
PyTravisCI

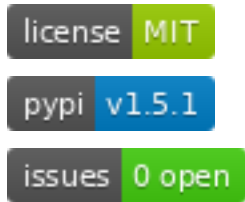
Release 2.0.0a1

Oct 17, 2020

1	History	3
2	Installation	5
2.1	Using pip	5
3	Update	7
3.1	Using pip	7
4	Usage	9
4.1	Authentication	9
4.2	Initialization	9
4.3	Working with resource type objects	10
4.4	Examples	15
5	API	21
6	Resource Types	27
6.1	Not implemented	27
6.2	Base	27
6.3	Active	29
6.4	Beta feature	30
6.5	Beta features	31
6.6	Branch	32
6.7	Branches	33
6.8	Broadcast	34
6.9	Broadcasts	35
6.10	Build	36
6.11	Builds	38
6.12	Cache	39
6.13	Caches	40
6.14	Cron	41
6.15	Crons	42
6.16	Env var	43
6.17	Env vars	45
6.18	Installation	46
6.19	Job	47
6.20	Jobs	49

6.21	Key pair (Generated)	49
6.22	Key pair	50
6.23	Lint	52
6.24	Log	53
6.25	Message	54
6.26	Messages	55
6.27	Organization	55
6.28	Organizations	57
6.29	Repositories	58
6.30	Repository	58
6.31	Request	65
6.32	Requests	67
6.33	Setting	67
6.34	Settings	68
6.35	Stage	69
6.36	Stages	70
6.37	User	71
7	Encryption	75
7.1	Data	75
7.2	File	76
8	Exceptions	79
9	Standardization	81
10	Requester	83
11	Communicator	85
11.1	Base	85
11.2	Active	87
11.3	Beta feature	88
11.4	Beta features	88
11.5	Branch	89
11.6	Branches	90
11.7	Broadcasts	91
11.8	Build	92
11.9	Builds	92
11.10	Cache	93
11.11	Caches	94
11.12	Cron	95
11.13	Crons	96
11.14	Env Var	96
11.15	Env Vars	97
11.16	Job	98
11.17	Jobs	99
11.18	Key Pair (Generated)	100
11.19	Key Pair	101
11.20	Lint	102
11.21	Log	102
11.22	Log	103
11.23	Messages	104
11.24	Organization	105
11.25	Organizations	106
11.26	Repositories	106

11.27 Repository	107
11.28 Request	108
11.29 Requests	109
11.30 Stages	110
11.31 User	111
12 Indices and tables	113
Python Module Index	115
Index	117



Just another Travis CI (API) Python interface. It has been meant to fit my needs but in hope that it will be useful to others.

In other words, PyTravisCI gives us a Python interface to interact with the [Travis API v3](#) or the Travis CI Infrastructure.

It can interact with repositories, jobs, build and almost everything the Travis CI API v3 has to offer.

It can also help you with the encryption of information and files. Give the interface a variable name and its value and it will give you what you are supposed to write into your `.travis.yml` configuration file. Same with the encryption of files and the linting of your configuration files.

Please keep in mind that:

- this project was meant to be used as “an imported module” than something else.
- this project has nothing to do with [the ruby version of the Travis client](#) and will never have similarities with it (except if someone develops a client on top of this).

Have a question ? Fill a new issue!

Have an issue ? Report it!

Have an improvement idea ? Submit a PR!

Every improvement and constructive discussions are welcome!

CHAPTER 1

History

While working on the monitoring and automation tool (private source code) for the [Ultimate-Hosts-Blacklist](#) and [Dead-Hosts](#) infrastructure; I constantly needed to interact with the Travis CI infrastructure - so implicitly its API.

And unfortunately, I could make [The Travis Client](#) work on my (new) machine.

So, as the automation tool itself is written in Python and I could not find any tool which could fit my needs or simplify the way to communicate and/or interpret the response of the Travis CI API V3, I wrote my own.

Therefore, please keep in mind that this project was mainly developed for my needs while developing the first version of the infrastructure behind [Ultimate-Hosts-Blacklist](#) and/or [Dead-Hosts](#).

2.1 Using `pip`

Choose your repository, install and enjoy PyTravisCI!

