
TravisPy Documentation

Release 0.3.5

Fabio Menegazzo

June 12, 2016

1	Install	3
2	Documentation	5
2.1	Getting started	5
2.2	API	6
2.3	Entities	9
3	Support	19
4	Contribute	21
5	License	23
	Python Module Index	25

TravisPy is a Python API for *Travis CI*. It follows the [official API](#) and is implemented as similar as possible to [Ruby](#) implementation.

Experimental methods will not be supported until they become official.

For full documentation please refer to *Travis CI* [official API](#) documentation.

Install

To install **TravisPy** all it takes is one command line:

```
pip install travispy
```


2.1 Getting started

TravisPy works just as *Travis CI*: it authenticates against *GitHub*. So as a **requirement** you must have a *GitHub* access token with the following scopes:

- read:org
- user:email
- repo_deployment
- repo:status
- write:repo_hook

With your token in hands all is easy:

```
>>> from travispy import TravisPy
>>> t = TravisPy.github_auth(<your_github_token>)
>>> user = t.user()
>>> user
<travispy.entities.user.User object at 0x02C26C48>
```

Now you can access information related to user current logged in:

```
>>> user.login
'travispy'
>>> user['login']
'travispy'
```

To get the list of repositories that you are member of:

```
>>> repos = t.repos(member=user.login)
>>> len(repos) # Ordered by recent activity
5
>>> repos[0]
<travispy.entities.repo.Repo object at 0x02C26C49>
>>> repos[0].slug
'travispy/on_py34'
```

Or simply request for repository you want:

```
>>> repo = t.repo('travispy/on_py34')
<travispy.entities.repo.Repo object at 0x02C26C51>
```

And finally, getting build information:

```
>>> build = t.build(repo.last_build_id)
>>> build
<travispy.entities.build.Build object at 0x02C26C50>
>>> build.restart()
True
>>> build.cancel()
True
>>> build.cancel() # As build was already cancelled it will return False.
False
```

Please refer to the [official API](#) to learn more about which [entities](#) are supported. Soon a specific and detailed documentation related to this library will be available.

2.2 API

This document brings the public API of **TravisPy**.

`travispy.travispy.PUBLIC` = URI for Travis CI free service.

`travispy.travispy.PRIVATE` = URI for Travis CI paid service for GitHub private repositories.

`travispy.travispy.ENTERPRISE` = URI template for Travis CI service running under a personal domain. Usage will be s

class `travispy.travispy.TravisPy` (*token=None, uri='https://api.travis-ci.org'*)

Instances of this class are responsible for communicating with *Travis CI*, sending requests and handling responses properly. You can create as much instances as you want since each one will create a separated session.

Parameters

- **token** (*str* | *None*) – Travis CI token linked to your *GitHub* account.
Even if you have a public repository, some information are related to your user account and not the repository itself so if token is not provided an error will be returned.
Required for private and enterprise repositories to access any information.
- **uri** (*PUBLIC* | *PRIVATE* | *ENTERPRISE* | *str*) – URI where Travis CI service is running.

Note: Do not confuse `token` with the one found on your profile page.

account (*account_id*)

Parameters `account_id` (*int*) – ID of the account to obtain information.

Return type *Account*

Note: This request always needs to be authenticated.

accounts (*all=False*)

Parameters `all` (*bool*) – Whether or not to include accounts the user does not have admin access to.

Return type `list(Account)`

Returns Information of all accounts that the user might have access. This is usually the account corresponding to the user directly and one account per *GitHub* organization.

Note: This request always needs to be authenticated.

branch (*name*, *repo_id_or_slug*, ***kwargs*)

Parameters

- **name** (*str*) – Branch name that should be retrieved.
- **repo_id_or_slug** (*int* | *str*) – Repository where branch is located.

Return type *Branch*

branches (***kwargs*)

Parameters

- **repository_id** (*int*) – Repository id the build belongs to.
- **slug** (*str*) – Repository slug the build belongs to.

