4.30.8 ForumCategories

`Forum.categories()`

Return a resource corresponding to all the categories on this forum.

`ForumCategories.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`ForumCategories.delete()`

Delete this resource.

`ForumCategories.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as `None`, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. If left as `None`, 10 objects are returned.
- `sort` (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumCategories.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.9 ForumCategory

`Forum.category(category_id)`

Return a resource corresponding to a single category on this forum.

`ForumCategory.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`ForumCategory.delete()`

Delete this resource.

`ForumCategory.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumCategory.update(obj)`

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.10 Stream

`Forum.stream()`

Return a resource corresponding to the stream of events for this forum.

`Stream.private(date=None, filter=None, since=None)`

Fetch all private events.

#### Variables

- **date** (*str*) – Fetch only events from that day (EST). See upstream documentation for details.
- **filter** – Specify which event types you want. See upstream documentation for allowed values.
- **filter** – *str*
- **since** (*str*) – Fetch events from that moment onward. If set, the *date* parameter is ignored. See upstream documentation for details.

`Stream.public(date=None, filter=None, since=None)`

Fetch all public events.

#### Variables

- **date** (*str*) – Fetch only events from that day (EST). See upstream documentation for details.
- **filter** – Specify which event types you want. See upstream documentation for allowed values.
- **filter** – *str*
- **since** (*str*) – Fetch events from that moment onward. If set, the *date* parameter is ignored. See upstream documentation for details.

### 4.30.11 ForumSuggestion

Forum.**suggestion** (*suggestion\_id*)

Return a resource corresponding to a single suggestion on a forum.

### 4.30.12 ForumSuggestionComment

ForumSuggestion.**comment** (*comment\_id*)

Return the resource corresponding to a single comment on this suggestion.

### 4.30.13 SuggestionCommentFlags

ForumSuggestionComment.**flags** ()

Return the resource corresponding to all the flags of this comment.

SuggestionCommentFlags.**create** (*flag*)

Create a new flag.

**Variables** **flag** (*str*) – The flag name. Refer to the upstream documentation for details.

SuggestionCommentFlags.**delete** ()

Delete this resource.

SuggestionCommentFlags.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

ForumSuggestionComment.**create** (*text*)

Create a new resource.

**Variables** **text** (*str*) – the text of the resource to be created.

ForumSuggestionComment.**delete** ()

Delete this resource.

ForumSuggestionComment.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

ForumSuggestionComment.**update** (*text*)

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

#### 4.30.14 ForumSuggestionComments

`ForumSuggestion.comments()`

Return the resource corresponding to all the comments on this suggestion.

`ForumSuggestionComments.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`ForumSuggestionComments.delete()`

Delete this resource.

`ForumSuggestionComments.get(page=None, per_page=None, filter=None, sort=None)`

Fetch comments on this suggestion.

##### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **filter** (*str*) – The kind of comments to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumSuggestionComments.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

#### 4.30.15 SuggestionFlags

`ForumSuggestion.flags()`

Return the resource corresponding to all the flags of this suggestion.

`SuggestionFlags.create(flag)`

Create a new flag.

**Variables** `flag` (*str*) – The flag name. Refer to the upstream documentation for details.

`SuggestionFlags.delete()`

Delete this resource.

`SuggestionFlags.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

### 4.30.16 ForumSuggestionNote

`ForumSuggestion.note(note_id)`

Return the resource corresponding to a single note on this suggestion.

`ForumSuggestionNote.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`ForumSuggestionNote.delete()`

Delete this resource.

`ForumSuggestionNote.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumSuggestionNote.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

### 4.30.17 ForumSuggestionNotes

`ForumSuggestion.notes()`

Return the resource corresponding to all the notes on this suggestion.

`ForumSuggestionNotes.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`ForumSuggestionNotes.delete()`

Delete this resource.

`ForumSuggestionNotes.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumSuggestionNotes.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.



`ForumSuggestion.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`ForumSuggestion.delete()`

Delete this resource.

`ForumSuggestion.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- `page` (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- `sort` (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumSuggestion.respond(obj)`

Respond to a suggestion.

**Variables** `obj` – a Python object representing the response. Refer to the upstream documentation for details.

`ForumSuggestion.supporters(page=None, per_page=None, sort=None)`

Fetch the supporters for this suggestion.

**Variables**

- `page` (*int*) – Where should paging start. If left as *None*, the first page is returned.
- `per_page` (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- `sort` (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`ForumSuggestion.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`ForumSuggestion.vote()`

Vote for this suggestion.

### 4.30.18 ForumSuggestions

`Forum.suggestions()`

Return a resource corresponding to all the suggestion on a forum.

`ForumSuggestions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{‘forum’: {‘name’: ‘Forum Name’}}`, then `obj` should be `{‘name’: ‘New Forum’}`.

Refer to the upstream documentation for details.

`ForumSuggestions.delete()`

Delete this resource.

`ForumSuggestions.get(page=None, per_page=None, category=None, filter=None, sort=None, updated_after_date=None)`

Fetch suggestions from this forum.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **category** (*str*) – Either a category ID, *all* or *uncategorized*. See upstream documentation for details.
- **filter** (*str*) – The kind of suggestions to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.
- **updated\_after\_date** – If *filter* is *assigned\_after*, a date string formatted *yyyy-mm-dd HH:MM:SS -0000*.
- **updated\_after\_date** – *str*

`ForumSuggestions.search(page=None, per_page=None, category_id=None, query=None)`

Search for suggestions on this forum.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **category\_id** (*int*) – A category ID.
- **query** (*str*) – Search string.

`ForumSuggestions.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{‘forum’: {‘name’: ‘Forum Name’}}`, then `obj` should be `{‘name’: ‘New Forum’}`.

Refer to the upstream documentation for details.

### 4.30.19 ForumUserSuggestions

`Forum.user_suggestions(user_id)`

Return a resource corresponding to all the suggestions of a single user on this forum.

`ForumUserSuggestions.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`ForumUserSuggestions.get(page=None, per_page=None, category=None, filter=None, sort=None)`

Fetch suggestions from this user on this forum.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **category** (*str*) – Either a category ID, *all* or *uncategorized*. See upstream documentation for details.
- **filter** (*str*) – The kind of suggestions to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`Forum.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Forum.delete()`

Delete this resource.

`Forum.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Forum.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.20 Forums

`UserVoice.forums()`

Return the resource corresponding to all the forums.

`Forums.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Forums.delete()`

Delete this resource.

`Forums.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Forums.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.21 Gadget

`UserVoice.gadget(gadget_id)`

Return the resource corresponding to a single gadget.

`Gadget.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Gadget.delete()`

Delete this resource.

`Gadget.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

Gadget.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

## 4.30.22 Gadgets

UserVoice.**gadgets** ()

Return the resource corresponding to all the gadgets.

Gadgets.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

Gadgets.**delete** ()

Delete this resource.

Gadgets.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

Gadgets.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.23 Notes

`UserVoice.notes()`

Return the resource corresponding to all the notes.

`Notes.delete()`

Delete this resource.

`Notes.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Notes.update(obj)`

Update this resource.

**Variables obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.24 Stream

`UserVoice.stream()`

Return the resource corresponding to a stream.

`Stream.private(date=None, filter=None, since=None)`

Fetch all private events.

#### Variables

- **date** (*str*) – Fetch only events from that day (EST). See upstream documentation for details.
- **filter** – Specify which event types you want. See upstream documentation for allowed values.
- **filter** – str
- **since** (*str*) – Fetch events from that moment onward. If set, the *date* parameter is ignored. See upstream documentation for details.

`Stream.public(date=None, filter=None, since=None)`

Fetch all public events.

#### Variables

- **date** (*str*) – Fetch only events from that day (EST). See upstream documentation for details.
- **filter** – Specify which event types you want. See upstream documentation for allowed values.

- *filter* – str
- *since* (*str*) – Fetch events from that moment onward. If set, the *date* parameter is ignored. See upstream documentation for details.

### 4.30.25 Subdomain

UserVoice.**subdomain** (*subdomain*)

Return the resource corresponding to a UserVoice subdomain.

Subdomain.**get** ()

Fetch information about the subdomain.

### 4.30.26 Suggestion

UserVoice.**suggestion** (*suggestion\_id*)

Return the resource corresponding to a single suggestion.

Suggestion.**create** (*obj*)

Create a new resource.

**Variables** *obj* – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

Suggestion.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

### 4.30.27 Suggestions

UserVoice.**suggestions** ()

Return the resource corresponding to all the suggestions.

Suggestions.**delete** ()

Delete this resource.

Suggestions.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.

- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

Suggestions.**search** (*page=None, per\_page=None, query=None*)

Search for suggestions.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

Suggestions.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.28 SupportQueue

UserVoice.**support\_queue** (*queue\_id*)

Return the resource corresponding to a single support queue.

SupportQueue.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

SupportQueue.**delete** ()

Delete this resource.

SupportQueue.**get** (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

SupportQueue.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.



### 4.30.29 SupportQueues

`UserVoice.support_queues()`

Return the resource corresponding to all the support queues.

`SupportQueues.create(obj)`

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`SupportQueues.delete()`

Delete this resource.

`SupportQueues.get(page=None, per_page=None)`

Fetch all the support queues.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.

`SupportQueues.sort(order)`

Change the order of support queues.

**Variables** `order` (*list*) – A list of support queue IDs in the desired new ordering.

`SupportQueues.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.30 Ticket

`UserVoice.ticket(ticket_id)`

Return the resource corresponding to a single ticket.

### 4.30.31 TicketMessages

`Ticket.messages()`

Return the resource corresponding to all the ticket messages.

`TicketMessages.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`TicketMessages.delete()`

Delete this resource.

`TicketMessages.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`TicketMessages.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

### 4.30.32 TicketNote

`Ticket.note(note_id)`

Return the resource corresponding to a single ticket note.

`TicketNote.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`TicketNote.delete()`

Delete this resource.

`TicketNote.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`TicketNote.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

### 4.30.33 TicketNotes

`Ticket.notes()`

Return the resource corresponding to all the ticket notes.

`TicketNotes.create(text)`

Create a new resource.

**Variables** `text` (*str*) – the text of the resource to be created.

`TicketNotes.delete()`

Delete this resource.

`TicketNotes.get` (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`TicketNotes.update` (*text*)

Update this resource.

**Variables** **text** (*str*) – the new text of the resource.

`Ticket.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Ticket.delete` ()

Delete this resource.

`Ticket.get` (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Ticket.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.34 Tickets

`UserVoice.tickets` ()

Return the resource corresponding to all the tickets.

`Tickets.create` (*obj*)

Create a new resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Tickets.delete()`

Delete this resource.

`Tickets.get(page=None, per_page=None, assignee_id=None, support_queue_id=None, support_queue=None, filter=None, sort=None, state=None, updated_after_date=None)`  
Fetch all of the tickets.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **assignee\_id** (*int*) – The ID of the user assigned to the ticket.
- **support\_queue\_id** (*int*) – The ID of the support queue the ticket is in.
- **support\_queue** (*str*) – The name of the support queue the ticket is in.
- **filter** (*str*) – Either *all* or *assigned\_after*.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.
- **state** (*str*) – Ticket state. Refer to upstream documentation for possible values.
- **updated\_after\_date** – If *filter* is *assigned\_after*, a date string formatted *yyyy-mm-dd HH:MM:SS -0000*.
- **updated\_after\_date** – *str*

`Tickets.search(page=None, per_page=None, query=None)`

Search for tickets.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

`Tickets.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Tickets.upsert(obj)`

Create or update a ticket

**Variables** `obj` – a Python object representing the ticket. Refer to the upstream documentation for details.

### 4.30.35 Topic

`UserVoice.topic(topic_id)`

Return the resource corresponding a single topic.

`Topic.articles(page=None, per_page=None, sort=None)`

Fetch the articles on a given topic.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`Topic.create(obj)`

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Topic.delete()`

Delete this resource.

`Topic.search(page=None, per_page=None, query=None)`

Search for articles on a given topic.

#### Variables

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

### 4.30.36 Topics

`UserVoice.topics()`

Return the resource corresponding all the topics.

`Topics.delete()`

Delete this resource.

`Topics.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Topics.update(obj)`

Update this resource.

**Variables** `obj` – a Python dictionary representing the resource to be created, in the same as returned from `get`, but one level less nested. For instance, if `get` returns `{'forum': {'name': 'Forum Name'}}`, then `obj` should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.37 User

`UserVoice.user(user_id=None)`

Return the resource corresponding to a single user. If `user_id` is `None`, the returned resource is the currently authenticated user, otherwise it is the user with the given ID number.

### 4.30.38 UserComments

`User.comments()`

Return the resource corresponding to all of this user's comments.

`UserComments.delete()`

Delete this resource.

`UserComments.get(page=None, per_page=None, filter=None, sort=None)`

Fetch comments from this user.

**Variables**

- **page** (*int*) – Where should paging start. If left as `None`, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as `None`, 10 objects are returned.
- **filter** (*str*) – The kind of comments to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

`UserComments.update(text)`

Update this resource.

**Variables** `text` (*str*) – the new text of the resource.

### 4.30.39 UserNotes

`User.notes()`

Return the resource corresponding to all of this user's notes.

`UserNotes.delete()`

Delete this resource.

`UserNotes.get(page=None, per_page=None, sort=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as `None`, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

UserNotes.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

#### 4.30.40 UserSuggestions

User.**suggestions** ()

Return a resource corresponding to all of this user's suggestions.

UserSuggestions.**delete** ()

Delete this resource.

UserSuggestions.**get** (*page=None, per\_page=None, category=None, filter=None, sort=None*)

Fetch suggestions from this user on this forum.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **category** (*str*) – Either a category ID, *all* or *uncategorized*. See upstream documentation for details.
- **filter** (*str*) – The kind of suggestions to return, see upstream documentation for possible values.
- **sort** (*str*) – How should the returned collection be sorted. Refer to upstream documentation for possible values.

UserSuggestions.**update** (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

User.**create** (*obj*)

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

User.**delete** ()

Delete this resource.

`User.get` (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`User.update` (*obj*)

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.41 Users

`UserVoice.users` ()

Return the resource corresponding to all the users.

`Users.create` (*obj*)

Create a new resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

`Users.delete` ()

Delete this resource.

`Users.get` (*page=None, per\_page=None, sort=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects should be returned. If left as *None*, 10 objects are returned.
- **sort** (*str*) – For collections, how should the returned collection be sorted. Refer to upstream documentation for possible values.

`Users.search` (*page=None, per\_page=None, guid=None, query=None*)

Search for users. One of *guid* or *query* must be present.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects should be returned. If left as *None*, 10 objects are returned.



- **guid** (*str*) – Search by SSO GUID
- **query** (*str*) – Search by username substring.

`Users.update(obj)`

Update this resource.

**Variables** **obj** – a Python dictionary representing the resource to be created, in the same as returned from *get*, but one level less nested. For instance, if *get* returns `{'forum': {'name': 'Forum Name'}}`, then *obj* should be `{'name': 'New Forum'}`.

Refer to the upstream documentation for details.

### 4.30.42 Service methods

`UserVoice.instant_answers_search(page=None, per_page=None, query=None)`

Search for instant answers.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

`UserVoice.oembed(url)`

Fetch the HTML used to embed a suggestion.

**Variables** **url** (*str*) – URL to the Suggestion you want to embed (ex: `forums/1/suggestions/1`)

`UserVoice.search(page=None, per_page=None, query=None)`

Generic search for all objects.

**Variables**

- **page** (*int*) – Where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – How many objects could be returned. If left as *None*, 10 objects are returned.
- **query** (*str*) – Search string.

## 4.31 Analytics

**class** `youtube.Analytics` (*access\_token=None*)

The YouTube Analytics API currently provides a single method that lets you retrieve Analytics reports for a YouTube channel.

[https://developers.google.com/youtube/analytics/v1/available\\_reports.html](https://developers.google.com/youtube/analytics/v1/available_reports.html) identifies the different reports that you can retrieve. For each report, it lists the dimensions that are used to aggregate data, available metrics, and supported filtering options.

Create a YouTube service.

**Variables** **access\_token** –

### 4.31.1 Service methods

`Analytics.get` (*ids*, *metrics*, *start\_date*, *end\_date*, *dimensions=None*, *filters=None*, *max\_results=None*, *start\_index=None*, *sort=None*)  
Retrieve YouTube Analytics data

#### Variables

- **ids** (*str*) – Identifies the YouTube channel or content owner for which you are retrieving YouTube Analytics data. To request data for a YouTube user, set the `ids` parameter value to `channel==USER_ID`. To request data for a YouTube CMS content owner, set the `ids` parameter value to `contentOwner==OWNER_NAME`
- **metrics** (*str*) – A comma-separated list of YouTube Analytics metrics, such as views or likes, dislikes. See [https://developers.google.com/youtube/analytics/v1/available\\_reports.html](https://developers.google.com/youtube/analytics/v1/available_reports.html) for a list of the reports that you can retrieve and the metrics available in each report, and see <https://developers.google.com/youtube/analytics/v1/dimsmets/mets.html> for definitions of those metrics.
- **start\_date** (*str*) – The start date for fetching YouTube Analytics data. The value should be in YYYY-MM-DD format.
- **end\_date** (*str*) – The start date for fetching YouTube Analytics data. The value should be in YYYY-MM-DD format.
- **dimensions** (*str*) – A comma-separated list of YouTube Analytics dimensions, such as video or ageGroup, gender. See [https://developers.google.com/youtube/analytics/v1/available\\_reports.html](https://developers.google.com/youtube/analytics/v1/available_reports.html) for a list of the reports that you can retrieve and the dimensions used for those reports. Also see <https://developers.google.com/youtube/analytics/v1/dimsmets/dims> for definitions of those dimensions.
- **filters** (*str*) – A list of filters that should be applied when retrieving YouTube Analytics data. The [https://developers.google.com/youtube/analytics/v1/available\\_reports.html](https://developers.google.com/youtube/analytics/v1/available_reports.html) identifies the dimensions that can be used to filter each report, and <https://developers.google.com/youtube/analytics/v1/dimsmets/dims> defines those dimensions. If a request uses multiple filters, join them together with a semicolon (;), and the returned result table will satisfy both filters. For example, a `filters` parameter value of `video==dMH0bHeiRNg;country==IT` restricts the result set to include data for the given video in Italy.
- **max\_results** (*str*) – The maximum number of rows to include in the response
- **start\_index** (*int*) – The 1-based index of the first entity to retrieve. Use this parameter as a pagination mechanism along with the `max-results` parameter.
- **sort** (*str*) – A comma-separated list of dimensions or metrics that determine the sort order for YouTube Analytics data. By default the sort order is ascending. The ‘-’ prefix causes descending sort order

## 4.32 Zendesk

**class** `zendesk.Zendesk` (*subdomain*, *username=None*, *password=None*, *access\_token=None*)  
Create a Zendesk service.

#### Variables

- **subdomain** (*str*) – The account-specific part of the Zendesk domain, for instance use *mycompany* if your Zendesk domain is *mycompany.zendesk.com*.