2.1.1 From PyPi

```
$ pip3 install PyTravisCI
```

2.1.2 From GitHub

```
$ pip3 install git+https://github.com/funilrys/PyTravisCI.git@master#egg=PyTravisCI
```


3.1 Using pip

Choose your repository, install and enjoy PyTravisCI!

3.1.1 From PyPi

```
$ pip3 install --upgrade PyTravisCI
```

3.1.2 From GitHub

```
$ pip3 install --upgrade git+https://github.com/funilrys/PyTravisCI.git@master  
↪ #egg=PyTravisCI
```


4.1 Authentication

As the API v3 documentation states:

To authenticate against the Travis CI API, you need an API access token generated by the Travis CI [command line client](#).

You are required to your API Token though their client or though one of those 2 links:

- <https://travis-ci.org/account/preferences>
- <https://travis-ci.com/account/preferences>

4.2 Initialization

There is 4 way to get started, but 1 gateway.

1. Without any token.
2. With a token from <https://travis-ci.org> (for public GitHub repositories)
3. With a token from <https://travis-ci.com> (private GitHub repositories)
4. With a token from an enterprise domain (e.g <https://example.org/>)

The gateway of all the mechanism is the *TravisCI* object. From there, you can start to navigate through the Travis CI API.

4.2.1 Without any token

```
from PyTravisCI import TravisCI

travis = TravisCI()
```

4.2.2 With a token from travis-ci.org

```
from PyTravisCI import TravisCI

travis = TravisCI(access_token="XYZ")
```

4.2.3 With a token from travis-ci.com

```
from PyTravisCI import defaults, TravisCI

travis = TravisCI(
    access_token="XYZ", access_point=defaults.access_points.PRIVATE
)
```

4.2.4 With a token from an enterprise domain

```
from PyTravisCI import defaults, TravisCI

travis = TravisCI(
    access_token="XYZ", access_point=defaults.access_points.ENTERPRISE.format(
        ↪ "example.org")
)
```

4.3 Working with resource type objects

Although resource types have some nice method to help you, PyTravisCI provides you some other useful “helpers” to make it “easier”.

4.3.1 Access to information

Let’s read our information.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(access_token="XYZ")

# We get our very own account information.
me = travis.get_user()

# We can read the JSON representation to get an idea of what we get.
print(me.json())
```

Now let’s say we want to read our `id` and `login`. We can do it like we will do with a `dict`.

```
my_id = me["id"]
my_login = me["login"]
```

Or by accessing the attribute


```
my_id = me.id
my_login = me.login
```

4.3.2 repr representation

If you print a resource type, you will get the following format.

```
<ResourceTypeClassName {... attributes ...} />
```

4.3.3 dict representation

You can at anytime get the `dict` representation of a resource type object.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3})

# We can now get the dict representation of our object.
# The .dict() method is actually an alias to .to_dict()!
print(my_repositories.dict())
```

4.3.4 JSON representation

You can at anytime get the JSON representation of a representation of a resource type object.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3})

# We can now get the dict representation of our object.
# The .json() method is actually an alias to .to_json()!
print(my_repositories.json())
```

4.3.5 Loop over collection of resources

Most of time, we may work with single entries or resources. But while working with a collection of resource, you may want to loop over them.

Guess what ? You can do it here too!

Warning: If the current resource type is not a collection a `NotImplementedError` will be raised.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3})

for repository in my_repositories:
    print(repository.json())
```

4.3.6 Comparison of resource types

You can easily compare 2 resource types object by checking if `x == y`. Where `x` and `y` are 2 resource type objects.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get a single repository
my_repo = travis.get_repository("funilrys/PyTravisCI")

# In this example because it is a collection we can directly access
# a member of the collection through its position.
# Or (most of time) you will have an attribute which held them.
#
# As presented here, we are accessing the same object in 2 different way.
wanted_job = my_repo.get_builds().builds[0].jobs[0]
wanted_job2 = my_repo.get_builds()[0].jobs[0]

assert wanted_job == wanted_job2
```

4.3.7 Handling incomplete or minimal representation of a resource type

Most of the time, Travis CI give us incomplete attributes which represents a resource type but in its minimal representation. With PyTravisCI you can directly get the complete representation directly.

Let's take our previous example.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get a single repository
my_repo = travis.get_repository("funilrys/PyTravisCI")

# In this example because it is a collection we can directly access
# a member of the collection through its position.
```