Return type list(*Branch*)

Note: You have to supply either `repository_id` or `slug`.

broadcasts ()

Return type list(*Broadcast*)

Note: This request always needs to be authenticated.

build (*build_id*)

Parameters **build_id** (*int*) – ID of the build to obtain information.

Return type *Build*

builds (***kwargs*)

Parameters

- **ids** (*list(int)*) – List of build ids to fetch.
- **repository_id** (*int*) – Repository id the build belongs to.
- **slug** (*str*) – Repository slug the build belongs to.
- **number** (*str*) – Filter by build number, requires `slug` or `repository_id`.
- **after_number** (*str*) – List build after a given build number (use for pagination), requires `slug` or `repository_id`.
- **event_type** (*str*) – Limit build to given event type (push or pull_request).

Return type list(*Build*)

Note: You have to supply either `ids`, `repository_id` or `slug`.

classmethod **github_auth** (*token*, *uri='https://api.travis-ci.org'*)

Parameters

- **token** (*str*) – GitHub access token.
- **uri** – See `__init__()`

Return type *TravisPy*

Returns A *TravisPy* instance authenticated with GitHub account.

Raises **TravisError** – when authentication against GitHub fails.

hooks ()

Return type `list(Hook)`

Returns Returns list of existing hooks that user have access.

Note: This request always needs to be authenticated.

job (*job_id*)

Parameters **job_id** (*int*) – ID of the job to obtain information.

Return type *Job*

jobs (***kwargs*)

Parameters

- **ids** (`list(int)`) – List of jobs IDs.
- **state** (*str*) – Job state to filter by. Possible values are `passed`, `canceled`, `failed` and `errored`.
- **queue** (*str*) – Job queue to filter by.

Return type `list(Job)`

Note: You need to provide exactly one of the above parameters. If you provide `state` or `queue`, a maximum of 250 jobs will be returned.

log (*log_id*)

Parameters **log_id** (*int*) – ID of the log to obtain information.

Return type *Log*

repo (*id_or_slug*)

Parameters **id_or_slug** (*int | str*) – ID of slug of repository to obtain information.

Return type *Repo*

repos (***kwargs*)

Parameters

- **ids** (`list(int)`) – List of repository ids to fetch, cannot be combined with other parameters.
- **member** (*str*) – Filter by user that has access to it (*GitHub* login).
- **owner_name** (*str*) – Filter by owner name (first segment of slug).
- **slug** (*str*) – Filter by slug.

- **search** (*str*) – Filter by search term.
- **active** (*bool*) – If `True`, will only return repositories that are enabled. Default is `False`.

Return type `list(Repo)`

Note: If no parameters are given, a list of repositories with recent activity is returned.

user ()

Return type `User`

Returns Information about user currently logged in.

Note: This request always needs to be authenticated.

2.3 Entities

This document brings information about all entities that are used by **TravisPy** API.

class `travispy.entities._entity.Entity` (*session*)

Base class for all *Travis CI* entities.

Parameters `session` (`Session`) – Internet session in which entity information will be requested.

Variables `id` (*int*) – The entity ID.

classmethod `find_many` (*session*, ***kwargs*)

Method responsible for returning as many as possible matches for current class.

Parameters `session` (`Session`) – Session that must be used to search for results.

Return type `list(Entity)`

Raises `TravisError` – when response has status code different than 200.

classmethod `find_one` (*session*, *entity_id*, ***kwargs*)

Method responsible for returning exactly one instance of current class.

Parameters

- **session** (`Session`) – Session that must be used to search for result.
- **entity_id** (*int*) – The ID of the entity.

Return type `Entity`

Raises `TravisError` – when response has status code different than 200.

classmethod `many` ()

Return type `str`

Returns String representation for multiple entities. Example: for `Account` will be `accounts`.

classmethod `one` ()

Return type `str`

Returns String representation for a single entity. Example: for `Account` will be `account`.

class `travispy.entities._stateful.Stateful` (*session*)

Bases: `travispy.entities._entity.Entity`

Base class for stateful entities such as `Repo`, `Build` and `Job`.