- **username** (*str*) – The email of the authenticated agent. Use *user@company.com/token* for token-based authentication.
- **password** (*str*) – The password of the authenticated agent, or an API token if using token-based authentication.
- **access\_token** (*str*) – An OAuth Access token. Username and password are not required if the OAuth Access token is provided.

### 4.32.1 Activities

`Zendesk.activities()`

Return the resource corresponding to all activities.

`Activities.get(since=None, page=None, per_page=None)`

Fetch the list of activities

**Variables** `since` (*str*) – Timestamp offset in UTC on ISO8601 form *%Y-%m-%dT%H:%M:%SZ*

### 4.32.2 Activity

`Zendesk.activity(activity_id)`

Return the resource corresponding to a single activity.

`Activity.get(page=None, per_page=None)`

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

### 4.32.3 Exports

`Zendesk.exports()`

Return the resource corresponding to exports.

`Exports.sample(start_time)`

This end point is only to be used for testing the incremental export format. It is more relaxed in terms of rate limiting, but will only return up to 50 records.

**Variables** `start_time` (*int*) – The time of the oldest ticket you are interested in. Tickets modified on or since this time will be returned. The start time is provided as the number of seconds since epoch UTC.

`Exports.tickets(start_time)`

Retrieve tickets that changed in Zendesk “since last you asked”

**Variables** `start_time` (*int*) – The time of the oldest ticket you are interested in. Tickets modified on or since this time will be returned. The start time is provided as the number of seconds since epoch UTC.

#### 4.32.4 Group

`Zendesk.group (group_id)`

Return the resource corresponding to a single group.

`Group.delete ()`

Delete this resource.

`Group.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

`Group.update (obj)`

Update this resource.

**Variables** **obj** – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.32.5 Groups

`Zendesk.groups ()`

Return the resource corresponding to all groups.

`Groups.assignable (page=None, per_page=None)`

Fetch assignable groups.

`Groups.create (obj)`

Create a new resource.

**Variables** **obj** – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Groups.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

#### 4.32.6 SatisfactionRating

`Zendesk.satisfaction_rating (rating_id)`

Return the resource corresponding to a single satisfaction rating.

`SatisfactionRating.get (page=None, per_page=None)`

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

### 4.32.7 SatisfactionRatings

Zendesk.**satisfaction\_ratings** ()

Return the resource corresponding to all satisfaction ratings.

SatisfactionRatings.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

SatisfactionRatings.**received** (*page=None, per\_page=None, sort\_order=None*)

Fetch ratings provided by customers.

### 4.32.8 Tags

Zendesk.**tags** ()

Return the resource corresponding to tags.

Tags.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

### 4.32.9 Ticket

Zendesk.**ticket** (*ticket\_id*)

Return the resource corresponding to a single ticket.

### 4.32.10 Tags

Ticket.**tags** ()

Return the resource corresponding to tags.

Tags.**get** (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

#### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

`Ticket.audits` (*page=None, per\_page=None*)

Fetch the audits on a ticket.

`Ticket.collaborators` (*page=None, per\_page=None*)

Fetch the collaborators on a ticket.

`Ticket.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Ticket.delete` ()

Delete this resource.

`Ticket.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

`Ticket.metrics` ()

Fetch the ticket metrics.

`Ticket.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

### 4.32.11 Tickets

`Zendesk.tickets` ()

Return the resource corresponding to all the tickets.

`Tickets.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Tickets.delete` ()

Delete this resource.

`Tickets.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.

- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

`Tickets.recent` (*page=None, per\_page=None*)

Fetch all recent tickets. The parameters are the same as for the *get* method.

`Tickets.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

#### 4.32.12 User, CurrentUser

`Zendesk.user` (*user\_id=None*)

Return the resource corresponding to a single user. If *user\_id* is *None* the returned resource is the currently authenticated user, otherwise it is the user with the given ID number.

`User.delete` ()

Delete this resource.

`User.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

`User.tickets_ccd` (*page=None, per\_page=None*)

Fetch tickets where this user is CC'd.

`User.tickets_requested` (*page=None, per\_page=None*)

Fetch tickets requested by this user.

`User.update` (*obj*)

Update this resource.

**Variables** *obj* – a Python object representing the updated resource, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`CurrentUser.get` ()

#### 4.32.13 Users

`Zendesk.users` ()

Return the resource corresponding to all users.

`Users.create` (*obj*)

Create a new resource.

**Variables** *obj* – a Python object representing the resource to be created, usually in the same format as returned from *get*. Refer to the upstream documentation for details.

`Users.get` (*page=None, per\_page=None*)

For single-object resources, fetch the object's data. For collections, fetch all of the objects.

##### Variables

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

Users.**search** (*query*, *page=None*, *per\_page=None*)

Fetch users based on their usernames or email addresses.

**Variables** *query* (*str*) – A username or an email address.

#### 4.32.14 View

Zendesk.**view** (*view\_id*)

Return the resource corresponding to a single view.

View.**count** ()

Returns the ticket count for a single view.

View.**execute** (*sort\_by=None*, *sort\_order=None*)

Get the view output. View output sorting can be controlled by passing the *sort\_by* and *sort\_order* parameters.

**Variables**

- **sort\_by** (*str*) – The field used for sorting. This will either be a title or a custom field id.
- **sort\_order** (*str*) – The direction the tickets are sorted. May be one of ‘asc’ or ‘desc’

View.**get** (*page=None*, *per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**

- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

View.**tickets** (*page=None*, *per\_page=None*)

Returns the ticket for a single view.

#### 4.32.15 Views

Zendesk.**views** ()

Return the resource corresponding to all views.

Views.**active** (*page=None*, *per\_page=None*)

Fetch active shared and personal Views available to the current user.

Views.**count\_many** (*ids*)

Calculates the size of the view in terms of number of tickets the view will return. Only returns values for personal and shared views accessible to the user performing the request.

**Variables** *ids* (*tuple of int*) – List of view ids

Views.**get** (*page=None*, *per\_page=None*)

For single-object resources, fetch the object’s data. For collections, fetch all of the objects.

**Variables**



- **page** (*int*) – For collections, where should paging start. If left as *None*, the first page is returned.
- **per\_page** (*int*) – For collections, how many objects could be returned. The maximum is 100. If left as *None*, 100 objects are returned.

Views.**preview**(*conditions*, *columns=None*, *group\_by=None*, *group\_order=None*, *sort\_by=None*, *sort\_order=None*)

Views can be previewed by constructing the conditions in the proper format. See [http://developer.zendesk.com/documentation/rest\\_api/views.html#previewing-views](http://developer.zendesk.com/documentation/rest_api/views.html#previewing-views).

#### Variables

- **conditions** (*dict*) – A representation of the conditions that constitute the view. See [http://developer.zendesk.com/documentation/rest\\_api/views.html#conditions](http://developer.zendesk.com/documentation/rest_api/views.html#conditions).
- **columns** (*tuple of int or str*) – The ticket fields to display. System fields are looked up by name, custom fields by title or id.
- **group\_by** (*str*) – When present, the field by which the tickets are grouped
- **group\_order** (*str*) – The direction the tickets are grouped. May be one of ‘asc’ or ‘desc’
- **sort\_by** (*str*) – The field used for sorting. This will either be a title or a custom field id.
- **sort\_order** (*str*) – The direction the tickets are sorted. May be one of ‘asc’ or ‘desc’

Views.**preview\_count**(*conditions*)

Views can be previewed by constructing the conditions in the proper format. See [http://developer.zendesk.com/documentation/rest\\_api/views.html#previewing-views](http://developer.zendesk.com/documentation/rest_api/views.html#previewing-views).

#### Variables

- **conditions** (*dict*) – A representation of the conditions that constitute the view. See [http://developer.zendesk.com/documentation/rest\\_api/views.html#conditions](http://developer.zendesk.com/documentation/rest_api/views.html#conditions).
- **columns** (*tuple of int or str*) – The ticket fields to display. System fields are looked up by name, custom fields by title or id.
- **group\_by** (*str*) – When present, the field by which the tickets are grouped
- **group\_order** (*str*) – The direction the tickets are grouped. May be one of ‘asc’ or ‘desc’
- **sort\_by** (*str*) – The field used for sorting. This will either be a title or a custom field id.
- **sort\_order** (*str*) – The direction the tickets are sorted. May be one of ‘asc’ or ‘desc’

## 4.32.16 Service methods

Zendesk.**search**(*query*, *sort\_order=None*, *sort\_by=None*, *page=None*, *per\_page=None*)

Fetch the results of a search on your Zendesk account. For details on searching, see [http://developer.zendesk.com/documentation/rest\\_api/search.html](http://developer.zendesk.com/documentation/rest_api/search.html)

#### Variables

- **query** (*str*) – A free-form search term.
- **sort\_order** – Optional order in which to sort the results.
- **sort\_by** (*str*) – Optional term by which to sort the results.



## CHAPTER 5

---

### Indices and tables

---

- genindex
- search

build failing



## A

- access() (basecamp.calendars.Calendar method), 12
- access() (basecamp.projects.Project method), 16
- access\_token() (googleoauth2.service.GoogleOAuth2 method), 216
- accessed() (ducksboard.resources.Dashboard method), 57
- accesses() (basecamp.calendars.Calendar method), 12
- accesses() (basecamp.projects.Project method), 16
- account() (ducksboard.service.Ducksboard method), 56
- account() (googleanalytics.management.Management method), 206
- account() (recurly.service.Recurly method), 267
- account() (stripe.service.Stripe method), 279
- account() (twilio.service.Twilio method), 312
- accounts() (ducksboard.service.Ducksboard method), 56
- accounts() (googleanalytics.management.Management method), 208
- accounts() (recurly.service.Recurly method), 270
- accounts() (twilio.service.Twilio method), 331
- action() (desk.macros.Macro method), 51
- action() (trello.service.Trello method), 286
- actions() (desk.macros.Macro method), 51
- actions() (pingdom.service.Pingdom method), 242
- actions() (trello.boards.Board method), 288
- actions() (trello.cards.Card method), 292
- actions() (trello.lists.List method), 298
- actions() (trello.members.Member method), 299, 303
- actions() (trello.organizations.Organization method), 309
- active() (zendesk.resources.Views method), 364
- active\_users() (flurry.resource.Metrics method), 60
- active\_users\_by\_month() (flurry.resource.Metrics method), 60
- active\_users\_by\_week() (flurry.resource.Metrics method), 60
- activities() (pipedrive.deals.Deal method), 253
- activities() (pipedrive.organizations.Organization method), 257
- activities() (pipedrive.persons.Person method), 260
- activities() (pipedrive.service.Pipedrive method), 250
- activities() (pipedrive.users.User method), 265
- activities() (zendesk.service.Zendesk method), 359
- activity() (pipedrive.service.Pipedrive method), 250
- activity() (zendesk.service.Zendesk method), 359
- activity\_type() (pipedrive.service.Pipedrive method), 251
- activity\_types() (pipedrive.service.Pipedrive method), 251
- add() (bitbucket.emails.Emails method), 26
- add() (github.repos.RepoCollaborators method), 70, 85, 102, 119, 137, 153, 173, 189
- add() (github.teams.TeamMembership method), 118
- add() (github.teams.TeamRepo method), 134
- add() (github.users.UserEmails method), 170
- addon() (recurly.plans.Plan method), 273
- addons() (recurly.plans.Plan method), 274
- adjustment() (recurly.service.Recurly method), 271
- adjustments() (recurly.accounts.Account method), 267
- advertiser() (mixrank.service.MixRank method), 238
- advertisers() (mixrank.keywords.Keyword method), 239
- advertisers() (mixrank.publishers.Publisher method), 240
- alias() (segmentio.service.SegmentIO method), 278
- all\_time() (libsas.services.twilio.usage.Records method), 326
- analysis() (pingdom.checks.Check method), 242
- analysis() (pingdom.service.Pingdom method), 242
- Analytics (class in youtube), 357
- apikeyAdd() (mailchimp.service.Mailchimp method), 227
- apikeyExpire() (mailchimp.service.Mailchimp method), 227
- apikeyes() (mailchimp.service.Mailchimp method), 227
- application() (flurry.service.Flurry method), 59
- application() (twilio.accounts.Account method), 312
- applications() (flurry.service.Flurry method), 62
- applications() (twilio.accounts.Account method), 312
- archive\_all\_cards() (trello.lists.List method), 299
- archived() (basecamp.projects.Projects method), 24
- archivelink() (github.repocontents.RepoContents method), 72, 88, 104, 122, 140, 155, 175, 191

article() (desk.service.Desk method), 46  
 article() (uservoice.service.UserVoice method), 332  
 articles() (desk.contents.Topic method), 53  
 articles() (uservoice.service.UserVoice method), 333  
 articles() (uservoice.topics.Topic method), 353  
 asset() (github.releases.Release method), 83, 99, 116, 133, 151, 167, 186, 202  
 assets() (github.releases.Release method), 84, 99, 116, 133, 151, 167, 187, 203  
 assignable() (zendesk.resources.Groups method), 360  
 assigned\_todos() (basecamp.people.Person method), 15  
 attachment() (trello.cards.Card method), 292  
 attachments() (basecamp.projects.Project method), 16  
 attachments() (basecamp.service.Basecamp method), 11  
 attachments() (trello.cards.Card method), 292  
 audits() (zendesk.resources.Ticket method), 362  
 authenticated\_user() (instagram.service.Instagram method), 219  
 authorization() (github.service.GitHub method), 65  
 authorizations() (github.service.GitHub method), 65  
 authorizations() (pipedrive.service.Pipedrive method), 252  
 authorized\_connect\_app() (twilio.accounts.Account method), 313  
 authorized\_connect\_apps() (twilio.accounts.Account method), 313  
 available\_phone\_numbers() (twilio.accounts.Account method), 313  
 average() (mixpanel.resources.Segmentation method), 236  
 average() (pingdom.summary.Summary method), 243, 249  
 avg\_page\_views\_per\_session() (flurry.resource.Metrics method), 61  
 avg\_session\_length() (flurry.resource.Metrics method), 61

## B

balance\_history() (stripe.service.Stripe method), 279  
 Basecamp (class in basecamp), 11  
 batch\_import() (segmentio.service.SegmentIO method), 278  
 billing\_info() (recurly.accounts.Account method), 267  
 BitBucket (class in bitbucket), 25  
 Bitly (class in bitly), 38  
 board() (trello.actions.Action method), 286  
 board() (trello.cards.Card method), 293  
 board() (trello.checklists.Checklist method), 296  
 board() (trello.lists.List method), 298  
 board() (trello.notifications.Notification method), 307  
 board() (trello.service.Trello method), 288  
 board\_background() (trello.members.Member method), 299, 303

board\_backgrounds() (trello.members.Member method), 300, 304  
 board\_star() (trello.members.Member method), 300, 304  
 board\_stars() (trello.members.Member method), 300, 304  
 boards() (trello.members.Member method), 300, 305  
 boards() (trello.organizations.Organization method), 309  
 boards() (trello.service.Trello method), 291  
 branches() (bitbucket.repositories.Repo method), 34  
 branches() (github.organizations.OrganizationRepo method), 84, 152, 188  
 branches() (github.repos.Repo method), 100, 117, 168, 203  
 branches() (github.teams.TeamRepo method), 134  
 bursting\_phrases() (bitly.resource.RealTime method), 40

## C

calendar() (basecamp.service.Basecamp method), 12  
 calendar() (googlecalendar.service.GoogleCalendar method), 210  
 calendar\_event() (basecamp.calendars.Calendar method), 12  
 calendar\_event() (basecamp.projects.Project method), 17  
 calendar\_events() (basecamp.calendars.Calendar method), 13  
 calendar\_events() (basecamp.projects.Project method), 17  
 calendar\_key() (trello.boards.Board method), 291  
 calendar\_list() (googlecalendar.user.User method), 215  
 calendar\_lists() (googlecalendar.user.User method), 215  
 calendars() (basecamp.service.Basecamp method), 14  
 calendars() (googlecalendar.service.GoogleCalendar method), 214  
 call() (twilio.accounts.Account method), 314  
 calls() (twilio.accounts.Account method), 315  
 campaignAbuseReports() (mailchimp.service.Mailchimp method), 228  
 campaignAdvice() (mailchimp.service.Mailchimp method), 228  
 campaignAnalytics() (mailchimp.service.Mailchimp method), 228  
 campaignBounceMessage() (mailchimp.service.Mailchimp method), 228  
 campaignBounceMessages() (mailchimp.service.Mailchimp method), 228  
 campaignClickDetailAIM() (mailchimp.service.Mailchimp method), 228  
 campaignClickStats() (mailchimp.service.Mailchimp method), 228  
 campaignContent() (mailchimp.service.Mailchimp method), 228

- campaignCreate() (mailchimp.service.Mailchimp method), 228
- campaignDelete() (mailchimp.service.Mailchimp method), 228
- campaignEcommOrderAdd() (mailchimp.service.Mailchimp method), 228
- campaignEcommOrders() (mailchimp.service.Mailchimp method), 228
- campaignEepUriStats() (mailchimp.service.Mailchimp method), 228
- campaignEmailStatsAIM() (mailchimp.service.Mailchimp method), 229
- campaignGeoOpens() (mailchimp.service.Mailchimp method), 229
- campaignGeoOpensForCountry() (mailchimp.service.Mailchimp method), 229
- campaignMembers() (mailchimp.service.Mailchimp method), 229
- campaignNotOpenedAIM() (mailchimp.service.Mailchimp method), 229
- campaignOpenedAIM() (mailchimp.service.Mailchimp method), 229
- campaignPause() (mailchimp.service.Mailchimp method), 229
- campaignReplicate() (mailchimp.service.Mailchimp method), 229
- campaignResume() (mailchimp.service.Mailchimp method), 229
- campaigns() (mailchimp.service.Mailchimp method), 230
- campaignSchedule() (mailchimp.service.Mailchimp method), 229
- campaignSegmentTest() (mailchimp.service.Mailchimp method), 229
- campaignSendNow() (mailchimp.service.Mailchimp method), 229
- campaignSendTest() (mailchimp.service.Mailchimp method), 229
- campaignsForEmail() (mailchimp.service.Mailchimp method), 230
- campaignShareReport() (mailchimp.service.Mailchimp method), 230
- campaignStats() (mailchimp.service.Mailchimp method), 230
- campaignTemplateContent() (mailchimp.service.Mailchimp method), 230
- campaignUnschedule() (mailchimp.service.Mailchimp method), 230
- campaignUnsubscribes() (mailchimp.service.Mailchimp method), 230
- campaignUpdate() (mailchimp.service.Mailchimp method), 230
- cancel() (recurly.subscriptions.Subscription method), 275
- card() (trello.actions.Action method), 286
- card() (trello.boards.Board method), 288
- card() (trello.notifications.Notification method), 308
- card() (trello.service.Trello method), 291
- cards() (trello.boards.Board method), 288
- cards() (trello.checklists.Checklist method), 297
- cards() (trello.lists.List method), 298
- cards() (trello.members.Member method), 301, 305
- cards() (trello.service.Trello method), 296
- CartoDB (class in cartodb), 44
- case() (desk.service.Desk method), 46
- cases() (desk.companies.Company method), 48
- cases() (desk.customers.Customer method), 49
- cases() (desk.service.Desk method), 47
- categories() (uservoice.forums.Forum method), 336
- category() (bitly.links.Link method), 38
- category() (uservoice.forums.Forum method), 336
- cell() (googlespreadsheets.resource.Worksheet method), 217
- cells() (googlespreadsheets.resource.Worksheet method), 217
- changeset() (bitbucket.repositories.Repo method), 26
- changesets() (bitbucket.repositories.Repo method), 27
- charge() (stripe.service.Stripe method), 280
- charges() (stripe.service.Stripe method), 280
- check() (pingdom.service.Pingdom method), 242
- checkitem() (trello.cards.Checklist method), 293
- checkitem() (trello.checklists.Checklist method), 297
- checkitem\_states() (trello.cards.Card method), 293
- checkitems() (trello.cards.Checklist method), 293
- checkitems() (trello.checklists.Checklist method), 297
- checklist() (trello.cards.Card method), 293
- checklist() (trello.service.Trello method), 296
- checklists() (trello.boards.Board method), 289
- checklists() (trello.cards.Card method), 294
- checklists() (trello.service.Trello method), 298
- checks() (pingdom.service.Pingdom method), 244
- chimpChatter() (mailchimp.service.Mailchimp method), 230
- clear() (googlecalendar.calendar.Calendar method), 214
- clickrate() (bitly.resource.RealTime method), 40
- clicks() (bitly.links.Link method), 38
- clicks() (bitly.users.User method), 41
- collaborators() (github.organizations.OrganizationRepo method), 70, 137, 173
- collaborators() (github.repos.Repo method), 85, 102, 153, 189
- collaborators() (github.teams.TeamRepo method), 119
- collaborators() (zendesk.resources.Ticket method), 362
- colors() (googlecalendar.service.GoogleCalendar method), 214