(continues on next page)

(continued from previous page)

```
# Or (most of time) you will have an attribute which held them.
#
# As presented here, we are accessing the same object in 2 different way.

# Both variables are minimal representation of the same job.
wanted_job = my_repo.get_builds().builds[0].jobs[0]
wanted_job2 = my_repo.get_builds()[0].jobs[0]

assert wanted_job == wanted_job2

if wanted_job2.is_incomplete():
    print(wanted_job.json()) # incomplete representation
    print("*" * 100)
    print(wanted_job2.get_complete().json()) # complete/standard representation
```

4.3.8 Next page of a resource type

Most of the times, you will have to play with the paging system of the Travis CI API. We made it a bit simpler :-).

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3})

while True:
    for repository in my_repositories:
        print(repository.json())

    if my_repositories.has_next_page():
        my_repositories = my_repositories.next_page()
        continue
    break
```

4.3.9 Last page of a resource type

You can get the last page of a resource type too.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3}).last_page()

for repository in my_repositories:
    print(repository.json())
```

4.3.10 First page of a resource type

Sometime you are in a middle of a loop but for whatever reason, you want to go back to the first page. It's possible too!

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3}).last_page()

while True:
    funilrys_repo_found = False

    for repository in my_repositories:
        print(repository.json())
        if "funilrys" in repository.slug:
            funilrys_repo_found = True

    if not funilrys_repo_found:
        my_repositories = my_repositories.next_page()
        continue
    break

# Now we work from the first page :-)
my_repositories = my_repositories.first_page()
```

4.3.11 Previous page of a resource type

Sometime you want to loop backwards :-).

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the list of repositories we have access to.
# Note: we limit to 3 because we have much more!
my_repositories = travis.get_repositories(params={"limit": 3}).last_page()

while True:
    for repository in my_repositories:
        print(repository.json())

    if my_repositories.has_previous_page():
        my_repositories = my_repositories.previous_page()
        continue
    break
```

4.4 Examples

You are invited to look at the resource types objects in order to get the method and attributes you look for.

In this section we present some example which may be asked a lot in issues. If we are missing some or if you have some questions, let me know per issue!

Warning: This section may be useless if you don't know how to interact with resource type objects.

4.4.1 Information of the current user

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the current user information.
me = travis.get_user()

print(me.json())
```

4.4.2 Repositories of the current user

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the initial list of Repositories
repositories = travis.repositoris()

while True:
    # We loop until there is no more page to navigate.

    for repository in repositories:
        print(repository.json())

    if repositories.has_next_page():
        repositories = repositories.next_page()
        continue
    break
```

4.4.3 Organizations of the current user

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the jobs.
```

(continues on next page)

(continued from previous page)

```
organizations = travis.get_organizations()

while True:
    # We loop until there is no more page to navigate.

    for organization in organizations:
        print(organization.json())

    if organizations.has_next_page():
        organizations = organizations.next_page()
        continue
    break
```

4.4.4 Active builds of the current user

Note: Active builds are build which are builds which have the state pending or started.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the current user information.
me = travis.get_user()

# We can now get the list of active builds.
active_builds = me.get_active()

while True:
    # We loop until there is no more page to navigate.

    for active_build in active_builds:
        print(active_build.json())

    if active_build.has_next_page():
        active_builds = active_builds.next_page()
        continue
    break
```

4.4.5 Builds of the current user

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the builds.
builds = travis.get_builds()

while True:
    # We loop until there is no more page to navigate.
```

(continues on next page)

(continued from previous page)

```

for build in builds:
    print(build.json())

if builds.has_next_page():
    builds = builds.next_page()
    continue
break

```

4.4.6 Jobs of the current user

```

from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the jobs.
jobs = travis.get_jobs()

while True:
    # We loop until there is no more page to navigate.