CANCELED

Constant representing state canceled. Should not be changed.

CREATED

Constant representing state created. Should not be changed.

QUEUED

Constant representing state queued. Should not be changed.

STARTED

Constant representing state started. Should not be changed.

PASSED

Constant representing state passed. Should not be changed.

FAILED

Constant representing state failed. Should not be changed.

ERRORED

Constant representing state errored. Should not be changed.

READY

Constant representing state ready. Should not be changed.

GREEN

Constant representing state color green. Should not be changed.

YELLOW

Constant representing state color yellow. Should not be changed.

RED

Constant representing state color red. Should not be changed.

Variables `state` (*str*) – Current state. Possible values are:

- `CANCELED`
- `CREATED`
- `QUEUED`
- `STARTED`
- `PASSED`
- `FAILED`
- `ERRORED`
- `READY`

canceled

Return type `bool`

Returns `True` if build process was canceled.

See also:

`check_state()`

check_state()

Method responsible for checking and validating current *state*.

Raises

- **AttributeError** – when *state* does not exist.
- **ValueError** – when *state* value is not supported.

color

Return type bool

Returns

The color related to current build state. Possible values are:

- *GREEN*: when build has passed or it is ready.
- *YELLOW*: when build process is running.
- *RED*: when build has failed somehow.

created

Return type bool

Returns True if entity build process was created successfully.

See also:

check_state()

errored

Return type bool

Returns True if build process got errors.

See also:

check_state()

failed

Return type bool

Returns True if build process failed. This is usually related to failures on tests.

See also:

check_state()

finished

Return type bool

Returns True if build process is finished.

green

Return type bool

Returns True if build *color* is *GREEN*.

passed

Return type bool

Returns True if build process was finished successfully.

See also:

`check_state()`

pending

Return type bool

Returns True if build was scheduled but was not finished.

See also:

`check_state()`

queued

Return type bool

Returns True if entity was already queued sometime in the build process.

See also:

`check_state()`

ready

Return type bool

Returns True if build process is ready.

See also:

`check_state()`

red

Return type bool

Returns True if build *color* is *RED*.

running

Return type bool

Returns True if build process is running.

See also:

`check_state()`

started

Return type bool

Returns True if entity was already started sometime in the build process.

See also:

`check_state()`

successful

See also:

`passed`

unsuccessful

Return type bool

Returns True if build process was finished unsuccessfully.

yellow

Return type bool

Returns True if build *color* is *YELLOW*.

class `travispy.entities._restartable.Restartable` (*session*)

Bases: `travispy.entities._stateful.Stateful`

Base class for restartable entities such as *Build* and *Job*.

cancel ()

Method responsible for canceling current action of this object.

Return type bool

Returns True if cancel request was send successfully to *Travis CI*.

restart ()

Method responsible for restarting the last action executed by this action.

Return type bool

Returns True if restart request was send successfully to *Travis CI*.

class `travispy.entities.Account` (*session*)

Bases: `travispy.entities._entity.Entity`

A user might have access to multiple accounts. This is usually the account corresponding to the user directly and one account per *GitHub* organization.

Variables

- **name** (*str*) – User or organization id.
- **login** (*str*) – Account name on *GitHub*.
- **type** (*str*) – Account login on *GitHub*.
- **repos_count** (*int*) – Number of repositories.
- **subscribed** (*bool*) – Whether or not the account has a valid subscription. Only available on *Travis Pro*.
- **avatar_url** (*str*) – Link to avatar.

class `travispy.entities.Branch` (*session*)

Bases: `travispy.entities._stateful.Stateful`

Variables

- **repository_id** (*int*) – Repository ID.
- **commit_id** (*str*) – Commit ID.
- **number** (*str*) – Build number.
- **config** (*dict*) – Build config (secure values and ssh key removed). It comes from `.travis.yml` file.
- **started_at** (*str*) – Time the build was started.
- **finished_at** (*str*) – Time the build finished.
- **duration** (*str*) – Build duration. It might not correspond to `finished_at - started_at` if the build was restarted at a later point.