- comment() (basecamp.projects.Project method), 18
- comment() (bitbucket.issues.RepoIssue method), 27
- comment() (github.gists.Gists method), 67
- comment() (github.issues.RepoIssues method), 77, 93, 110, 127, 145, 161, 181, 196
- comment() (github.repocommits.RepoCommits method), 71, 87, 103, 120, 138, 154, 174, 190
- comment() (instagram.media.Media method), 221
- comment() (uservoice.suggestions.ForumSuggestion method), 338
- comments() (basecamp.calendars.CalendarEvent method), 12, 17
- comments() (basecamp.documents.Document method), 18
- comments() (basecamp.projects.Message method), 20
- comments() (basecamp.todolists.TODO method), 21
- comments() (basecamp.uploads.Upload method), 23
- comments() (bitbucket.issues.RepoIssue method), 28
- comments() (github.issues.RepoIssue method), 75, 91, 107, 124, 142, 158, 178, 194
- comments() (github.repocommits.RepoCommit method), 70, 86, 103, 120, 138, 154, 173, 189
- comments() (github.repocommits.RepoCommits method), 71, 87, 104, 121, 139, 155, 174, 190
- comments() (instagram.media.Media method), 221
- comments() (trello.cards.Actions method), 292
- comments() (uservoice.service.UserVoice method), 334
- comments() (uservoice.suggestions.ForumSuggestion method), 339
- comments() (uservoice.users.User method), 354
- commit() (github.organizations.OrganizationRepo method), 70, 138, 173
- commit() (github.repos.Repo method), 86, 102, 154, 189
- commit() (github.teams.TeamRepo method), 120
- commits() (github.organizations.OrganizationRepo method), 71, 138, 174
- commits() (github.pullrequests.PullRequest method), 82, 98, 114, 132, 150, 166, 185, 201
- commits() (github.repos.Repo method), 86, 103, 154, 190
- commits() (github.teams.TeamRepo method), 120
- companies() (desk.service.Desk method), 48
- companies() (intercom.service.Intercom method), 224
- company() (desk.service.Desk method), 48
- company() (intercom.service.Intercom method), 225
- compare() (github.repocommits.RepoCommits method), 72, 87, 104, 121, 139, 155, 175, 191
- Compete (class in compete), 45
- completed() (basecamp.todolists.GlobalTodolists method), 24
- completed() (basecamp.todolists.Todolists method), 22
- component() (bitbucket.issues.RepoIssues method), 28
- components() (bitbucket.issues.RepoIssues method), 29
- condition\_filter() (pipedrive.service.Pipedrive method), 252
- condition\_filters() (pipedrive.service.Pipedrive method), 252
- conference() (twilio.accounts.Account method), 316
- conferences() (twilio.accounts.Account method), 317
- connect\_app() (twilio.accounts.Account method), 318
- connect\_apps() (twilio.accounts.Account method), 318
- contact() (pingdom.service.Pingdom method), 244
- contacts() (pingdom.service.Pingdom method), 245
- content() (bitly.links.Link method), 38
- contents() (github.organizations.OrganizationRepo method), 72, 140, 175
- contents() (github.repos.Repo method), 88, 104, 155, 191
- contents() (github.teams.TeamRepo method), 121
- contributors() (github.organizations.OrganizationRepo method), 85, 152, 188
- contributors() (github.repos.Repo method), 100, 117, 168, 204
- contributors() (github.teams.TeamRepo method), 134
- conversion\_rates() (pipedrive.pipelines.Pipeline method), 262
- convert\_to\_card() (trello.cards.CheckItem method), 293
- copy() (ducksboard.resources.Widget method), 58
- core() (googleanalytics.reporting.Reporting method), 209
- count() (recurly.accounts.Account method), 270
- count() (recurly.accounts.Accounts method), 270
- count() (recurly.accounts.BillingInfo method), 267
- count() (recurly.adjustments.AccountAdjustments method), 267
- count() (recurly.adjustments.Adjustment method), 271
- count() (recurly.coupons.Coupon method), 271
- count() (recurly.coupons.CouponRedemption method), 268
- count() (recurly.coupons.Coupons method), 272
- count() (recurly.invoices.AccountInvoices method), 268
- count() (recurly.invoices.Invoice method), 272
- count() (recurly.invoices.Invoices method), 273
- count() (recurly.plans.Addon method), 273
- count() (recurly.plans.Addons method), 274
- count() (recurly.plans.Plan method), 274
- count() (recurly.plans.Plans method), 274
- count() (recurly.subscriptions.AccountSubscriptions method), 269
- count() (recurly.subscriptions.Subscription method), 275
- count() (recurly.subscriptions.Subscriptions method), 276
- count() (recurly.transactions.AccountTransactions method), 269
- count() (recurly.transactions.Transaction method), 276
- count() (recurly.transactions.Transactions method), 277
- count() (zendesk.resources.View method), 364
- count\_many() (zendesk.resources.Views method), 364
- countries() (bitly.links.Link method), 38
- countries() (bitly.users.User method), 41
- counts() (intercom.service.Intercom method), 225



- coupon() (recurly.service.Recurly method), 271
- coupon() (stripe.service.Stripe method), 280
- coupons() (recurly.service.Recurly method), 272
- coupons() (stripe.service.Stripe method), 281
- create() (basecamp.attachments.GlobalAttachments method), 11
- create() (basecamp.calendars.Calendar method), 13
- create() (basecamp.calendars.CalendarEvent method), 13, 17
- create() (basecamp.calendars.CalendarEvents method), 13, 17
- create() (basecamp.calendars.CalendarEvents method), 14
- create() (basecamp.comments.Comment method), 18
- create() (basecamp.comments.Comments method), 12, 17, 18, 20, 21, 23
- create() (basecamp.documents.Document method), 18
- create() (basecamp.documents.Documents method), 19
- create() (basecamp.projects.Message method), 20
- create() (basecamp.projects.Messages method), 20
- create() (basecamp.projects.Project method), 24
- create() (basecamp.projects.Projects method), 24
- create() (basecamp.todolists.TODO method), 21
- create() (basecamp.todolists.Todolist method), 22
- create() (basecamp.todolists.Todolists method), 22
- create() (basecamp.todolists.Todos method), 21
- create() (basecamp.uploads.Upload method), 23
- create() (basecamp.uploads.Uploads method), 23
- create() (bitbucket.changesets.Changeset method), 26
- create() (bitbucket.changesets.Changesets method), 27
- create() (bitbucket.emails.Email method), 25
- create() (bitbucket.groups.Group method), 36
- create() (bitbucket.groups.GroupMember method), 35
- create() (bitbucket.groups.GroupMembers method), 36
- create() (bitbucket.groups.Groups method), 37
- create() (bitbucket.issues.IssueComment method), 27, 28
- create() (bitbucket.issues.IssueComments method), 28
- create() (bitbucket.issues.IssueComponents method), 29
- create() (bitbucket.issues.IssueMilestone method), 29
- create() (bitbucket.issues.IssueMilestones method), 30
- create() (bitbucket.issues.IssueVersion method), 30
- create() (bitbucket.issues.IssueVersions method), 30
- create() (bitbucket.issues.RepoIssue method), 28
- create() (bitbucket.issues.RepoIssues method), 31
- create() (bitbucket.links.RepoLink method), 32
- create() (bitbucket.links.RepoLinks method), 32
- create() (bitbucket.privileges.RepoPrivileges method), 33
- create() (bitbucket.repositories.Repo method), 34
- create() (bitbucket.repositories.Repos method), 35
- create() (bitbucket.services.Service method), 33
- create() (bitbucket.services.Services method), 33
- create() (bitbucket.users.User method), 37
- create() (desk.cases.Case method), 47
- create() (desk.cases.Replies method), 46
- create() (desk.cases.Reply method), 46
- create() (desk.companies.Companies method), 48
- create() (desk.companies.Company method), 49
- create() (desk.contents.Article method), 46
- create() (desk.contents.Articles method), 53
- create() (desk.contents.Topic method), 54
- create() (desk.contents.Topics method), 55
- create() (desk.contents.Translation method), 53, 54
- create() (desk.contents.Translations method), 53, 54
- create() (desk.customers.Customer method), 49
- create() (desk.customers.Customers method), 48, 50
- create() (desk.macros.Actions method), 51
- create() (desk.macros.Macro method), 52
- create() (desk.macros.Macros method), 52
- create() (ducksboard.resources.Accounts method), 56
- create() (ducksboard.resources.Dashboards method), 57
- create() (ducksboard.resources.Tokens method), 57
- create() (ducksboard.resources.Widgets method), 59
- create() (github.authorizations.Authorization method), 65
- create() (github.authorizations.Authorizations method), 65
- create() (github.downloads.Download method), 72, 88, 105, 122, 140, 156, 175, 191
- create() (github.downloads.Downloads method), 73, 89, 105, 122, 140, 156, 176, 192
- create() (github.forks.Forks method), 73, 89, 106, 123, 141, 157, 176, 192
- create() (github.gists.Gist method), 66
- create() (github.gists.GistComment method), 67
- create() (github.gists.Gists method), 67
- create() (github.hooks.RepoHook method), 73, 89, 106, 123, 141, 157, 177, 192
- create() (github.hooks.RepoHooks method), 74, 90, 106, 124, 142, 158, 177, 193
- create() (github.issues.IssueComment method), 77, 93, 110, 127, 145, 161, 181, 196
- create() (github.issues.IssueComments method), 75, 91, 107, 124, 142, 158, 178, 194
- create() (github.issues.Issues method), 68
- create() (github.issues.RepoIssue method), 77, 93, 109, 126, 144, 160, 180, 196
- create() (github.issues.RepoIssues method), 78, 94, 111, 128, 146, 162, 181, 197
- create() (github.keys.RepoKey method), 79, 94, 111, 128, 146, 162, 182, 198
- create() (github.keys.RepoKeys method), 79, 95, 111, 129, 147, 163, 182, 198
- create() (github.labels.IssueLabel method), 76, 92, 108, 125, 143, 159, 179, 195
- create() (github.labels.IssueLabels method), 76, 92, 109, 126, 144, 160, 179, 195
- create() (github.labels.MilestoneLabels method), 81, 97, 113, 130, 148, 164, 184, 200
- create() (github.labels.RepoLabel method), 80, 95, 112, 129, 147, 163, 183, 199

- create() (github.labels.RepoLabels method), 80, 96, 113, 130, 148, 164, 183, 199
- create() (github.members.UserMembership method), 171
- create() (github.members.UserMemberships method), 171
- create() (github.milestones.Milestone method), 81, 97, 114, 131, 149, 165, 184, 200
- create() (github.milestones.Milestones method), 82, 97, 114, 131, 149, 165, 185, 201
- create() (github.organizations.OrganizationRepo method), 85, 152, 188
- create() (github.pullrequests.PullRequest method), 82, 98, 114, 132, 150, 166, 185, 201
- create() (github.pullrequests.PullRequests method), 83, 99, 115, 132, 150, 166, 186, 202
- create() (github.releases.Release method), 84, 100, 116, 133, 151, 167, 187, 203
- create() (github.releases.Releases method), 84, 100, 117, 134, 152, 168, 187, 203
- create() (github.repocommits.RepoCommitsComment method), 71, 87, 103, 120, 138, 154, 174, 190
- create() (github.repocommits.RepoCommitsComments method), 70, 71, 86, 87, 103, 104, 120, 121, 138, 139, 154, 155, 173, 174, 189, 190
- create() (github.repos.Repo method), 101, 117, 168, 204
- create() (github.repos.Repos method), 118
- create() (github.teams.Team method), 135
- create() (github.teams.TeamMembership method), 118
- create() (github.teams.TeamRepo method), 134
- create() (github.teams.TeamRepos method), 135
- create() (github.teams.Teams method), 101, 169, 204, 205
- create() (github.users.CurrentUser method), 205
- create() (github.users.User method), 170
- create() (googlecalendar.acl.Acls method), 213
- create() (googlecalendar.calendar.CalendarLists method), 215
- create() (googlecalendar.calendar.Calendars method), 214
- create() (googlecalendar.event.Events method), 212
- create() (googlespreadsheets.resource.Rows method), 218
- create() (googlespreadsheets.resource.Worksheets method), 218
- create() (instagram.comments.Comments method), 221
- create() (instagram.likes.Likes method), 221
- create() (intercom.resource.Companies method), 224
- create() (intercom.resource.Events method), 225
- create() (intercom.resource.Impressions method), 226
- create() (intercom.resource.MessageThreads method), 226
- create() (intercom.resource.Users method), 227
- create() (libsas.services.twilio.applications.Applications method), 312
- create() (libsas.services.twilio.calls.Calls method), 315
- create() (libsas.services.twilio.numbers.IncomingPhoneNumbers method), 319
- create() (libsas.services.twilio.numbers.IncomingPhoneNumbersLocal method), 319
- create() (libsas.services.twilio.numbers.IncomingPhoneNumbersTollFree method), 319
- create() (libsas.services.twilio.numbers.OutgoingCallerIds method), 321
- create() (libsas.services.twilio.queues.Queues method), 322
- create() (libsas.services.twilio.sms.Messages method), 324
- create() (libsas.services.twilio.usage.Triggers method), 330
- create() (pingdom.checks.Checks method), 244
- create() (pingdom.contacts.Contacts method), 245
- create() (pingdom.reports.ReportsEmail method), 246, 247
- create() (pingdom.reports.ReportsPublic method), 246, 247
- create() (pingdom.reports.ReportsShared method), 246, 247
- create() (pipedrive.activities.Activities method), 250
- create() (pipedrive.activities.Activity method), 250
- create() (pipedrive.activities.ActivityType method), 251
- create() (pipedrive.activities.ActivityTypes method), 251
- create() (pipedrive.deals.Deal method), 253
- create() (pipedrive.deals.DealField method), 254
- create() (pipedrive.deals.DealFields method), 254
- create() (pipedrive.deals.Deals method), 254
- create() (pipedrive.deals.Products method), 253
- create() (pipedrive.goals.Goal method), 256
- create() (pipedrive.goals.Goals method), 256
- create() (pipedrive.notes.Note method), 257
- create() (pipedrive.notes.Notes method), 257
- create() (pipedrive.organizations.Organization method), 258
- create() (pipedrive.organizations.OrganizationField method), 258
- create() (pipedrive.organizations.OrganizationFields method), 259
- create() (pipedrive.organizations.Organizations method), 259
- create() (pipedrive.persons.Person method), 260
- create() (pipedrive.persons.PersonField method), 261
- create() (pipedrive.persons.PersonFields method), 261
- create() (pipedrive.persons.Persons method), 261
- create() (pipedrive.pipelines.Pipeline method), 262
- create() (pipedrive.pipelines.Pipelines method), 262
- create() (pipedrive.pipelines.Stage method), 264
- create() (pipedrive.pipelines.Stages method), 265
- create() (pipedrive.products.Product method), 263
- create() (pipedrive.products.ProductField method), 263
- create() (pipedrive.products.ProductFields method), 263
- create() (pipedrive.products.Products method), 264
- create() (pipedrive.users.User method), 265

- create() (pipedrive.users.Users method), 266
- create() (recurly.accounts.Account method), 270
- create() (recurly.accounts.Accounts method), 270
- create() (recurly.adjustments.AccountAdjustments method), 267
- create() (recurly.coupons.Coupon method), 271
- create() (recurly.coupons.CouponRedemption method), 268
- create() (recurly.coupons.Coupons method), 272
- create() (recurly.invoices.AccountInvoices method), 268
- create() (recurly.plans.Addons method), 274
- create() (recurly.plans.Plans method), 275
- create() (recurly.subscriptions.Subscriptions method), 276
- create() (recurly.transactions.Transactions method), 277
- create() (stripe.charges.Charges method), 280
- create() (stripe.coupons.Coupons method), 281
- create() (stripe.customers.Customers method), 282
- create() (stripe.invoices.InvoiceItems method), 284
- create() (stripe.invoices.Invoices method), 284
- create() (stripe.plans.Plans method), 285
- create() (stripe.tokens.Tokens method), 286
- create() (trello.boards.Boards method), 291
- create() (trello.boards.Checklists method), 289
- create() (trello.boards.Lists method), 289
- create() (trello.cards.Attachments method), 292
- create() (trello.cards.Cards method), 296
- create() (trello.cards.CheckItems method), 293
- create() (trello.cards.Checklists method), 294
- create() (trello.cards.Comments method), 292
- create() (trello.cards.Labels method), 294
- create() (trello.cards.MembersVoted method), 295
- create() (trello.cards.Stickers method), 295
- create() (trello.checklists.CheckItems method), 297
- create() (trello.checklists.Checklists method), 298
- create() (trello.lists.Lists method), 299
- create() (trello.members.BoardBackgrounds method), 299, 300, 303, 304
- create() (trello.members.BoardStars method), 300, 304
- create() (trello.members.CustomBoardBackgrounds method), 301, 305
- create() (trello.members.CustomStickers method), 302, 306
- create() (trello.organizations.Organizations method), 311
- create() (twilio.accounts.Accounts method), 331
- create() (uservoice.articles.Article method), 332
- create() (uservoice.articles.Articles method), 333
- create() (uservoice.categories.ForumCategories method), 336
- create() (uservoice.categories.ForumCategory method), 336
- create() (uservoice.comments.ForumSuggestionComment method), 338
- create() (uservoice.comments.ForumSuggestionComments method), 339
- create() (uservoice.flags.FaqFlags method), 335
- create() (uservoice.flags.SuggestionCommentFlags method), 338
- create() (uservoice.flags.SuggestionFlags method), 339
- create() (uservoice.forums.Forum method), 343
- create() (uservoice.forums.Forums method), 344
- create() (uservoice.gadgets.Gadget method), 344
- create() (uservoice.gadgets.Gadgets method), 345
- create() (uservoice.notes.ForumSuggestionNote method), 340
- create() (uservoice.notes.ForumSuggestionNotes method), 340
- create() (uservoice.suggestions.ForumSuggestion method), 340
- create() (uservoice.suggestions.ForumSuggestions method), 341
- create() (uservoice.suggestions.ForumUserSuggestions method), 342
- create() (uservoice.suggestions.Suggestion method), 347
- create() (uservoice.support\_queues.SupportQueue method), 348
- create() (uservoice.support\_queues.SupportQueues method), 349
- create() (uservoice.tickets.Ticket method), 351
- create() (uservoice.tickets.TicketMessages method), 349
- create() (uservoice.tickets.TicketNote method), 350
- create() (uservoice.tickets.TicketNotes method), 350
- create() (uservoice.tickets.Tickets method), 351
- create() (uservoice.topics.Topic method), 353
- create() (uservoice.users.User method), 355
- create() (uservoice.users.Users method), 356
- create() (zendesk.resources.Groups method), 360
- create() (zendesk.resources.Ticket method), 362
- create() (zendesk.resources.Tickets method), 362
- create() (zendesk.resources.Users method), 363
- creator() (trello.actions.Action method), 287
- creator() (trello.notifications.Notification method), 308
- credits() (pingdom.service.Pingdom method), 245
- currencies() (pipedrive.service.Pipedrive method), 252
- custom\_board\_background() (trello.members.Member method), 301, 305
- custom\_board\_backgrounds() (trello.members.Member method), 301, 305
- custom\_fields() (uservoice.service.UserVoice method), 334
- custom\_sticker() (trello.members.Member method), 302, 306
- custom\_stickers() (trello.members.Member method), 302, 306
- customer() (desk.service.Desk method), 49
- customer() (stripe.service.Stripe method), 281
- customers() (desk.companies.Company method), 48



- delete() (github.hooks.RepoHook method), 74, 89, 106, 123, 141, 157, 177, 193
- delete() (github.hooks.RepoHooks method), 74, 90, 107, 124, 142, 158, 177, 193
- delete() (github.issues.IssueComment method), 78, 93, 110, 127, 145, 161, 181, 197
- delete() (github.issues.IssueComments method), 75, 91, 107, 124, 143, 158, 178, 194
- delete() (github.issues.Issues method), 68
- delete() (github.issues.RepoIssue method), 77, 93, 109, 126, 145, 160, 180, 196
- delete() (github.issues.RepoIssues method), 78, 94, 111, 128, 146, 162, 181, 197
- delete() (github.keys.RepoKey method), 79, 95, 111, 128, 146, 162, 182, 198
- delete() (github.keys.RepoKeys method), 79, 95, 112, 129, 147, 163, 182, 198
- delete() (github.labels.IssueLabel method), 76, 92, 108, 125, 144, 159, 179, 195
- delete() (github.labels.IssueLabels method), 76, 92, 109, 126, 144, 160, 180, 195
- delete() (github.labels.MilestoneLabels method), 81, 97, 113, 130, 148, 164, 184, 200
- delete() (github.labels.RepoLabel method), 80, 96, 112, 129, 147, 163, 183, 199
- delete() (github.labels.RepoLabels method), 80, 96, 113, 130, 148, 164, 183, 199
- delete() (github.members.Member method), 68, 136, 171
- delete() (github.members.PublicMember method), 69, 137, 172
- delete() (github.members.UserMembership method), 171
- delete() (github.members.UserMemberships method), 171
- delete() (github.milestones.Milestone method), 81, 97, 114, 131, 149, 165, 184, 200
- delete() (github.milestones.Milestones method), 82, 98, 114, 131, 149, 165, 185, 201
- delete() (github.organizations.OrganizationRepo method), 85, 152, 188
- delete() (github.pullrequests.PullRequest method), 82, 98, 115, 132, 150, 166, 185, 201
- delete() (github.pullrequests.PullRequests method), 83, 99, 115, 132, 150, 166, 186, 202
- delete() (github.releases.Release method), 84, 100, 116, 133, 152, 167, 187, 203
- delete() (github.releases.ReleaseAsset method), 83, 99, 116, 133, 151, 167, 186, 202
- delete() (github.repocommits.RepoCommitsComment method), 71, 87, 103, 120, 139, 154, 174, 190
- delete() (github.repocommits.RepoCommitsComments method), 70, 71, 86, 87, 103, 104, 120, 121, 138, 139, 154, 155, 173, 174, 189, 190
- delete() (github.repos.Repo method), 101, 117, 168, 204
- delete() (github.repos.Repos method), 118
- delete() (github.teams.Team method), 135
- delete() (github.teams.TeamMembership method), 118
- delete() (github.teams.TeamRepo method), 134
- delete() (github.teams.TeamRepos method), 135
- delete() (github.users.UserRepos method), 170
- delete() (googlecalendar.acl.Acl method), 213
- delete() (googlecalendar.calendar.Calendar method), 214
- delete() (googlecalendar.calendar.CalendarList method), 215
- delete() (googlecalendar.event.Event method), 211
- delete() (googlespreadsheets.resource.Row method), 217
- delete() (googlespreadsheets.resource.Worksheet method), 218
- delete() (instagram.comments.Comment method), 221
- delete() (instagram.likes.Likes method), 222
- delete() (libsas.services.twilio.applications.Application method), 312
- delete() (libsas.services.twilio.conferences.Participant method), 316
- delete() (libsas.services.twilio.notifications.Notification method), 320
- delete() (libsas.services.twilio.numbers.IncomingPhoneNumber method), 318
- delete() (libsas.services.twilio.numbers.OutgoingCallerId method), 321
- delete() (libsas.services.twilio.queues.Queue method), 322
- delete() (libsas.services.twilio.recordings.Recording method), 323
- delete() (libsas.services.twilio.usage.Trigger method), 330
- delete() (pingdom.checks.Check method), 243
- delete() (pingdom.checks.Checks method), 244
- delete() (pingdom.contacts.Contact method), 244
- delete() (pingdom.contacts.Contacts method), 245
- delete() (pingdom.reports.ReportsEmail method), 246, 247
- delete() (pingdom.reports.ReportsPublic method), 246, 247
- delete() (pingdom.reports.ReportsShared method), 246, 248
- delete() (pipedrive.activities.Activities method), 250
- delete() (pipedrive.activities.Activity method), 250
- delete() (pipedrive.activities.ActivityType method), 251
- delete() (pipedrive.activities.ActivityTypes method), 251
- delete() (pipedrive.deals.Deal method), 253
- delete() (pipedrive.deals.DealField method), 254
- delete() (pipedrive.deals.DealFields method), 254
- delete() (pipedrive.deals.Deals method), 254
- delete() (pipedrive.deals.Products method), 253
- delete() (pipedrive.files.File method), 255
- delete() (pipedrive.files.Files method), 255
- delete() (pipedrive.filters.Filter method), 252
- delete() (pipedrive.filters.Filters method), 252

- delete() (pipedrive.goals.Goal method), 256
- delete() (pipedrive.goals.Goals method), 256
- delete() (pipedrive.notes.Note method), 257
- delete() (pipedrive.notes.Notes method), 257
- delete() (pipedrive.organizations.Organization method), 258
- delete() (pipedrive.organizations.OrganizationField method), 259
- delete() (pipedrive.organizations.OrganizationFields method), 259
- delete() (pipedrive.organizations.Organizations method), 259
- delete() (pipedrive.persons.Person method), 260
- delete() (pipedrive.persons.PersonField method), 261
- delete() (pipedrive.persons.PersonFields method), 261
- delete() (pipedrive.persons.Persons method), 261
- delete() (pipedrive.pipelines.Pipeline method), 262
- delete() (pipedrive.pipelines.Pipelines method), 262
- delete() (pipedrive.pipelines.Stage method), 265
- delete() (pipedrive.pipelines.Stages method), 265
- delete() (pipedrive.products.Product method), 263
- delete() (pipedrive.products.ProductField method), 263
- delete() (pipedrive.products.ProductFields method), 264
- delete() (pipedrive.products.Products method), 264
- delete() (recurly.accounts.Account method), 270
- delete() (recurly.accounts.BillingInfo method), 267
- delete() (recurly.adjustments.AccountAdjustments method), 267
- delete() (recurly.adjustments.Adjustment method), 271
- delete() (recurly.coupons.Coupon method), 271
- delete() (recurly.coupons.CouponRedemption method), 269
- delete() (recurly.plans.Addon method), 273
- delete() (recurly.plans.Plan method), 274
- delete() (stripe.coupons.Coupon method), 280
- delete() (stripe.customers.Customer method), 282
- delete() (stripe.customers.DiscountResource method), 281
- delete() (stripe.customers.SubscriptionResource method), 281
- delete() (stripe.invoices.InvoiceItem method), 284
- delete() (stripe.plans.Plan method), 285
- delete() (trello.actions.Action method), 287
- delete() (trello.boards.Member method), 289
- delete() (trello.cards.Attachment method), 292
- delete() (trello.cards.CheckItem method), 293
- delete() (trello.cards.Comments method), 292
- delete() (trello.cards.Label method), 294
- delete() (trello.cards.Sticker method), 295
- delete() (trello.checklists.CheckItem method), 297
- delete() (trello.checklists.Checklist method), 297
- delete() (trello.members.BoardStar method), 300, 304
- delete() (trello.members.CustomBoardBackground method), 301, 305
- delete() (trello.members.CustomSticker method), 302, 306
- delete() (trello.organizations.Member method), 310
- delete() (trello.organizations.Membership method), 311
- delete() (trello.organizations.Organization method), 311
- delete() (uservoice.articles.Article method), 332
- delete() (uservoice.articles.Articles method), 333
- delete() (uservoice.categories.ForumCategories method), 336
- delete() (uservoice.categories.ForumCategory method), 336
- delete() (uservoice.comments.Comments method), 334
- delete() (uservoice.comments.ForumSuggestionComment method), 338
- delete() (uservoice.comments.ForumSuggestionComments method), 339
- delete() (uservoice.comments.UserComments method), 354
- delete() (uservoice.custom\_fields.CustomFields method), 334
- delete() (uservoice.faqs.Faq method), 335
- delete() (uservoice.flags.FaqFlags method), 335
- delete() (uservoice.flags.SuggestionCommentFlags method), 338
- delete() (uservoice.flags.SuggestionFlags method), 339
- delete() (uservoice.forums.Forum method), 343
- delete() (uservoice.forums.Forums method), 344
- delete() (uservoice.gadgets.Gadget method), 344
- delete() (uservoice.gadgets.Gadgets method), 345
- delete() (uservoice.notes.ForumSuggestionNote method), 340
- delete() (uservoice.notes.ForumSuggestionNotes method), 340
- delete() (uservoice.notes.Notes method), 346
- delete() (uservoice.notes.UserNotes method), 354
- delete() (uservoice.suggestions.ForumSuggestion method), 341
- delete() (uservoice.suggestions.ForumSuggestions method), 342
- delete() (uservoice.suggestions.Suggestions method), 347
- delete() (uservoice.suggestions.UserSuggestions method), 355
- delete() (uservoice.support\_queues.SupportQueue method), 348
- delete() (uservoice.support\_queues.SupportQueues method), 349
- delete() (uservoice.tickets.Ticket method), 351
- delete() (uservoice.tickets.TicketMessages method), 349
- delete() (uservoice.tickets.TicketNote method), 350
- delete() (uservoice.tickets.TicketNotes method), 350
- delete() (uservoice.tickets.Tickets method), 352
- delete() (uservoice.topics.Topic method), 353
- delete() (uservoice.topics.Topics method), 353
- delete() (uservoice.users.User method), 355

delete() (uservice.users.Users method), 356  
 delete() (zendesk.resources.Group method), 360  
 delete() (zendesk.resources.Ticket method), 362  
 delete() (zendesk.resources.Tickets method), 362  
 delete() (zendesk.resources.User method), 363  
 Desk (class in desk), 45  
 destinations() (mixrank.ads.DisplayAd method), 238  
 destinations() (mixrank.ads.TextAd method), 238  
 diffstat() (bitbucket.changesets.Changeset method), 26  
 discount() (stripe.customers.Customer method), 281  
 displayad() (mixrank.advertisers.Advertiser method), 238  
 displayads() (mixrank.advertisers.Advertiser method), 239  
 displayads() (mixrank.keywords.Keyword method), 239  
 displayads() (mixrank.publishers.Publisher method), 240  
 dispute() (stripe.charges.Charge method), 280  
 document() (basecamp.projects.Project method), 18  
 documents() (basecamp.projects.Project method), 19  
 documents() (basecamp.service.Basecamp method), 14  
 download() (github.organizations.OrganizationRepo method), 72, 140, 175  
 download() (github.repos.Repo method), 88, 105, 156, 191  
 download() (github.teams.TeamRepo method), 122  
 downloads() (github.organizations.OrganizationRepo method), 73, 140, 176  
 downloads() (github.repos.Repo method), 89, 105, 156, 192  
 downloads() (github.teams.TeamRepo method), 122  
 Ducksboard (class in ducksboard), 56

## E

echo() (mixrank.service.MixRank method), 240  
 ecommOrderAdd() (mailchimp.service.Mailchimp method), 230  
 ecommOrderDel() (mailchimp.service.Mailchimp method), 230  
 ecommOrders() (mailchimp.service.Mailchimp method), 230  
 email() (bitbucket.service.BitBucket method), 25  
 email\_key() (trello.boards.Board method), 291  
 emails() (bitbucket.service.BitBucket method), 26  
 emails() (github.users.CurrentUser method), 170  
 encoders\_count() (bitly.links.Link method), 39  
 engage() (mixpanel.service.Mixpanel method), 237  
 enhanced() (fullcontact.service.Fullcontact method), 63  
 enrichment() (fullcontact.resource.Location method), 63  
 event() (flurry.resource.Application method), 59  
 event() (github.issues.RepoIssue method), 75, 91, 108, 125, 143, 159, 178, 194  
 event() (googlecalendar.calendar.Calendar method), 211  
 event() (stripe.service.Stripe method), 282  
 events() (basecamp.people.Person method), 15  
 events() (basecamp.projects.Project method), 19

events() (basecamp.service.Basecamp method), 14  
 events() (bitbucket.repositories.Repo method), 34  
 events() (bitbucket.users.User method), 37  
 events() (flurry.resource.Application method), 60  
 events() (github.issues.RepoIssue method), 76, 91, 108, 125, 143, 159, 179, 195  
 events() (github.issues.RepoIssues method), 78, 94, 110, 128, 146, 162, 181, 197  
 events() (googlecalendar.calendar.Calendar method), 212  
 events() (intercom.service.Intercom method), 225  
 events() (mixpanel.service.Mixpanel method), 235  
 events() (stripe.service.Stripe method), 283  
 execute() (zendesk.resources.View method), 364  
 export() (mixpanel.service.Mixpanel method), 237  
 exports() (zendesk.service.Zendesk method), 359

## F

faq() (uservice.service.UserVoice method), 335  
 feed() (instagram.users.AuthenticatedUser method), 219  
 field() (trello.actions.Action method), 288  
 field() (trello.actions.Board method), 286  
 field() (trello.actions.Card method), 286  
 field() (trello.actions.List method), 287  
 field() (trello.actions.Member method), 287  
 field() (trello.actions.MemberCreator method), 287  
 field() (trello.actions.Organization method), 287  
 field() (trello.boards.Board method), 291  
 field() (trello.boards.MembersInvited method), 290  
 field() (trello.boards.Organization method), 291  
 field() (trello.cards.Board method), 293  
 field() (trello.cards.Card method), 296  
 field() (trello.cards.List method), 294  
 field() (trello.checklists.Board method), 296  
 field() (trello.checklists.Checklist method), 297  
 field() (trello.lists.Board method), 298  
 field() (trello.lists.List method), 299  
 field() (trello.members.Member method), 303, 307  
 field() (trello.notifications.Board method), 307  
 field() (trello.notifications.Card method), 308  
 field() (trello.notifications.List method), 308  
 field() (trello.notifications.Member method), 308  
 field() (trello.notifications.MemberCreator method), 308  
 field() (trello.notifications.Notification method), 309  
 field() (trello.notifications.Organization method), 309  
 field() (trello.organizations.MembersInvited method), 310  
 field() (trello.organizations.Organization method), 311  
 file() (pipedrive.service.Pipedrive method), 255  
 files() (github.pullrequests.PullRequest method), 82, 98, 115, 132, 150, 166, 185, 201  
 files() (pipedrive.deals.Deal method), 253  
 files() (pipedrive.organizations.Organization method), 258  
 files() (pipedrive.persons.Person method), 260

files() (pipedrive.service.Pipedrive method), 255  
 filter() (bitbucket.issues.RepoIssues method), 31  
 filter() (trello.boards.Cards method), 288  
 filter() (trello.boards.Lists method), 289  
 filter() (trello.boards.Members method), 290  
 filter() (trello.checklists.Cards method), 297  
 filter() (trello.lists.Cards method), 298  
 filter() (trello.members.Boards method), 300, 305  
 filter() (trello.members.Cards method), 301, 305  
 filter() (trello.members.Notifications method), 302, 306  
 filter() (trello.members.Organizations method), 302, 306  
 filter() (trello.organizations.Boards method), 309  
 filter() (trello.organizations.Members method), 310  
 find() (pipedrive.deals.Deals method), 254  
 find() (pipedrive.organizations.Organizations method), 259  
 find() (pipedrive.persons.Persons method), 261  
 find() (pipedrive.products.Products method), 264  
 find() (pipedrive.users.Users method), 266  
 flags() (uservoice.comments.ForumSuggestionComment method), 338  
 flags() (uservoice.faqs.Faq method), 335  
 flags() (uservoice.suggestions.ForumSuggestion method), 339  
 Flurry (class in flurry), 59  
 folderAdd() (mailchimp.service.Mailchimp method), 230  
 folderDel() (mailchimp.service.Mailchimp method), 231  
 folders() (mailchimp.service.Mailchimp method), 231  
 folderUpdate() (mailchimp.service.Mailchimp method), 231  
 follow() (github.users.CurrentUser method), 205  
 followed\_by() (instagram.users.User method), 223  
 followers() (bitbucket.issues.RepoIssue method), 28  
 followers() (bitbucket.repositories.Repo method), 34  
 followers() (bitbucket.users.User method), 37  
 followers() (github.users.CurrentUser method), 205  
 followers() (github.users.User method), 170  
 followers() (pipedrive.deals.Deal method), 253  
 followers() (pipedrive.organizations.Organization method), 258  
 followers() (pipedrive.persons.Person method), 260  
 followers() (pipedrive.users.User method), 265  
 following() (github.users.CurrentUser method), 205  
 following() (github.users.User method), 170  
 follows() (bitbucket.users.User method), 37  
 follows() (github.users.CurrentUser method), 205  
 follows() (instagram.users.User method), 223  
 fork() (github.gists.Gist method), 66  
 forks() (github.organizations.OrganizationRepo method), 73, 141, 176  
 forks() (github.repos.Repo method), 89, 106, 157, 192  
 forks() (github.teams.TeamRepo method), 123  
 forum() (uservoice.service.UserVoice method), 336  
 forums() (uservoice.service.UserVoice method), 344

freebusy() (googlecalendar.service.GoogleCalendar method), 214  
 Fullcontact (class in fullcontact), 63  
 funnels() (mixpanel.service.Mixpanel method), 235