- **job_ids** (*list (int)*) – List of job IDs in the build matrix.
- **pull_request** (*bool*) – Whether or not the build comes from a pull request.
- **commit** (*Commit*) – *Commit* information.

jobs

Return type *list(Job)*

Returns A list of *Job* objects with information related to current *job_ids*.

repository

Return type *Repo*

Returns A *Repo* object with information related to current *repository_id*.

class `travispy.entities.Broadcast` (*session*)

Bases: `travispy.entities._entity.Entity`

Variables **message** (*str*) – Broadcast message.

class `travispy.entities.Build` (*session*)

Bases: `travispy.entities._restartable.Restartable`

Variables

- **repository_id** (*int*) – Repository ID.
- **commit_id** (*str*) – Commit ID.
- **number** (*str*) – Build number.
- **pull_request** (*bool*) – Whether or not the build comes from a pull request.
- **pull_request_title** (*str*) – PR title if *pull_request* is True.
- **pull_request_number** (*str*) – PR number if *pull_request* is True.
- **config** (*dict*) – Build config (secure values and ssh key removed). It comes from `.travis.yml` file.
- **started_at** (*str*) – Time the build was started.
- **finished_at** (*str*) – Time the build finished.
- **duration** (*str*) – Build duration. It might not correspond to *finished_at* - *started_at* if the build was restarted at a later point.
- **job_ids** (*list (int)*) – List of job IDs in the build matrix.
- **jobs** (*list (Job)*) – List of *Job* in the build matrix.
- **commit** (*Commit*) – *Commit* information.

repository

Return type *Repo*

Returns A *Repo* object with information related to current *repository_id*.

class `travispy.entities.Commit` (*session*)

Bases: `travispy.entities._entity.Entity`

There is no API endpoint for resolving commits, however commit data might be included in other API entities, like *Build* or *Job*.

Variables

- **sha** (*str*) – Commit SHA.
- **branch** (*str*) – Branch the commit is on.
- **message** (*str*) – Commit message.
- **committed_at** (*str*) – Commit date.
- **author_name** (*str*) – Author name.
- **author_email** (*str*) – Author email.
- **committer_name** (*str*) – Committer name.
- **committer_email** (*str*) – Committer email.
- **compare_url** (*str*) – Link to diff on GitHub.
- **tag** (*str*) – Tag name.
- **pull_request_number** (*int*) – Pull request number.

class `travispy.entities.Hook` (*session*)

Bases: `travispy.entities._entity.Entity`

Variables

- **name** (*str*) – Hook name.
- **description** (*str*) – Hook description.
- **owner_name** (*str*) – Owner name.
- **active** (*str*) – Whether or not the hook is active.
- **private** (*str*) – Whether or not the hook is private.
- **admin** (*bool*) – Whether or not current user has administrator privileges.

class `travispy.entities.Job` (*session*)

Bases: `travispy.entities._restartable.Restartable`

Variables

- **build_id** (*int*) – Build ID.
- **repository_id** (*int*) – Repository ID.
- **commit_id** (*int*) – Commit ID.
- **log_id** (*int*) – Log ID.
- **number** (*str*) – Job number.
- **config** (*dict*) – Job config (secure values and ssh key removed). It comes from `.travis.yml` file.
- **started_at** (*str*) – Time the job was started.
- **finished_at** (*str*) – Time the job finished.
- **duration** (*str*) – Job duration. It might not correspond to `finished_at` - `started_at` if the job was restarted at a later point.
- **queue** (*str*) – Job queue.
- **allow_failure** (*bool*) – Whether or not the job state influences build state.
- **annotation_ids** (*list(int)*) – List of annotation IDs.