## G

gadget() (uservoice.service.UserVoice method), 344  
 gadgets() (uservoice.service.UserVoice method), 345  
 generateText() (mailchimp.service.Mailchimp method), 231  
 geography() (instagram.service.Instagram method), 220  
 get() (basecamp.accesses.Accesses method), 12, 16  
 get() (basecamp.attachments.Attachments method), 16  
 get() (basecamp.attachments.GlobalAttachments method), 11  
 get() (basecamp.calendars.Calendar method), 13  
 get() (basecamp.calendars.CalendarEvent method), 13, 17  
 get() (basecamp.calendars.CalendarEvents method), 13, 17  
 get() (basecamp.calendars.CalendarEvents method), 14  
 get() (basecamp.documents.Document method), 18  
 get() (basecamp.documents.Documents method), 19  
 get() (basecamp.documents.GlobalDocuments method), 14  
 get() (basecamp.events.Events method), 14, 15, 19  
 get() (basecamp.people.CurrentPerson method), 16  
 get() (basecamp.people.People method), 15  
 get() (basecamp.people.Person method), 16  
 get() (basecamp.projects.Message method), 20  
 get() (basecamp.projects.Project method), 24  
 get() (basecamp.projects.Projects method), 24  
 get() (basecamp.todolists.AssignedTodos method), 15  
 get() (basecamp.todolists.GlobalTodolists method), 24  
 get() (basecamp.todolists.TODO method), 21  
 get() (basecamp.todolists.Todolist method), 22  
 get() (basecamp.todolists.Todolists method), 22  
 get() (basecamp.topics.ProjectTopics method), 22  
 get() (basecamp.topics.Topics method), 25  
 get() (basecamp.uploads.Upload method), 23  
 get() (bitbucket.changesets.Changeset method), 26  
 get() (bitbucket.changesets.Changesets method), 27  
 get() (bitbucket.emails.Email method), 25  
 get() (bitbucket.emails.Emails method), 26  
 get() (bitbucket.groups.Group method), 36  
 get() (bitbucket.groups.GroupMember method), 35  
 get() (bitbucket.groups.GroupMembers method), 36  
 get() (bitbucket.groups.Groups method), 37  
 get() (bitbucket.issues.IssueComment method), 27, 29  
 get() (bitbucket.issues.IssueComments method), 28  
 get() (bitbucket.issues.IssueComponents method), 29  
 get() (bitbucket.issues.IssueMilestone method), 29  
 get() (bitbucket.issues.IssueMilestones method), 30  
 get() (bitbucket.issues.IssueVersion method), 30  
 get() (bitbucket.issues.IssueVersions method), 30



- get() (bitbucket.issues.RepoIssue method), 28
- get() (bitbucket.issues.RepoIssues method), 31
- get() (bitbucket.links.RepoLink method), 32
- get() (bitbucket.links.RepoLinks method), 32
- get() (bitbucket.privileges.GroupPrivileges method), 36
- get() (bitbucket.privileges.RepoPrivileges method), 33
- get() (bitbucket.repositories.Repo method), 34
- get() (bitbucket.repositories.Repos method), 35
- get() (bitbucket.services.Service method), 33
- get() (bitbucket.services.Services method), 34
- get() (bitbucket.users.User method), 37
- get() (bitly.resource.HighValue method), 38
- get() (bitly.resource.Search method), 41
- get() (compete.resource.Metric method), 45
- get() (desk.cases.Case method), 47
- get() (desk.cases.Cases method), 47–49
- get() (desk.cases.Replies method), 46
- get() (desk.cases.Reply method), 47
- get() (desk.companies.Companies method), 48
- get() (desk.companies.Company method), 49
- get() (desk.contents.Article method), 46
- get() (desk.contents.Articles method), 53
- get() (desk.contents.Topic method), 54
- get() (desk.contents.Topics method), 55
- get() (desk.contents.Translation method), 53, 54
- get() (desk.contents.Translations method), 53, 54
- get() (desk.customers.Customer method), 49
- get() (desk.customers.Customers method), 48, 50
- get() (desk.macros.Actions method), 51
- get() (desk.macros.Macro method), 52
- get() (desk.macros.Macros method), 52
- get() (desk.users.Group method), 50
- get() (desk.users.Groups method), 50
- get() (desk.users.SiteSettings method), 52
- get() (desk.users.User method), 55
- get() (desk.users.Users method), 55
- get() (ducksboard.resources.Account method), 56, 58
- get() (ducksboard.resources.Accounts method), 56
- get() (ducksboard.resources.Dashboard method), 57
- get() (ducksboard.resources.Dashboards method), 57
- get() (ducksboard.resources.Token method), 56
- get() (ducksboard.resources.Tokens method), 57
- get() (ducksboard.resources.Widget method), 58
- get() (ducksboard.resources.Widgets method), 59
- get() (flurry.resource.Application method), 62
- get() (flurry.resource.Applications method), 62
- get() (flurry.resource.Event method), 59
- get() (flurry.resource.Events method), 60
- get() (fullcontact.resource.Enhanced method), 63
- get() (fullcontact.resource.Person method), 64
- get() (github.authorizations.Authorization method), 65
- get() (github.authorizations.Authorizations method), 66
- get() (github.downloads.Download method), 72, 88, 105, 122, 140, 156, 176, 191
- get() (github.downloads.Downloads method), 73, 89, 105, 122, 141, 156, 176, 192
- get() (github.forks.Forks method), 73, 89, 106, 123, 141, 157, 176, 192
- get() (github.gists.Gist method), 66
- get() (github.gists.GistComment method), 67
- get() (github.gists.Gists method), 67
- get() (github.hooks.RepoHook method), 74, 89, 106, 123, 141, 157, 177, 193
- get() (github.hooks.RepoHooks method), 74, 90, 107, 124, 142, 158, 177, 193
- get() (github.issues.IssueComment method), 78, 93, 110, 127, 145, 161, 181, 197
- get() (github.issues.IssueComments method), 75, 91, 107, 124, 143, 158, 178, 194
- get() (github.issues.IssueEvent method), 75, 91, 108, 125, 143, 159, 178, 194
- get() (github.issues.IssueEvents method), 76, 78, 91, 94, 108, 110, 125, 128, 143, 146, 159, 162, 179, 181, 195, 197
- get() (github.issues.Issues method), 68
- get() (github.issues.RepoIssue method), 77, 93, 109, 127, 145, 160, 180, 196
- get() (github.issues.RepoIssues method), 78, 94, 111, 128, 146, 162, 181, 197
- get() (github.keys.RepoKey method), 79, 95, 111, 128, 146, 162, 182, 198
- get() (github.keys.RepoKeys method), 79, 95, 112, 129, 147, 163, 182, 198
- get() (github.labels.IssueLabel method), 76, 92, 108, 126, 144, 159, 179, 195
- get() (github.labels.IssueLabels method), 76, 92, 109, 126, 144, 160, 180, 195
- get() (github.labels.MilestoneLabels method), 81, 97, 113, 130, 149, 164, 184, 200
- get() (github.labels.RepoLabel method), 80, 96, 112, 129, 147, 163, 183, 199
- get() (github.labels.RepoLabels method), 80, 96, 113, 130, 148, 164, 183, 199
- get() (github.members.Member method), 68, 136, 171
- get() (github.members.Members method), 69, 119, 136, 172
- get() (github.members.PublicMember method), 69, 137, 172
- get() (github.members.PublicMembers method), 69, 137, 172
- get() (github.members.UserMembership method), 171
- get() (github.members.UserMemberships method), 171
- get() (github.milestones.Milestone method), 81, 97, 114, 131, 149, 165, 184, 200
- get() (github.milestones.Milestones method), 82, 98, 114, 131, 149, 165, 185, 201
- get() (github.organizations.OrganizationRepo method), 85, 153, 188

- get() (github.organizations.Organizations method), 101, 169, 205
- get() (github.pullrequests.PullRequest method), 82, 98, 115, 132, 150, 166, 185, 201
- get() (github.pullrequests.PullRequests method), 83, 99, 115, 132, 151, 166, 186, 202
- get() (github.releases.Release method), 84, 100, 116, 133, 152, 167, 187, 203
- get() (github.releases.ReleaseAsset method), 83, 99, 116, 133, 151, 167, 186, 202
- get() (github.releases.ReleaseAssets method), 84, 99, 116, 133, 151, 167, 187, 203
- get() (github.releases.Releases method), 84, 100, 117, 134, 152, 168, 187, 203
- get() (github.repocommits.RepoCommit method), 71, 86, 103, 120, 138, 154, 174, 190
- get() (github.repocommits.RepoCommits method), 72, 87, 104, 121, 139, 155, 175, 191
- get() (github.repocommits.RepoCommitsComment method), 71, 87, 103, 121, 139, 154, 174, 190
- get() (github.repocommits.RepoCommitsComments method), 70, 71, 86, 87, 103, 104, 120, 121, 138, 139, 154, 155, 173, 174, 189, 190
- get() (github.repocontents.RepoContents method), 72, 88, 104, 122, 140, 156, 175, 191
- get() (github.repos.Repo method), 101, 117, 168, 204
- get() (github.repos.RepoCollaborators method), 70, 85, 102, 119, 137, 153, 173, 189
- get() (github.repos.Repos method), 118
- get() (github.teams.Team method), 135
- get() (github.teams.TeamMembership method), 118
- get() (github.teams.TeamRepo method), 134
- get() (github.teams.TeamRepos method), 135
- get() (github.teams.Teams method), 101, 169, 204, 205
- get() (github.users.CurrentUser method), 205
- get() (github.users.User method), 170
- get() (github.users.UserEmails method), 170
- get() (github.users.UserRepos method), 170
- get() (googleanalytics.resources.Account method), 208
- get() (googleanalytics.resources.Accounts method), 208
- get() (googleanalytics.resources.Goal method), 207
- get() (googleanalytics.resources.Goals method), 207
- get() (googleanalytics.resources.Segments method), 209
- get() (googleanalytics.resources.View method), 207
- get() (googleanalytics.resources.Views method), 208
- get() (googleanalytics.resources.WebProperties method), 206
- get() (googleanalytics.resources.WebProperty method), 208
- get() (googlecalendar.acl.Acl method), 213
- get() (googlecalendar.acl.Acls method), 214
- get() (googlecalendar.calendar.Calendar method), 214
- get() (googlecalendar.calendar.CalendarList method), 215
- get() (googlecalendar.calendar.CalendarLists method), 215
- get() (googlecalendar.event.Event method), 211
- get() (googlecalendar.event.Events method), 212
- get() (googlecalendar.resource.ColorsResource method), 214
- get() (googlecalendar.user.SettingResource method), 215
- get() (googlespreadsheets.resource.Cell method), 217
- get() (googlespreadsheets.resource.Cells method), 217
- get() (googlespreadsheets.resource.Rows method), 218
- get() (googlespreadsheets.resource.Spreadsheets method), 218
- get() (googlespreadsheets.resource.Worksheet method), 218
- get() (googlespreadsheets.resource.Worksheets method), 218
- get() (instagram.comments.Comments method), 221
- get() (instagram.feed.Feed method), 219
- get() (instagram.likes.Likes method), 222
- get() (instagram.locations.Location method), 220
- get() (instagram.locations.Locations method), 221
- get() (instagram.media.LikedMedia method), 219
- get() (instagram.media.Media method), 222
- get() (instagram.media.Medias method), 222
- get() (instagram.media.PopularMedia method), 222
- get() (instagram.media.RecentMedia method), 220, 222, 223
- get() (instagram.relationships.FollowedBy method), 223
- get() (instagram.relationships.Follows method), 223
- get() (instagram.relationships.Relationship method), 224
- get() (instagram.relationships.RequestedBy method), 219
- get() (instagram.tags.Tag method), 223
- get() (instagram.tags.Tags method), 223
- get() (instagram.users.AuthenticatedUser method), 219
- get() (instagram.users.User method), 224
- get() (instagram.users.Users method), 224
- get() (intercom.resource.Companies method), 225
- get() (intercom.resource.Company method), 225
- get() (intercom.resource.Counts method), 225
- get() (intercom.resource.MessageThread method), 226
- get() (intercom.resource.MessageThreads method), 226
- get() (intercom.resource.User method), 226
- get() (intercom.resource.Users method), 227
- get() (libsas.services.twilio.applications.Application method), 312
- get() (libsas.services.twilio.applications.Applications method), 312
- get() (libsas.services.twilio.applications.AuthorizedConnectApp method), 313
- get() (libsas.services.twilio.applications.AuthorizedConnectApps method), 313
- get() (libsas.services.twilio.applications.ConnectApp method), 318
- get() (libsas.services.twilio.applications.ConnectApps method), 318

- get() (libsaas.services.twilio.calls.Call method), 315
- get() (libsaas.services.twilio.calls.Calls method), 315
- get() (libsaas.services.twilio.conferences.Conference method), 317
- get() (libsaas.services.twilio.conferences.Conferences method), 317
- get() (libsaas.services.twilio.conferences.Participant method), 316
- get() (libsaas.services.twilio.conferences.Participants method), 316
- get() (libsaas.services.twilio.notifications.Notification method), 320
- get() (libsaas.services.twilio.notifications.Notifications method), 314, 320
- get() (libsaas.services.twilio.numbers.AvailablePhoneNumbers method), 313
- get() (libsaas.services.twilio.numbers.AvailablePhoneNumbers method), 314
- get() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 318
- get() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 320
- get() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 319
- get() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 319
- get() (libsaas.services.twilio.numbers.OutgoingCallerId method), 321
- get() (libsaas.services.twilio.numbers.OutgoingCallerIds method), 321
- get() (libsaas.services.twilio.queues.Member method), 321
- get() (libsaas.services.twilio.queues.Members method), 322
- get() (libsaas.services.twilio.queues.Queue method), 322
- get() (libsaas.services.twilio.queues.Queues method), 322
- get() (libsaas.services.twilio.recordings.Recording method), 323
- get() (libsaas.services.twilio.recordings.Recordings method), 315, 323
- get() (libsaas.services.twilio.recordings.Transcription method), 325
- get() (libsaas.services.twilio.recordings.Transcriptions method), 323, 325
- get() (libsaas.services.twilio.sms.Message method), 324
- get() (libsaas.services.twilio.sms.Messages method), 324
- get() (libsaas.services.twilio.sms.ShortCode method), 325
- get() (libsaas.services.twilio.sms.ShortCodes method), 325
- get() (libsaas.services.twilio.usage.Records method), 330
- get() (libsaas.services.twilio.usage.RecordsAllTime method), 326
- get() (libsaas.services.twilio.usage.RecordsDaily method), 326
- get() (libsaas.services.twilio.usage.RecordsLastMonth method), 327
- get() (libsaas.services.twilio.usage.RecordsMonthly method), 327
- get() (libsaas.services.twilio.usage.RecordsThisMonth method), 328
- get() (libsaas.services.twilio.usage.RecordsToday method), 328
- get() (libsaas.services.twilio.usage.RecordsYearly method), 329
- get() (libsaas.services.twilio.usage.RecordsYesterday method), 329
- get() (libsaas.services.twilio.usage.Trigger method), 330
- get() (libsaas.services.twilio.usage.Triggers method), 330
- get() (mixpanel.resources.Events method), 235
- get() (mixpanel.resources.Funnels method), 235
- get() (mixpanel.resources.Properties method), 236
- get() (mixpanel.resources.Retention method), 236
- get() (mixpanel.resources.Segmentation method), 236
- get() (pingdom.actions.Actions method), 242
- get() (pingdom.analysis.Analysis method), 242
- get() (pingdom.checks.Check method), 243
- get() (pingdom.checks.Checks method), 244
- get() (pingdom.contacts.Contacts method), 245
- get() (pingdom.credit.Credits method), 245
- get() (pingdom.probes.Probes method), 245
- get() (pingdom.reference.Reference method), 245
- get() (pingdom.reports.ReportsEmail method), 246, 247
- get() (pingdom.reports.ReportsPublic method), 246, 247
- get() (pingdom.reports.ReportsShared method), 247, 248
- get() (pingdom.results.Results method), 243, 248
- get() (pingdom.servertime.Servertime method), 248
- get() (pingdom.settings.Settings method), 248
- get() (pingdom.single.Single method), 249
- get() (pingdom.traceroute.Traceroute method), 249
- get() (pipedrive.activities.Activities method), 250
- get() (pipedrive.activities.Activity method), 250
- get() (pipedrive.activities.ActivityType method), 251
- get() (pipedrive.activities.ActivityTypes method), 251
- get() (pipedrive.currencies.Currencies method), 252
- get() (pipedrive.deals.Deal method), 253
- get() (pipedrive.deals.DealField method), 254
- get() (pipedrive.deals.DealFields method), 254
- get() (pipedrive.deals.Deals method), 255
- get() (pipedrive.deals.Products method), 253
- get() (pipedrive.files.File method), 255
- get() (pipedrive.files.Files method), 255
- get() (pipedrive.filters.Filter method), 252
- get() (pipedrive.filters.Filters method), 252
- get() (pipedrive.goals.Goal method), 256
- get() (pipedrive.goals.Goals method), 256
- get() (pipedrive.notes.Note method), 257
- get() (pipedrive.notes.Notes method), 257

- get() (pipedrive.organizations.Organization method), 258
- get() (pipedrive.organizations.OrganizationField method), 259
- get() (pipedrive.organizations.OrganizationFields method), 259
- get() (pipedrive.organizations.Organizations method), 259
- get() (pipedrive.persons.Person method), 260
- get() (pipedrive.persons.PersonField method), 261
- get() (pipedrive.persons.PersonFields method), 261
- get() (pipedrive.persons.Persons method), 261
- get() (pipedrive.pipelines.Pipeline method), 262
- get() (pipedrive.pipelines.Pipelines method), 262
- get() (pipedrive.pipelines.Stage method), 265
- get() (pipedrive.pipelines.Stages method), 265
- get() (pipedrive.products.Product method), 263
- get() (pipedrive.products.ProductField method), 263
- get() (pipedrive.products.ProductFields method), 264
- get() (pipedrive.products.Products method), 264
- get() (pipedrive.users.Authorizations method), 252
- get() (pipedrive.users.User method), 266
- get() (pipedrive.users.UserConnections method), 266
- get() (pipedrive.users.Users method), 266
- get() (recurly.accounts.Account method), 270
- get() (recurly.accounts.Accounts method), 270
- get() (recurly.accounts.BillingInfo method), 267
- get() (recurly.adjustments.AccountAdjustments method), 267
- get() (recurly.adjustments.Adjustment method), 271
- get() (recurly.coupons.Coupon method), 271
- get() (recurly.coupons.CouponRedemption method), 269
- get() (recurly.coupons.Coupons method), 272
- get() (recurly.invoices.AccountInvoices method), 268
- get() (recurly.invoices.Invoice method), 272
- get() (recurly.invoices.Invoices method), 273
- get() (recurly.plans.Addon method), 273
- get() (recurly.plans.Addons method), 274
- get() (recurly.plans.Plan method), 274
- get() (recurly.plans.Plans method), 275
- get() (recurly.subscriptions.AccountSubscriptions method), 269
- get() (recurly.subscriptions.Subscription method), 275
- get() (recurly.subscriptions.Subscriptions method), 276
- get() (recurly.transactions.AccountTransactions method), 269
- get() (recurly.transactions.Transaction method), 276
- get() (recurly.transactions.Transactions method), 277
- get() (spotify.resources.Lookup method), 278
- get() (spotify.resources.Search method), 279
- get() (stripe.accounts.Account method), 279
- get() (stripe.balance\_history.BalanceHistory method), 279
- get() (stripe.charges.Charge method), 280
- get() (stripe.charges.Charges method), 280
- get() (stripe.coupons.Coupon method), 280
- get() (stripe.coupons.Coupons method), 281
- get() (stripe.customers.Customer method), 282
- get() (stripe.customers.Customers method), 282
- get() (stripe.customers.SubscriptionResource method), 281
- get() (stripe.customers.SubscriptionsResource method), 282
- get() (stripe.events.Event method), 282
- get() (stripe.events.Events method), 283
- get() (stripe.invoices.Invoice method), 283
- get() (stripe.invoices.InvoiceItem method), 284
- get() (stripe.invoices.InvoiceItems method), 284
- get() (stripe.invoices.Invoices method), 284
- get() (stripe.invoices.LineItems method), 283
- get() (stripe.plans.Plan method), 285
- get() (stripe.plans.Plans method), 285
- get() (stripe.tokens.Token method), 285
- get() (trello.actions.Action method), 288
- get() (trello.actions.Board method), 286
- get() (trello.actions.Card method), 286
- get() (trello.actions.List method), 287
- get() (trello.actions.Member method), 287
- get() (trello.actions.MemberCreator method), 287
- get() (trello.actions.Organization method), 287
- get() (trello.boards.Actions method), 288
- get() (trello.boards.Board method), 291
- get() (trello.boards.Card method), 288
- get() (trello.boards.Cards method), 288
- get() (trello.boards.Checklists method), 289
- get() (trello.boards.Lists method), 289
- get() (trello.boards.Members method), 290
- get() (trello.boards.Membership method), 290
- get() (trello.boards.Memberships method), 290
- get() (trello.boards.MembersInvited method), 290
- get() (trello.boards.Organization method), 291
- get() (trello.cards.Actions method), 292
- get() (trello.cards.Attachment method), 292
- get() (trello.cards.Attachments method), 292
- get() (trello.cards.Board method), 293
- get() (trello.cards.Card method), 296
- get() (trello.cards.CheckItemStates method), 293
- get() (trello.cards.Checklists method), 294
- get() (trello.cards.List method), 294
- get() (trello.cards.Members method), 295
- get() (trello.cards.MembersVoted method), 295
- get() (trello.cards.Sticker method), 295
- get() (trello.cards.Stickers method), 295
- get() (trello.checklists.Board method), 296
- get() (trello.checklists.Cards method), 297
- get() (trello.checklists.CheckItem method), 297
- get() (trello.checklists.CheckItems method), 297
- get() (trello.checklists.Checklist method), 297
- get() (trello.lists.Actions method), 298

- get() (trello.lists.Board method), 298
- get() (trello.lists.Cards method), 298
- get() (trello.lists.List method), 299
- get() (trello.members.Actions method), 299, 303
- get() (trello.members.BoardBackgrounds method), 300, 304
- get() (trello.members.Boards method), 301, 305
- get() (trello.members.BoardStars method), 300, 304
- get() (trello.members.Cards method), 301, 305
- get() (trello.members.CustomBoardBackground method), 301, 305
- get() (trello.members.CustomBoardBackgrounds method), 301, 305
- get() (trello.members.CustomSticker method), 302, 306
- get() (trello.members.CustomStickers method), 302, 306
- get() (trello.members.Member method), 303, 307
- get() (trello.members.Notifications method), 302, 306
- get() (trello.members.Organizations method), 302, 307
- get() (trello.members.Sessions method), 303, 307
- get() (trello.members.Tokens method), 303, 307
- get() (trello.notifications.Board method), 307
- get() (trello.notifications.Card method), 308
- get() (trello.notifications.List method), 308
- get() (trello.notifications.Member method), 309
- get() (trello.notifications.MemberCreator method), 308
- get() (trello.notifications.Notification method), 309
- get() (trello.notifications.Organization method), 309
- get() (trello.organizations.Actions method), 309
- get() (trello.organizations.Boards method), 310
- get() (trello.organizations.Members method), 310
- get() (trello.organizations.Membership method), 311
- get() (trello.organizations.Memberships method), 311
- get() (trello.organizations.MembersInvited method), 310
- get() (trello.organizations.Organization method), 311
- get() (twilio.accounts.Account method), 331
- get() (twilio.accounts.Accounts method), 331
- get() (uservoice.articles.Article method), 332
- get() (uservoice.articles.Articles method), 333
- get() (uservoice.categories.ForumCategories method), 336
- get() (uservoice.categories.ForumCategory method), 337
- get() (uservoice.comments.Comments method), 334
- get() (uservoice.comments.ForumSuggestionComment method), 338
- get() (uservoice.comments.ForumSuggestionComments method), 339
- get() (uservoice.comments.UserComments method), 354
- get() (uservoice.custom\_fields.CustomFields method), 334
- get() (uservoice.flags.FaqFlags method), 335
- get() (uservoice.flags.SuggestionCommentFlags method), 338
- get() (uservoice.flags.SuggestionFlags method), 339
- get() (uservoice.forums.Forum method), 343
- get() (uservoice.forums.Forums method), 344
- get() (uservoice.gadgets.Gadget method), 344
- get() (uservoice.gadgets.Gadgets method), 345
- get() (uservoice.notes.ForumSuggestionNote method), 340
- get() (uservoice.notes.ForumSuggestionNotes method), 340
- get() (uservoice.notes.Notes method), 346
- get() (uservoice.notes.UserNotes method), 354
- get() (uservoice.subdomains.Subdomain method), 347
- get() (uservoice.suggestions.ForumSuggestion method), 341
- get() (uservoice.suggestions.ForumSuggestions method), 342
- get() (uservoice.suggestions.ForumUserSuggestions method), 343
- get() (uservoice.suggestions.Suggestion method), 347
- get() (uservoice.suggestions.Suggestions method), 347
- get() (uservoice.suggestions.UserSuggestions method), 355
- get() (uservoice.support\_queues.SupportQueue method), 348
- get() (uservoice.support\_queues.SupportQueues method), 349
- get() (uservoice.tickets.Ticket method), 351
- get() (uservoice.tickets.TicketMessages method), 349
- get() (uservoice.tickets.TicketNote method), 350
- get() (uservoice.tickets.TicketNotes method), 350
- get() (uservoice.tickets.Tickets method), 352
- get() (uservoice.topics.Topics method), 353
- get() (uservoice.users.User method), 355
- get() (uservoice.users.Users method), 356
- get() (youtube.analytics.Analytics method), 358
- get() (zendesk.resources.Activities method), 359
- get() (zendesk.resources.Activity method), 359
- get() (zendesk.resources.CurrentUser method), 363
- get() (zendesk.resources.Group method), 360
- get() (zendesk.resources.Groups method), 360
- get() (zendesk.resources.SatisfactionRating method), 360
- get() (zendesk.resources.SatisfactionRatings method), 361
- get() (zendesk.resources.Tags method), 361
- get() (zendesk.resources.Ticket method), 362
- get() (zendesk.resources.Tickets method), 362
- get() (zendesk.resources.User method), 363
- get() (zendesk.resources.Users method), 363
- get() (zendesk.resources.View method), 364
- get() (zendesk.resources.Views method), 364
- get\_pdf() (recurly.invoices.Invoice method), 272
- get\_raw\_analysis() (pingdom.analysis.Analysis method), 242
- getAccountDetails() (mailchimp.service.Mailchimp method), 231
- gist() (github.service.GitHub method), 66

gists() (github.service.GitHub method), 67  
 GitHub (class in github), 65  
 gmonkeyActivity() (mailchimp.service.Mailchimp method), 231  
 gmonkeyAdd() (mailchimp.service.Mailchimp method), 231  
 gmonkeyDel() (mailchimp.service.Mailchimp method), 231  
 gmonkeyMembers() (mailchimp.service.Mailchimp method), 231  
 goal() (googleanalytics.resources.View method), 207  
 goal() (pipedrive.service.Pipedrive method), 256  
 goals() (googleanalytics.resources.View method), 207  
 goals() (pipedrive.service.Pipedrive method), 256  
 GoogleAnalytics (class in googleanalytics), 206  
 GoogleCalendar (class in googlecalendar), 210  
 GoogleOAuth2 (class in googleoauth2), 216  
 GoogleSpreadsheets (class in googlespreadsheets), 216  
 grant() (basecamp.accesses.Accesses method), 12, 16  
 grant() (bitbucket.privileges.GroupPrivileges method), 36  
 grant() (bitbucket.privileges.RepoPrivileges method), 33  
 group() (bitbucket.users.User method), 35  
 group() (desk.service.Desk method), 50  
 group() (zendesk.service.Zendesk method), 360  
 group\_filters() (desk.users.Group method), 50  
 group\_privileges() (bitbucket.users.User method), 36  
 groups() (bitbucket.users.User method), 37  
 groups() (desk.service.Desk method), 50  
 groups() (zendesk.service.Zendesk method), 360

## H

highvalue() (bitly.service.Bitly method), 38  
 history() (desk.cases.Case method), 47  
 hook() (github.organizations.OrganizationRepo method), 73, 141, 177  
 hook() (github.repos.Repo method), 89, 106, 157, 192  
 hook() (github.teams.TeamRepo method), 123  
 hooks() (github.organizations.OrganizationRepo method), 74, 142, 177  
 hooks() (github.repos.Repo method), 90, 106, 158, 193  
 hooks() (github.teams.TeamRepo method), 124  
 hot\_phrases() (bitly.resource.RealTime method), 40  
 hoursofday() (pingdom.summary.Summary method), 243, 249

## I

identify() (segmentio.resources.User method), 277  
 importing() (googlecalendar.event.Events method), 213  
 impressions() (intercom.service.Intercom method), 226  
 incoming\_phone\_number() (twilio.accounts.Account method), 318  
 incoming\_phone\_numbers() (twilio.accounts.Account method), 318

index\_stats() (mozscape.resources.Metadata method), 241  
 info() (bitly.links.Link method), 39  
 info() (bitly.users.User method), 42  
 inlineCss() (mailchimp.service.Mailchimp method), 231  
 insert() (newrelic.service.Insights method), 241  
 Insights (class in newrelic), 241  
 insights() (desk.service.Desk method), 51  
 Instagram (class in instagram), 219  
 instances() (googlecalendar.event.Event method), 211  
 instant\_answers\_search() (uservoice.service.UserVoice method), 357  
 Intercom (class in intercom), 224  
 invite() (bitbucket.repositories.Repo method), 34  
 invoice() (recurly.service.Recurly method), 272  
 invoice() (stripe.service.Stripe method), 283  
 invoiceitem() (stripe.service.Stripe method), 284  
 invoiceitems() (stripe.service.Stripe method), 284  
 invoices() (recurly.accounts.Account method), 268  
 invoices() (recurly.service.Recurly method), 273  
 invoices() (stripe.service.Stripe method), 284  
 is\_collaborator() (github.repos.RepoCollaborators method), 70, 86, 102, 119, 138, 153, 173, 189  
 is\_merged() (github.pullrequests.PullRequest method), 82, 98, 115, 132, 150, 166, 186, 201  
 is\_starred() (github.gists.Gist method), 66  
 issue() (bitbucket.repositories.Repo method), 27  
 issue() (github.organizations.OrganizationRepo method), 75, 142, 178  
 issue() (github.repos.Repo method), 90, 107, 158, 194  
 issue() (github.teams.TeamRepo method), 124  
 issues() (bitbucket.repositories.Repo method), 28  
 issues() (github.organizations.OrganizationRepo method), 77, 145, 180  
 issues() (github.repos.Repo method), 93, 110, 161, 196  
 issues() (github.service.GitHub method), 68  
 issues() (github.teams.TeamRepo method), 127

## K

key() (github.organizations.OrganizationRepo method), 79, 146, 182  
 key() (github.repos.Repo method), 94, 111, 162, 198  
 key() (github.teams.TeamRepo method), 128  
 keys() (github.organizations.OrganizationRepo method), 79, 147, 182  
 keys() (github.repos.Repo method), 95, 111, 163, 198  
 keys() (github.teams.TeamRepo method), 129  
 keyword() (mixrank.service.MixRank method), 239  
 keywords() (mixrank.advertisers.Advertiser method), 239

## L

label() (github.issues.RepoIssue method), 76, 92, 108, 125, 143, 159, 179, 195

- label() (github.organizations.OrganizationRepo method), 80, 147, 183
- label() (github.repos.Repo method), 95, 112, 163, 199
- label() (github.teams.TeamRepo method), 129
- label() (trello.cards.Card method), 294
- labels() (github.issues.RepoIssue method), 76, 92, 109, 126, 144, 160, 179, 195
- labels() (github.milestones.Milestone method), 81, 97, 113, 130, 148, 164, 184, 200
- labels() (github.organizations.OrganizationRepo method), 80, 148, 183
- labels() (github.repos.Repo method), 96, 112, 164, 199
- labels() (github.teams.TeamRepo method), 130
- labels() (trello.cards.Card method), 294
- language() (bitly.links.Link method), 39
- languages() (github.organizations.OrganizationRepo method), 85, 153, 188
- languages() (github.repos.Repo method), 101, 117, 169, 204
- languages() (github.teams.TeamRepo method), 135
- last() (ducksboard.datasource.Datasource method), 57
- last\_month() (libsas.services.twilio.usage.Records method), 327
- last\_update() (mozscape.resources.Metadata method), 241
- liked\_media() (instagram.users.AuthenticatedUser method), 219
- likes() (instagram.media.Media method), 221
- lines() (stripe.invoices.Invoice method), 283
- link() (bitbucket.repositories.Repo method), 32
- link() (bitly.service.Bitly method), 38
- link\_history() (bitly.users.User method), 42
- links() (bitbucket.repositories.Repo method), 32
- list() (mixpanel.resources.Funnels method), 235
- list() (trello.actions.Action method), 287
- list() (trello.cards.Card method), 294
- list() (trello.notifications.Notification method), 308
- list() (trello.service.Trello method), 298
- listAbuseReports() (mailchimp.service.Mailchimp method), 231
- listActivity() (mailchimp.service.Mailchimp method), 231
- listBatchSubscribe() (mailchimp.service.Mailchimp method), 231
- listBatchUnsubscribe() (mailchimp.service.Mailchimp method), 232
- listClients() (mailchimp.service.Mailchimp method), 232
- listGrowthHistory() (mailchimp.service.Mailchimp method), 232
- listInterestGroupAdd() (mailchimp.service.Mailchimp method), 232
- listInterestGroupDel() (mailchimp.service.Mailchimp method), 232
- listInterestGroupingAdd() (mailchimp.service.Mailchimp method), 232
- listInterestGroupingDel() (mailchimp.service.Mailchimp method), 232
- listInterestGroupings() (mailchimp.service.Mailchimp method), 232
- listInterestGroupingUpdate() (mailchimp.service.Mailchimp method), 232
- listInterestGroupUpdate() (mailchimp.service.Mailchimp method), 232
- listLocations() (mailchimp.service.Mailchimp method), 232
- listMemberActivity() (mailchimp.service.Mailchimp method), 232
- listMemberInfo() (mailchimp.service.Mailchimp method), 232
- listMembers() (mailchimp.service.Mailchimp method), 233
- listMergeVarAdd() (mailchimp.service.Mailchimp method), 233
- listMergeVarDel() (mailchimp.service.Mailchimp method), 233
- listMergeVars() (mailchimp.service.Mailchimp method), 233
- listMergeVarUpdate() (mailchimp.service.Mailchimp method), 233
- lists() (mailchimp.service.Mailchimp method), 234
- lists() (trello.boards.Board method), 289
- lists() (trello.service.Trello method), 299
- listsForEmail() (mailchimp.service.Mailchimp method), 234
- listStaticSegmentAdd() (mailchimp.service.Mailchimp method), 233
- listStaticSegmentDel() (mailchimp.service.Mailchimp method), 233
- listStaticSegmentMembersAdd() (mailchimp.service.Mailchimp method), 233
- listStaticSegmentMembersDel() (mailchimp.service.Mailchimp method), 233
- listStaticSegmentReset() (mailchimp.service.Mailchimp method), 233
- listStaticSegments() (mailchimp.service.Mailchimp method), 233
- listSubscribe() (mailchimp.service.Mailchimp method), 233
- listUnsubscribe() (mailchimp.service.Mailchimp method), 233
- listUpdateMember() (mailchimp.service.Mailchimp method), 234
- listWebhookAdd() (mailchimp.service.Mailchimp method), 234
- listWebhookDel() (mailchimp.service.Mailchimp method), 234

method), 234  
 listWebhooks() (mailchimp.service.Mailchimp method), 234  
 local() (libsaas.services.twilio.numbers.AvailablePhoneNumbers method), 313  
 local() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 319  
 location() (bitly.links.Link method), 39  
 location() (instagram.service.Instagram method), 220  
 locations() (fullcontact.service.Fullcontact method), 63  
 locations() (instagram.service.Instagram method), 221  
 lookup() (spotify.service.Spotify method), 278