- **commit** (*Commit*) – *Commit* information.

build

Return type *Build*

Returns A *Build* object with information related to current `build_id`.

log

Return type *Log*

Returns A *Log* object with information related to current `log_id`.

repository

Return type *Repo*

Returns A *Repo* object with information related to current `repository_id`.

class `travispy.entities.Log(session)`

Bases: `travispy.entities._entity.Entity`

Variables

- **job_id** (*int*) – Job ID.
- **type** (*str*) –

body

Return type `str`

Returns The raw log text fetched on demand.

get_archived_log()

Return type `str`

Returns The archived log.

job

Return type *Job*

Returns A *Job* object with information related to current `job_id`.

class `travispy.entities.Repo(session)`

Bases: `travispy.entities._stateful.Stateful`

Variables

- **slug** (*str*) – Repository slug.
- **description** (*str*) – Description on *GitHub*.
- **last_build_id** (*int*) – Build ID of the last executed build.
- **last_build_number** (*str*) – Build number of the last executed build.
- **last_build_state** (*str*) – Build state of the last executed build.
- **last_build_duration** (*str*) – Build duration of the last executed build.
- **last_build_started_at** (*str*) – Build started at of the last executed build.
- **last_build_finished_at** (*str*) – Build finished at of the last executed build.
- **github_language** (*str*) – Language on *GitHub*.
- **active** (*bool*) – Whether or not the repository is active on *Travis CI*.

disable()

Disable Travis CI for the repository.

Return type bool

Returns True if API call was successful. False if API call was unsuccessful.

enable()

Enable Travis CI for the repository

Return type bool

Returns True if API call was successful False if API call was unsuccessful

last_build

Return type *Build*

Returns A *Build* object with information related to current `last_build_id`.

state

Repo state is given through `last_build_state`.

See also:

Stateful for state full documentation.

class `travispy.entities.User` (*session*)

Bases: `travispy.entities._entity.Entity`

Variables

- **login** (*str*) – User login on *GitHub*.
- **name** (*str*) – User name on *GitHub*.
- **email** (*str*) – Primary email address on *GitHub*.
- **gravatar_id** (*str*) – Avatar ID.
- **is_syncing** (*bool*) – Whether or not user account is currently being synced.
- **synced_at** (*str*) – Last synced at.
- **correct_scopes** (*bool*) – Whether or not *GitHub* token has the correct scopes.
- **channels** (*str*) – Pusher channels for this user.
- **created_at** (*str*) – When account was created.
- **locale** (*str*) – User main locale.

sync()

Triggers a new sync with *GitHub*. Might return status 409 if user is currently syncing.

Return type bool

Returns True if sync request was send successfully to *Travis CI* and response code is 200
False if a sync is already is progress

Need help? Click [here](#) and open a new issue. You'll get your answer ASAP.

Contribute

TravisPy is under development, so if you want to join the team, you are welcome.

- Feel free to [open issues](#) related to bugs or ideas.
- If you are a developer:
 - Fork **TravisPy** before making any changes.
 - Write tests.
 - Create a *Pull Request* so changes can be merged.

License

TravisPy is licensed under [GPL v3.0](#) license.

t

travispy, 3
travispy.entities, 13
travispy.entities._entity, 9
travispy.entities._restartable, 13
travispy.entities._stateful, 9
travispy.travispy, 6

A

Account (class in `travispy.entities`), 13
account() (`travispy.travispy.TravisPy` method), 6
accounts() (`travispy.travispy.TravisPy` method), 6

B

body (`travispy.entities.Log` attribute), 16
Branch (class in `travispy.entities`), 13
branch() (`travispy.travispy.TravisPy` method), 7
branches() (`travispy.travispy.TravisPy` method), 7
Broadcast (class in `travispy.entities`), 14
broadcasts() (`travispy.travispy.TravisPy` method), 7
Build (class in `travispy.entities`), 14
build (`travispy.entities.Job` attribute), 16
build() (`travispy.travispy.TravisPy` method), 7
builds() (`travispy.travispy.TravisPy` method), 7

C

cancel() (`travispy.entities._restartable.Restartable` method), 13
CANCELED (`travispy.entities._stateful.Stateful` attribute), 10
canceled (`travispy.entities._stateful.Stateful` attribute), 10
check_state() (`travispy.entities._stateful.Stateful` method), 10
color (`travispy.entities._stateful.Stateful` attribute), 11
Commit (class in `travispy.entities`), 14
CREATED (`travispy.entities._stateful.Stateful` attribute), 10
created (`travispy.entities._stateful.Stateful` attribute), 11