## M

macro() (desk.service.Desk method), 51  
 macros() (desk.service.Desk method), 52  
 Mailchimp (class in mailchimp), 227  
 management() (googleanalytics.service.GoogleAnalytics method), 206  
 mark\_as\_read() (trello.cards.Card method), 296  
 mark\_as\_viewed() (trello.boards.Board method), 291  
 mark\_failed() (recurly.invoices.Invoice method), 272  
 mark\_successful() (recurly.invoices.Invoice method), 272  
 me() (googlecalendar.service.GoogleCalendar method), 215  
 me() (trello.service.Trello method), 299  
 media() (instagram.service.Instagram method), 221  
 median\_session\_length() (flurry.resource.Metrics method), 61  
 medias() (instagram.service.Instagram method), 222  
 member() (bitbucket.groups.Group method), 35  
 member() (github.organizations.Organizations method), 68, 136, 171  
 member() (github.teams.Team method), 118  
 member() (libsaas.services.twilio.queues.Queue method), 321  
 member() (trello.actions.Action method), 287  
 member() (trello.boards.Board method), 289  
 member() (trello.notifications.Notification method), 308  
 member() (trello.organizations.Organization method), 310  
 member() (trello.service.Trello method), 303  
 members() (bitbucket.groups.Group method), 36  
 members() (github.organizations.Organizations method), 69, 136, 172  
 members() (github.teams.Team method), 119  
 members() (libsaas.services.twilio.queues.Queue method), 322  
 members() (trello.boards.Board method), 290  
 members() (trello.cards.Card method), 295  
 members() (trello.organizations.Organization method), 310  
 members\_invited() (trello.boards.Board method), 290

members\_invited() (trello.organizations.Organization method), 310  
 members\_voted() (trello.cards.Card method), 295  
 membership() (trello.boards.Board method), 290  
 membership() (trello.organizations.Organization method), 311  
 memberships() (github.users.CurrentUser method), 170  
 memberships() (trello.boards.Board method), 290  
 memberships() (trello.organizations.Organization method), 311  
 merge() (pipedrive.organizations.Organization method), 258  
 merge() (pipedrive.persons.Person method), 260  
 merge() (pipedrive.users.User method), 266  
 message() (basecamp.projects.Project method), 19  
 message() (desk.cases.Case method), 47  
 message() (libsaas.services.twilio.sms.SMS method), 324  
 message\_thread() (intercom.service.Intercom method), 226  
 message\_threads() (intercom.service.Intercom method), 226  
 messages() (basecamp.projects.Project method), 20  
 messages() (libsaas.services.twilio.sms.SMS method), 324  
 messages() (uservoice.tickets.Ticket method), 349  
 meta() (desk.insights.Insights method), 51  
 metadata() (mozscape.service.Mozscape method), 241  
 metric() (compete.resource.Site method), 45  
 metrics() (flurry.resource.Application method), 60  
 metrics() (zendesk.resources.Ticket method), 362  
 milestone() (bitbucket.issues.RepoIssues method), 29  
 milestone() (github.organizations.OrganizationRepo method), 81, 148, 184  
 milestone() (github.repos.Repo method), 96, 113, 164, 200  
 milestone() (github.teams.TeamRepo method), 130  
 milestones() (bitbucket.issues.RepoIssues method), 30  
 milestones() (github.organizations.OrganizationRepo method), 82, 149, 185  
 milestones() (github.repos.Repo method), 97, 114, 165, 201  
 milestones() (github.teams.TeamRepo method), 131  
 Mixpanel (class in mixpanel), 235  
 MixRank (class in mixrank), 238  
 monthly() (libsaas.services.twilio.usage.Records method), 327  
 move() (googlecalendar.event.Event method), 211  
 movements() (pipedrive.pipelines.Pipeline method), 262  
 Mozscape (class in mozscape), 240  
 multiseg() (mixpanel.resources.Segmentation method), 236

## N

names() (fullcontact.service.Fullcontact method), 63



names() (mixpanel.resources.Events method), 235  
network\_history() (bitly.users.User method), 42  
new\_users() (flurry.resource.Metrics method), 61  
next\_update() (mozscape.resources.Metadata method), 241  
normalizer() (fullcontact.resource.Location method), 63  
normalizer() (fullcontact.resource.Name method), 63  
note() (pipedrive.service.Pipedrive method), 257  
note() (uservice.suggestions.ForumSuggestion method), 340  
note() (uservice.tickets.Ticket method), 350  
notes() (pipedrive.service.Pipedrive method), 257  
notes() (uservice.service.UserVoice method), 346  
notes() (uservice.suggestions.ForumSuggestion method), 340  
notes() (uservice.tickets.Ticket method), 350  
notes() (uservice.users.User method), 354  
notification() (trello.service.Trello method), 307  
notification() (twilio.accounts.Account method), 320  
notifications() (libsas.services.twilio.calls.Call method), 314  
notifications() (trello.members.Member method), 302, 306  
notifications() (twilio.accounts.Account method), 320  
numeric() (mixpanel.resources.Segmentation method), 236

## O

oembed() (uservice.service.UserVoice method), 357  
org() (github.members.UserMemberships method), 171  
org() (github.service.GitHub method), 68  
organization() (pipedrive.service.Pipedrive method), 257  
organization() (trello.actions.Action method), 287  
organization() (trello.boards.Board method), 291  
organization() (trello.notifications.Notification method), 309  
organization() (trello.service.Trello method), 309  
organization\_field() (pipedrive.service.Pipedrive method), 258  
organization\_fields() (pipedrive.service.Pipedrive method), 259  
organizations() (pipedrive.service.Pipedrive method), 259  
organizations() (trello.members.Member method), 302, 306  
organizations() (trello.service.Trello method), 311  
orgs() (github.users.CurrentUser method), 171  
orgs() (github.users.User method), 136  
outage() (pingdom.summary.Summary method), 243, 249  
outgoing\_caller\_id() (twilio.accounts.Account method), 321  
outgoing\_caller\_ids() (twilio.accounts.Account method), 321

## P

page\_views() (flurry.resource.Metrics method), 61  
parser() (fullcontact.resource.Name method), 64  
participant() (libsas.services.twilio.conferences.Conference method), 316  
participants() (libsas.services.twilio.conferences.Conference method), 316  
participants() (pipedrive.deals.Deal method), 253  
past() (basecamp.calendars.CalendarEvents method), 13, 17  
patch() (googlecalendar.acl.Acl method), 213  
patch() (googlecalendar.calendar.Calendar method), 214  
patch() (googlecalendar.calendar.CalendarList method), 215  
patch() (googlecalendar.event.Event method), 211  
pay() (stripe.invoices.Invoice method), 283  
people() (basecamp.service.Basecamp method), 15  
performance() (pingdom.summary.Summary method), 243, 249  
person() (basecamp.service.Basecamp method), 15  
person() (fullcontact.service.Fullcontact method), 64  
person() (pipedrive.service.Pipedrive method), 260  
person\_field() (pipedrive.service.Pipedrive method), 260  
person\_fields() (pipedrive.service.Pipedrive method), 261  
persons() (pipedrive.organizations.Organization method), 258  
persons() (pipedrive.service.Pipedrive method), 261  
ping() (github.hooks.RepoHook method), 74, 90, 106, 123, 141, 157, 177, 193  
ping() (mailchimp.service.Mailchimp method), 234  
Pingdom (class in pingdom), 242  
Pipedrive (class in pipedrive), 250  
pipeline() (pipedrive.service.Pipedrive method), 262  
pipelines() (pipedrive.service.Pipedrive method), 262  
plan() (recurly.service.Recurly method), 273  
plan() (stripe.service.Stripe method), 285  
plans() (recurly.service.Recurly method), 274  
plans() (stripe.service.Stripe method), 285  
popular\_links() (bitly.users.User method), 43  
popular\_media() (instagram.service.Instagram method), 222  
positions() (ducksboard.resources.Widgets method), 59  
postpone() (recurly.subscriptions.Subscription method), 275  
preference() (desk.users.User method), 55  
preferences() (desk.users.User method), 55  
preview() (zendesk.resources.Views method), 365  
preview\_count() (zendesk.resources.Views method), 365  
primary() (bitbucket.emails.Email method), 25  
private() (uservice.streams.Stream method), 337, 346  
privileges() (bitbucket.repositories.Repo method), 33  
probes() (pingdom.service.Pingdom method), 245  
probes() (pingdom.summary.Summary method), 243, 249  
product() (pipedrive.service.Pipedrive method), 263

- product\_field() (pipedrive.service.Pipedrive method), 263
  - product\_fields() (pipedrive.service.Pipedrive method), 263
  - products() (pipedrive.deals.Deal method), 253
  - products() (pipedrive.persons.Person method), 260
  - products() (pipedrive.service.Pipedrive method), 264
  - project() (basecamp.service.Basecamp method), 16
  - projects() (basecamp.service.Basecamp method), 24
  - properties() (mixpanel.service.Mixpanel method), 236
  - public() (github.gists.Gists method), 67
  - public() (uservoice.custom\_fields.CustomFields method), 334
  - public() (uservoice.streams.Stream method), 337, 346
  - public\_member() (github.organizations.Organizations method), 69, 137, 172
  - public\_members() (github.organizations.Organizations method), 69, 137, 172
  - publicize() (github.members.PublicMember method), 69, 137, 172
  - publisher() (mixrank.service.MixRank method), 240
  - publishers() (mixrank.ads.DisplayAd method), 238
  - publishers() (mixrank.ads.TextAd method), 238
  - publishers() (mixrank.advertisers.Advertiser method), 239
  - pullrequest() (github.organizations.OrganizationRepo method), 82, 150, 185
  - pullrequest() (github.repos.Repo method), 98, 114, 166, 201
  - pullrequest() (github.teams.TeamRepo method), 132
  - pullrequests() (github.organizations.OrganizationRepo method), 83, 150, 186
  - pullrequests() (github.repos.Repo method), 99, 115, 166, 202
  - pullrequests() (github.teams.TeamRepo method), 132
  - push() (ducksboard.datasources.Datasource method), 57
- ## Q
- query() (googlecalendar.resource.FreeBusyResource method), 214
  - query() (newrelic.service.Insights method), 241
  - queue() (twilio.accounts.Account method), 321
  - queues() (twilio.accounts.Account method), 322
  - quick\_add() (googlecalendar.event.Events method), 213
- ## R
- reactivate() (recurly.subscriptions.Subscription method), 275
  - readme() (github.repocontents.RepoContents method), 72, 88, 105, 122, 140, 156, 175, 191
  - realtime() (bitly.service.Bitly method), 40
  - realtime() (googleanalytics.reporting.Reporting method), 210
  - received() (zendesk.resources.SatisfactionRatings method), 361
  - recent() (zendesk.resources.Tickets method), 363
  - recent\_media() (instagram.geographies.Geography method), 220
  - recent\_media() (instagram.locations.Location method), 220
  - recent\_media() (instagram.tags.Tag method), 222
  - recent\_media() (instagram.users.User method), 223
  - recording() (twilio.accounts.Account method), 323
  - recordings() (libsaas.services.twilio.calls.Call method), 315
  - recordings() (twilio.accounts.Account method), 323
  - records() (libsaas.services.twilio.usage.Usage method), 326
  - Recurly (class in recurly), 267
  - redemption() (recurly.accounts.Account method), 268
  - reference() (pingdom.service.Pingdom method), 245
  - referrers() (bitly.links.Link method), 39
  - referrers() (bitly.users.User method), 43
  - referrers\_by\_domain() (bitly.links.Link method), 39
  - referring\_domains() (bitly.links.Link method), 39
  - referring\_domains() (bitly.users.User method), 43
  - refresh\_token() (googleoauth2.service.GoogleOAuth2 method), 216
  - refund() (recurly.transactions.Transaction method), 276
  - refund() (stripe.charges.Charge method), 280
  - relationship() (instagram.users.User method), 224
  - release() (github.organizations.OrganizationRepo method), 83, 151, 186
  - release() (github.repos.Repo method), 99, 115, 167, 202
  - release() (github.teams.TeamRepo method), 133
  - releases() (github.organizations.OrganizationRepo method), 84, 152, 187
  - releases() (github.repos.Repo method), 100, 117, 168, 203
  - releases() (github.teams.TeamRepo method), 134
  - remove() (github.repos.RepoCollaborators method), 70, 86, 102, 120, 138, 153, 173, 189
  - remove() (github.users.UserEmails method), 170
  - replace() (github.labels.IssueLabels method), 77, 93, 109, 126, 144, 160, 180, 196
  - replies() (desk.cases.Case method), 46
  - reply() (desk.cases.Case method), 46
  - reply() (intercom.resource.MessageThreads method), 226
  - repo() (bitbucket.service.BitBucket method), 26
  - repo() (github.organizations.Organizations method), 69, 137, 173
  - repo() (github.service.GitHub method), 102
  - repo() (github.teams.Team method), 119
  - report() (desk.insights.Insights method), 51
  - report\_email() (pingdom.service.Pingdom method), 246
  - report\_public() (pingdom.service.Pingdom method), 246
  - report\_shared() (pingdom.service.Pingdom method), 246
  - reporting() (googleanalytics.service.GoogleAnalytics method), 209
  - reports\_email() (pingdom.service.Pingdom method), 247

- reports\_public() (pingdom.service.Pingdom method), 247
- reports\_shared() (pingdom.service.Pingdom method), 247
- repos() (bitbucket.service.BitBucket method), 35
- repos() (github.organizations.Organizations method), 85, 153, 188
- repos() (github.service.GitHub method), 118
- repos() (github.teams.Team method), 135
- repos() (github.users.User method), 169
- requested\_by() (instagram.users.AuthenticatedUser method), 219
- respond() (uservoice.suggestions.ForumSuggestion method), 341
- results() (pingdom.checks.Check method), 243
- results() (pingdom.service.Pingdom method), 248
- results() (pipedrive.goals.Goal method), 256
- retained\_users() (flurry.resource.Metrics method), 62
- retention() (mixpanel.service.Mixpanel method), 236
- revoke() (basecamp.accesses.Access method), 12, 16
- revoke() (bitbucket.privileges.GroupPrivileges method), 36
- revoke() (bitbucket.privileges.RepoPrivileges method), 33
- row() (googlespreadsheets.resource.Worksheet method), 217
- rows() (googlespreadsheets.resource.Worksheet method), 218
- rule() (googlecalendar.calendar.Calendar method), 213
- rules() (googlecalendar.calendar.Calendar method), 213
- ## S
- sample() (zendesk.resources.Exports method), 359
- satisfaction\_rating() (zendesk.service.Zendesk method), 360
- satisfaction\_ratings() (zendesk.service.Zendesk method), 361
- search() (bitbucket.issues.RepoIssues method), 32
- search() (bitbucket.repositories.Repos method), 35
- search() (bitly.service.Bitly method), 41
- search() (desk.cases.Cases method), 47–49
- search() (desk.companies.Companies method), 48
- search() (desk.contents.Articles method), 53
- search() (desk.customers.Customers method), 49, 50
- search() (pipedrive.service.Pipedrive method), 266
- search() (spotify.service.Spotify method), 279
- search() (uservoice.articles.Articles method), 333
- search() (uservoice.service.UserVoice method), 357
- search() (uservoice.suggestions.ForumSuggestions method), 342
- search() (uservoice.suggestions.Suggestions method), 348
- search() (uservoice.tickets.Tickets method), 352
- search() (uservoice.topics.Topic method), 353
- search() (uservoice.users.Users method), 356
- search() (zendesk.resources.Users method), 364
- search() (zendesk.service.Zendesk method), 365
- segmentation() (mixpanel.service.Mixpanel method), 236
- SegmentIO (class in segmentio), 277
- segments() (googleanalytics.management.Management method), 209
- servertime() (pingdom.service.Pingdom method), 248
- service() (bitbucket.repositories.Repo method), 33
- services() (bitbucket.repositories.Repo method), 33
- sessions() (flurry.resource.Metrics method), 62
- sessions() (trello.members.Member method), 303, 307
- setting() (googlecalendar.user.User method), 215
- settings() (googlecalendar.user.User method), 215
- settings() (pingdom.service.Pingdom method), 248
- settings() (pipedrive.service.Pipedrive method), 266
- share\_counts() (bitly.users.User method), 44
- shares() (bitly.links.Link method), 40
- short\_code() (libsaas.services.twilio.sms.SMS method), 325
- short\_codes() (libsaas.services.twilio.sms.SMS method), 325
- similarity() (fullcontact.resource.Name method), 64
- since() (ducksboard.datasource.Datasource method), 58
- single() (pingdom.service.Pingdom method), 249
- site() (compete.service.Compete method), 45
- site\_settings() (desk.service.Desk method), 52
- sms() (twilio.accounts.Account method), 324
- social() (bitly.links.Link method), 40
- sort() (uservoice.support\_queues.SupportQueues method), 349
- Spotify (class in spotify), 278
- spreadsheet() (googlespreadsheets.service.GoogleSpreadsheets method), 216
- spreadsheets() (googlespreadsheets.service.GoogleSpreadsheets method), 218
- sql() (cartodb.service.CartoDB method), 44
- stage() (pipedrive.service.Pipedrive method), 264
- stages() (pipedrive.service.Pipedrive method), 265
- star() (github.gists.Gist method), 66
- starred() (github.gists.Gists method), 67
- stats() (fullcontact.resource.Name method), 64
- sticker() (trello.cards.Card method), 295
- stickers() (trello.cards.Card method), 295
- stream() (uservoice.forums.Forum method), 337
- stream() (uservoice.service.UserVoice method), 346
- Stripe (class in stripe), 279
- subdomain() (uservoice.service.UserVoice method), 347
- subscription() (recurly.service.Recurly method), 275
- subscription() (stripe.customers.Customer method), 281
- subscriptions() (recurly.accounts.Account method), 269
- subscriptions() (recurly.service.Recurly method), 276
- subscriptions() (stripe.customers.Customer method), 282

suggestion() (uservoice.forums.Forum method), 338  
 suggestion() (uservoice.service.UserVoice method), 347  
 suggestions() (uservoice.forums.Forum method), 341  
 suggestions() (uservoice.service.UserVoice method), 347  
 suggestions() (uservoice.users.User method), 355  
 sum() (mixpanel.resources.Segmentation method), 237  
 summary() (mixrank.advertisers.Advertiser method), 239  
 summary() (mixrank.keywords.Keyword method), 239  
 summary() (mixrank.publishers.Publisher method), 240  
 summary() (pingdom.checks.Check method), 243  
 summary() (pingdom.service.Pingdom method), 249  
 support\_queue() (uservoice.service.UserVoice method), 348  
 support\_queues() (uservoice.service.UserVoice method), 349  
 supporters() (uservoice.suggestions.ForumSuggestion method), 341

## T

tag() (instagram.service.Instagram method), 222  
 tags() (bitbucket.repositories.Repo method), 34  
 tags() (github.organizations.OrganizationRepo method), 85, 153, 188  
 tags() (github.repos.Repo method), 101, 117, 169, 204  
 tags() (github.teams.TeamRepo method), 135  
 tags() (instagram.service.Instagram method), 223  
 tags() (zendesk.resources.Ticket method), 361  
 tags() (zendesk.service.Zendesk method), 361  
 team() (github.service.GitHub method), 118  
 teams() (github.organizations.OrganizationRepo method), 85, 153, 188  
 teams() (github.organizations.Organizations method), 101, 169, 204  
 teams() (github.repos.Repo method), 101, 117, 169, 204  
 teams() (github.teams.TeamRepo method), 135  
 teams() (github.users.CurrentUser method), 205  
 templateAdd() (mailchimp.service.Mailchimp method), 234  
 templateDel() (mailchimp.service.Mailchimp method), 234  
 templateInfo() (mailchimp.service.Mailchimp method), 234  
 templates() (mailchimp.service.Mailchimp method), 234  
 templateUndel() (mailchimp.service.Mailchimp method), 234  
 templateUpdate() (mailchimp.service.Mailchimp method), 234  
 terminate() (recurly.subscriptions.Subscription method), 275  
 test() (github.hooks.RepoHook method), 74, 90, 106, 123, 142, 157, 177, 193  
 textad() (mixrank.advertisers.Advertiser method), 238  
 textads() (mixrank.advertisers.Advertiser method), 239  
 textads() (mixrank.keywords.Keyword method), 239

textads() (mixrank.publishers.Publisher method), 240  
 this\_month() (libsaas.services.twilio.usage.Records method), 328  
 ticket() (uservoice.service.UserVoice method), 349  
 ticket() (zendesk.service.Zendesk method), 361  
 tickets() (uservoice.service.UserVoice method), 351  
 tickets() (zendesk.resources.Exports method), 359  
 tickets() (zendesk.resources.View method), 364  
 tickets() (zendesk.service.Zendesk method), 362  
 tickets\_ccd() (zendesk.resources.User method), 363  
 tickets\_requested() (zendesk.resources.User method), 363  
 timeline() (pipedrive.deals.Deals method), 255  
 timespan() (ducksboard.datasource.Datasource method), 58  
 today() (libsaas.services.twilio.usage.Records method), 328  
 todo() (basecamp.projects.Project method), 21  
 todolist() (basecamp.projects.Project method), 21  
 todolists() (basecamp.projects.Project method), 22  
 todolists() (basecamp.service.Basecamp method), 24  
 todos() (basecamp.todolists.Todolist method), 21  
 token() (ducksboard.resources.Dashboard method), 56  
 token() (stripe.service.Stripe method), 285  
 tokens() (ducksboard.resources.Dashboard method), 57  
 tokens() (stripe.service.Stripe method), 286  
 tokens() (trello.members.Member method), 303, 307  
 toll\_free() (libsaas.services.twilio.numbers.AvailablePhoneNumbers method), 314  
 toll\_free() (libsaas.services.twilio.numbers.IncomingPhoneNumbers method), 319  
 top() (mixpanel.resources.Events method), 235  
 top() (mixpanel.resources.Properties method), 236  
 topic() (desk.service.Desk method), 52  
 topic() (uservoice.service.UserVoice method), 353  
 topics() (basecamp.projects.Project method), 22  
 topics() (basecamp.service.Basecamp method), 25  
 topics() (desk.service.Desk method), 55  
 topics() (uservoice.service.UserVoice method), 353  
 traceroute() (pingdom.service.Pingdom method), 249  
 track() (mixpanel.service.Mixpanel method), 237  
 track() (segmentio.resources.User method), 277  
 tracking\_domain\_list() (bitly.users.User method), 44  
 transaction() (recurly.service.Recurly method), 276  
 transactions() (recurly.accounts.Account method), 269  
 transactions() (recurly.service.Recurly method), 277  
 transcription() (twilio.accounts.Account method), 325  
 transcriptions() (libsaas.services.twilio.recordings.Recording method), 323  
 transcriptions() (twilio.accounts.Account method), 325  
 translation() (desk.contents.Articles method), 53  
 translation() (desk.contents.Topic method), 54  
 translations() (desk.contents.Articles method), 53  
 translations() (desk.contents.Topic method), 54  
 Trello (class in trello), 286

- trigger() (libsaas.services.twilio.usage.Usage method), 330
- triggers() (libsaas.services.twilio.usage.Usage method), 330
- Twilio (class in twilio), 312
- ## U
- unfollow() (github.users.CurrentUser method), 206
- unstar() (github.gists.Gist method), 66
- upcoming() (stripe.invoices.Invoices method), 285
- update() (basecamp.attachments.Attachments method), 16
- update() (basecamp.attachments.GlobalAttachments method), 11
- update() (basecamp.calendars.Calendar method), 13
- update() (basecamp.calendars.CalendarEvent method), 13, 17
- update() (basecamp.calendars.CalendarEvents method), 13, 18
- update() (basecamp.calendars.CalendarEvents method), 14
- update() (basecamp.comments.Comments method), 12, 17, 18, 20, 21, 23
- update() (basecamp.documents.Document method), 18
- update() (basecamp.documents.Documents method), 19
- update() (basecamp.documents.GlobalDocuments method), 14
- update() (basecamp.events.Events method), 14, 15, 19
- update() (basecamp.projects.Message method), 20
- update() (basecamp.projects.Messages method), 20
- update() (basecamp.projects.Project method), 24
- update() (basecamp.projects.Projects method), 24
- update() (basecamp.todolists.AssignedTodos method), 15
- update() (basecamp.todolists.GlobalTodolists method), 25
- update() (basecamp.todolists.TODO method), 21
- update() (basecamp.todolists.Todolist method), 22
- update() (basecamp.todolists.Todolists method), 22
- update() (basecamp.todolists.Todos method), 22
- update() (basecamp.topics.ProjectTopics method), 23
- update() (basecamp.topics.Topics method), 25
- update() (basecamp.uploads.Uploads method), 23
- update() (bitbucket.changesets.Changeset method), 26
- update() (bitbucket.changesets.Changesets method), 27
- update() (bitbucket.emails.Email method), 25
- update() (bitbucket.emails.Emails method), 26
- update() (bitbucket.groups.Group method), 36
- update() (bitbucket.groups.GroupMember method), 35
- update() (bitbucket.groups.GroupMembers method), 36
- update() (bitbucket.groups.Groups method), 37
- update() (bitbucket.issues.IssueComment method), 27, 29
- update() (bitbucket.issues.IssueComments method), 28
- update() (bitbucket.issues.IssueComponents method), 29
- update() (bitbucket.issues.IssueMilestone method), 29
- update() (bitbucket.issues.IssueMilestones method), 30
- update() (bitbucket.issues.IssueVersion method), 30
- update() (bitbucket.issues.IssueVersions method), 31
- update() (bitbucket.issues.RepoIssue method), 28
- update() (bitbucket.issues.RepoIssues method), 32
- update() (bitbucket.links.RepoLink method), 32
- update() (bitbucket.links.RepoLinks method), 32
- update() (bitbucket.privileges.RepoPrivileges method), 33
- update() (bitbucket.repositories.Repo method), 34
- update() (bitbucket.repositories.Repos method), 35
- update() (bitbucket.services.Service method), 33
- update() (bitbucket.services.Services method), 34
- update() (bitbucket.users.User method), 37
- update() (desk.cases.Case method), 47
- update() (desk.cases.Replies method), 46
- update() (desk.cases.Reply method), 47
- update() (desk.companies.Company method), 49
- update() (desk.contents.Article method), 46
- update() (desk.contents.Articles method), 54
- update() (desk.contents.Topic method), 55
- update() (desk.contents.Translation method), 53, 54
- update() (desk.contents.Translations method), 53, 54
- update() (desk.customers.Customer method), 50
- update() (desk.macros.Actions method), 51, 52
- update() (desk.macros.Macro method), 52
- update() (ducksboard.resources.Dashboard method), 57
- update() (ducksboard.resources.Widget method), 58
- update() (github.authorizations.Authorization method), 65
- update() (github.authorizations.Authorizations method), 66
- update() (github.gists.Gist method), 66
- update() (github.gists.GistComment method), 67
- update() (github.gists.Gists method), 67
- update() (github.hooks.RepoHook method), 74, 90, 106, 123, 142, 157, 177, 193
- update() (github.hooks.RepoHooks method), 74, 90, 107, 124, 142, 158, 177, 193
- update() (github.issues.IssueComment method), 78, 94, 110, 127, 145, 161, 181, 197
- update() (github.issues.IssueComments method), 75, 91, 107, 125, 143, 159, 178, 194
- update() (github.issues.Issues method), 68
- update() (github.issues.RepoIssue method), 77, 93, 109, 127, 145, 161, 180, 196
- update() (github.issues.RepoIssues method), 78, 94, 111, 128, 146, 162, 182, 197
- update() (github.keys.RepoKey method), 79, 95, 111, 128, 147, 162, 182, 198
- update() (github.keys.RepoKeys method), 79, 95, 112, 129, 147, 163, 183, 198
- update() (github.labels.IssueLabel method), 76, 92, 108, 126, 144, 160, 179, 195

- update() (github.labels.IssueLabels method), 77, 93, 109, 126, 144, 160, 180, 196
- update() (github.labels.MilestoneLabels method), 81, 97, 113, 131, 149, 165, 184, 200
- update() (github.labels.RepoLabel method), 80, 96, 112, 130, 148, 164, 183, 199
- update() (github.labels.RepoLabels method), 80, 96, 113, 130, 148, 164, 184, 199
- update() (github.members.UserMembership method), 171
- update() (github.members.UserMemberships method), 171
- update() (github.milestones.Milestone method), 81, 97, 114, 131, 149, 165, 185, 200
- update() (github.milestones.Milestones method), 82, 98, 114, 131, 150, 165, 185, 201
- update() (github.organizations.OrganizationRepo method), 85, 153, 188
- update() (github.organizations.Organizations method), 102, 169, 205
- update() (github.pullrequests.PullRequest method), 82, 98, 115, 132, 150, 166, 186, 201
- update() (github.pullrequests.PullRequests method), 83, 99, 115, 133, 151, 167, 186, 202
- update() (github.releases.Release method), 84, 100, 116, 134, 152, 168, 187, 203
- update() (github.releases.ReleaseAsset method), 83, 99, 116, 133, 151, 167, 187, 202
- update() (github.repocommits.RepoCommitsComment method), 71, 87, 103, 121, 139, 155, 174, 190
- update() (github.repocommits.RepoCommitsComments method), 70, 71, 86, 87, 103, 104, 120, 121, 138, 139, 154, 155, 174, 175, 189, 190
- update() (github.repos.Repo method), 101, 117, 169, 204
- update() (github.repos.Repos method), 118
- update() (github.teams.Team method), 136
- update() (github.teams.TeamMembership method), 119
- update() (github.teams.TeamRepo method), 135
- update() (github.teams.TeamRepos method), 135
- update() (github.users.CurrentUser method), 206
- update() (github.users.UserRepos method), 170
- update() (googlecalendar.acl.Acl method), 213
- update() (googlecalendar.calendar.Calendar method), 214
- update() (googlecalendar.calendar.CalendarList method), 215
- update() (googlecalendar.event.Event method), 212
- update() (googlespreadsheets.resource.Cell method), 217
- update() (googlespreadsheets.resource.Row method), 217
- update() (googlespreadsheets.resource.Worksheet method), 218
- update() (instagram.relationships.Relationship method), 224
- update() (intercom.resource.Users method), 227
- update() (libsaas.services.twilio.applications.Application method), 312
- update() (libsaas.services.twilio.applications.ConnectApp method), 318
- update() (libsaas.services.twilio.calls.Call method), 315
- update() (libsaas.services.twilio.conferences.Participant method), 316
- update() (libsaas.services.twilio.numbers.IncomingPhoneNumber method), 318
- update() (libsaas.services.twilio.numbers.OutgoingCallerId method), 321
- update() (libsaas.services.twilio.queues.Member method), 322
- update() (libsaas.services.twilio.queues.Queue method), 322
- update() (libsaas.services.twilio.sms.ShortCode method), 325
- update() (libsaas.services.twilio.usage.Trigger method), 330
- update() (pingdom.checks.Check method), 244
- update() (pingdom.checks.Checks method), 244
- update() (pingdom.contacts.Contact method), 244
- update() (pingdom.contacts.Contacts method), 245
- update() (pingdom.reports.ReportsEmail method), 246, 247
- update() (pingdom.reports.ReportsPublic method), 246, 247
- update() (pingdom.reports.ReportsShared method), 247, 248
- update() (pingdom.settings.Settings method), 248
- update() (pipedrive.activities.Activities method), 250
- update() (pipedrive.activities.Activity method), 250
- update() (pipedrive.activities.ActivityType method), 251
- update() (pipedrive.activities.ActivityTypes method), 251
- update() (pipedrive.deals.Deal method), 253
- update() (pipedrive.deals.Deals method), 255
- update() (pipedrive.deals.Products method), 253
- update() (pipedrive.files.File method), 255
- update() (pipedrive.files.Files method), 255
- update() (pipedrive.goals.Goal method), 256
- update() (pipedrive.goals.Goals method), 256
- update() (pipedrive.notes.Note method), 257
- update() (pipedrive.notes.Notes method), 257
- update() (pipedrive.organizations.Organization method), 258
- update() (pipedrive.organizations.Organizations method), 259
- update() (pipedrive.persons.Person method), 260
- update() (pipedrive.persons.Persons method), 261
- update() (pipedrive.pipelines.Pipeline method), 262
- update() (pipedrive.pipelines.Pipelines method), 262
- update() (pipedrive.pipelines.Stage method), 265
- update() (pipedrive.pipelines.Stages method), 265
- update() (pipedrive.products.Product method), 263
- update() (pipedrive.products.Products method), 264
- update() (recurly.accounts.Account method), 270

- update() (recurly.accounts.BillingInfo method), 268
- update() (recurly.plans.Addon method), 273
- update() (recurly.plans.Plan method), 274
- update() (recurly.subscriptions.Subscription method), 276
- update() (stripe.customers.Customer method), 282
- update() (stripe.customers.SubscriptionResource method), 281
- update() (stripe.invoices.Invoice method), 283
- update() (stripe.plans.Plan method), 285
- update() (trello.actions.Action method), 288
- update() (trello.boards.Board method), 291
- update() (trello.boards.Member method), 289
- update() (trello.boards.Membership method), 290
- update() (trello.cards.Card method), 296
- update() (trello.cards.CheckItem method), 293
- update() (trello.cards.Comments method), 292
- update() (trello.cards.Sticker method), 295
- update() (trello.checklists.Checklist method), 297
- update() (trello.lists.List method), 299
- update() (trello.members.BoardStar method), 300, 304
- update() (trello.members.CustomBoardBackground method), 301, 305
- update() (trello.members.Member method), 303, 307
- update() (trello.notifications.Notification method), 309
- update() (trello.organizations.Member method), 310
- update() (trello.organizations.Membership method), 311
- update() (trello.organizations.Organization method), 311
- update() (twilio.accounts.Account method), 331
- update() (uservoice.articles.Article method), 332
- update() (uservoice.articles.Articles method), 333
- update() (uservoice.categories.ForumCategories method), 336
- update() (uservoice.categories.ForumCategory method), 337
- update() (uservoice.comments.Comments method), 334
- update() (uservoice.comments.ForumSuggestionComment method), 338
- update() (uservoice.comments.ForumSuggestionComments method), 339
- update() (uservoice.comments.UserComments method), 354
- update() (uservoice.custom\_fields.CustomFields method), 335
- update() (uservoice.faqs.Faq method), 335
- update() (uservoice.forums.Forum method), 343
- update() (uservoice.forums.Forums method), 344
- update() (uservoice.gadgets.Gadget method), 345
- update() (uservoice.gadgets.Gadgets method), 345
- update() (uservoice.notes.ForumSuggestionNote method), 340
- update() (uservoice.notes.ForumSuggestionNotes method), 340
- update() (uservoice.notes.Notes method), 346
- update() (uservoice.notes.UserNotes method), 355
- update() (uservoice.suggestions.ForumSuggestion method), 341
- update() (uservoice.suggestions.ForumSuggestions method), 342
- update() (uservoice.suggestions.Suggestions method), 348
- update() (uservoice.suggestions.UserSuggestions method), 355
- update() (uservoice.support\_queues.SupportQueue method), 348
- update() (uservoice.support\_queues.SupportQueues method), 349
- update() (uservoice.tickets.Ticket method), 351
- update() (uservoice.tickets.TicketMessages method), 350
- update() (uservoice.tickets.TicketNote method), 350
- update() (uservoice.tickets.TicketNotes method), 351
- update() (uservoice.tickets.Tickets method), 352
- update() (uservoice.topics.Topics method), 353
- update() (uservoice.users.User method), 356
- update() (uservoice.users.Users method), 357
- update() (zendesk.resources.Group method), 360
- update() (zendesk.resources.Ticket method), 362
- update() (zendesk.resources.Tickets method), 363
- update() (zendesk.resources.User method), 363
- update\_preference() (desk.users.User method), 55
- updates() (pipedrive.deals.Deal method), 253
- updates() (pipedrive.organizations.Organization method), 258
- updates() (pipedrive.persons.Person method), 260
- updates() (pipedrive.users.User method), 266
- upload() (basecamp.projects.Project method), 23
- uploads() (basecamp.projects.Project method), 23
- upsert() (uservoice.tickets.Tickets method), 352
- urlmetrics() (mozscape.service.Mozscape method), 241
- usage() (twilio.accounts.Account method), 326
- useful() (uservoice.articles.Article method), 333
- user() (bitbucket.service.BitBucket method), 35
- user() (bitly.service.Bitly method), 41
- user() (desk.service.Desk method), 55
- user() (ducksboard.service.Ducksboard method), 58
- user() (github.service.GitHub method), 136
- user() (instagram.service.Instagram method), 223
- user() (intercom.service.Intercom method), 226
- user() (pipedrive.service.Pipedrive method), 265
- user() (segmentio.service.SegmentIO method), 277
- user() (uservoice.service.UserVoice method), 354
- user() (zendesk.service.Zendesk method), 363
- user\_connections() (pipedrive.service.Pipedrive method), 266
- user\_suggestions() (uservoice.forums.Forum method), 342
- users() (desk.service.Desk method), 55
- users() (desk.users.Group method), 50
- users() (instagram.service.Instagram method), 224

users() (intercom.resource.Company method), 225  
users() (intercom.service.Intercom method), 227  
users() (pipedrive.service.Pipedrive method), 266  
users() (uservoice.service.UserVoice method), 356  
users() (zendesk.service.Zendesk method), 363  
UserVoice (class in uservoice), 332

## V

values() (mixpanel.resources.Properties method), 236  
version() (bitbucket.issues.RepoIssues method), 30  
versions() (bitbucket.issues.RepoIssues method), 30  
view() (googleanalytics.resources.WebProperty method),  
207  
view() (zendesk.service.Zendesk method), 364  
views() (googleanalytics.resources.WebProperty  
method), 208  
views() (zendesk.service.Zendesk method), 364  
viz() (cartodb.service.CartoDB method), 44  
vote() (uservoice.suggestions.ForumSuggestion method),  
341

## W

webproperties() (googleanalytics.resources.Account  
method), 206  
webproperty() (googleanalytics.resources.Account  
method), 207  
widget() (ducksboard.service.Ducksboard method), 58  
widgets() (ducksboard.resources.Dashboard method), 57  
widgets() (ducksboard.service.Ducksboard method), 59  
worksheet() (googlespreadsheets.resource.Spreadsheet  
method), 216  
worksheets() (googlespreadsheets.resource.Spreadsheet  
method), 218

## Y

yearly() (libsaas.services.twilio.usage.Records method),  
329  
yesterday() (libsaas.services.twilio.usage.Records  
method), 329

## Z

Zendesk (class in zendesk), 358