D

disable() (`travispy.entities.Repo` method), 16

E

enable() (`travispy.entities.Repo` method), 17
ENTERPRISE (in module `travispy.travispy`), 6
Entity (class in `travispy.entities._entity`), 9
ERRORED (`travispy.entities._stateful.Stateful` attribute), 10

errored (`travispy.entities._stateful.Stateful` attribute), 11

F

FAILED (`travispy.entities._stateful.Stateful` attribute), 10
failed (`travispy.entities._stateful.Stateful` attribute), 11
find_many() (`travispy.entities._entity.Entity` class method), 9
find_one() (`travispy.entities._entity.Entity` class method), 9
finished (`travispy.entities._stateful.Stateful` attribute), 11

G

get_archived_log() (`travispy.entities.Log` method), 16
github_auth() (`travispy.travispy.TravisPy` class method), 7
GREEN (`travispy.entities._stateful.Stateful` attribute), 10
green (`travispy.entities._stateful.Stateful` attribute), 11

H

Hook (class in `travispy.entities`), 15
hooks() (`travispy.travispy.TravisPy` method), 8

J

Job (class in `travispy.entities`), 15
job (`travispy.entities.Log` attribute), 16
job() (`travispy.travispy.TravisPy` method), 8
jobs (`travispy.entities.Branch` attribute), 14
jobs() (`travispy.travispy.TravisPy` method), 8

L

last_build (`travispy.entities.Repo` attribute), 17
Log (class in `travispy.entities`), 16
log (`travispy.entities.Job` attribute), 16
log() (`travispy.travispy.TravisPy` method), 8

M

many() (`travispy.entities._entity.Entity` class method), 9

O

one() (`travispy.entities._entity.Entity` class method), 9

P

PASSED (travispy.entities._stateful.Stateful attribute), 10
passed (travispy.entities._stateful.Stateful attribute), 11
pending (travispy.entities._stateful.Stateful attribute), 12
PRIVATE (in module travispy.travispy), 6
PUBLIC (in module travispy.travispy), 6

Q

QUEUED (travispy.entities._stateful.Stateful attribute),
10
queued (travispy.entities._stateful.Stateful attribute), 12

R

READY (travispy.entities._stateful.Stateful attribute), 10
ready (travispy.entities._stateful.Stateful attribute), 12
RED (travispy.entities._stateful.Stateful attribute), 10
red (travispy.entities._stateful.Stateful attribute), 12
Repo (class in travispy.entities), 16
repo() (travispy.travispy.TravisPy method), 8
repos() (travispy.travispy.TravisPy method), 8
repository (travispy.entities.Branch attribute), 14
repository (travispy.entities.Build attribute), 14
repository (travispy.entities.Job attribute), 16
restart() (travispy.entities._restartable.Restartable
method), 13
Restartable (class in travispy.entities._restartable), 13
running (travispy.entities._stateful.Stateful attribute), 12

S

STARTED (travispy.entities._stateful.Stateful attribute),
10
started (travispy.entities._stateful.Stateful attribute), 12
state (travispy.entities.Repo attribute), 17
Stateful (class in travispy.entities._stateful), 9
successful (travispy.entities._stateful.Stateful attribute),
12
sync() (travispy.entities.User method), 17

T

TravisPy (class in travispy.travispy), 6
travispy (module), 1
travispy.entities (module), 13
travispy.entities._entity (module), 9
travispy.entities._restartable (module), 13
travispy.entities._stateful (module), 9
travispy.travispy (module), 6

U

unsuccessful (travispy.entities._stateful.Stateful at-
tribute), 12
User (class in travispy.entities), 17
user() (travispy.travispy.TravisPy method), 9

Y

YELLOW (travispy.entities._stateful.Stateful attribute),
10
yellow (travispy.entities._stateful.Stateful attribute), 13