    for job in jobs:
        print(job.json())

    if jobs.has_next_page():
        jobs = jobs.next_page()
        continue
    break

```

4.4.7 Restart the last build of a repository

```

from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# We get the repository that interests us.
repository = travis.get_repository("funilrys/PyTravisCI")
# We get the build that interest us (the latest one is always the first one).
build = repository.get_builds()[0]

try:
    build.restart()
except PyTravisCI.exceptions.BuildAlreadyStarted:
    # We really want to start so, we cancel it first.
    build.cancel()
    time.sleep(0.5)
    build.restart()

while build.is_active(sync=True):
    print("Build is running...")

```

(continues on next page)

(continued from previous page)

```
time.sleep(5)

print("Build finished!")
```

4.4.8 Lint a configuration file

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

with open(".travis.yml", "r") as file_stream:
    for index, warning in enumerate(travis.lint(file_stream)):
        if index > 0:
            print("*" * 100)

        print(
            f"{index + 1}. WARNING ({warning.warning_type}): \n"
            f"MESSAGE: \n\n{warning.message} \n\n"
        )
```

4.4.9 Create a new (build) request

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# Let's get the repository to work with.
repository = travis.get_repository("funilrys/PyTravisCI")

# Let's create a new request.
print(
    repository.create_request(
        "Hello, this request was created with PyTravisCI", "master"
    ).json()
)
```

4.4.10 Encrypt global environment variables for our configuration files

Travis allow us to directly give is a new environment variable. But we may want our environment variable to be encrypted into our configuration file.

Here is an example which show you how to get the encrypted string to put into your configuration file.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")
```

(continues on next page)

(continued from previous page)

```

# Let's get the repository we want to work with.
repository = travis.get_repository("funilrys/PyTravisCI")

# This is what we are going to encrypt.
# one decrypted by TravisCI it will produces:
#
#   $ export HELLO=[secure]
#   $ export WORLD=[secure]
#
env_vars = {"HELLO": "world", "WORLD": "hello"}

# We now encrypt the shell environment variable:
#   HELLO=world
encrypt_vars = repository.encrypt_env_var(env_vars)

print(
    "Please append the following into the global environment variables "
    "section of your configuration file:"
)

for encrypted_var in encrypt_vars:
    print(f"- secure: \"{encrypted_var['secure']}\"\\n\\n")

```

4.4.11 Encrypt file

You may want to encrypt a file for a repository. This is what we will do in this example.

In this example, we have an off git file which is called `id_rsa`. We want to use it in our build process, so we will encrypt it into `id_rsa.enc` which will be then pushed to the repository.

PyTravisCI can generate the `id_rsa.enc` file for you but you will have to manually write the command to decrypt it. But don't be stressed out, PyTravisCI will give you the command to run.

Here is an example which show you how to get help with the encryption of secret files.

```

from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(access_token="XYZ")

# Let's get the repository we want to work with.
repository = travis.get_repository("funilrys/PyTravisCI")

with open("id_rsa", "rb") as secret_file, open(
    "id_rsa.enc", "wb"
) as encrypted_secret_file:

    information = repository.encrypt_file(secret_file, encrypted_secret_file)

    print(
        "Please append the following into the script section of "
        "your configuration file:\\n\\n"
        f"{information['command']}\\n"
    )

```

4.4.12 Encrypt secrets

You may need to encrypt your/a password that you need to write into your configuration file for the deployment. This is what we will do in this example.

```
from PyTravisCI import TravisCI

# We initiate our "communication" object.
travis = TravisCI(acces_token="XYZ")

# Let's get the repository we want to work with.
repository = travis.get_repository("funilrys/PyTravisCI")

# Let's get our password :-)
password = "HeLlOW0rLd!"

# Let's encrypt our passowrd
encrypted_password = repository.encrypt_secrets([password])[0]

print(f'Here is your encrypted password:\n\n{encrypted_password}')
```

Just another Travis CI (API) Python interface.

A module which provides the gateway to the Travis CI API. All interaction starts with this module's class-es.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.travis_ci.TravisCI(*, access_token: Optional[str] = None, access_point:
                                     Optional[str] = 'https://api.travis-ci.org')
```

The gateway to the interaction with the Travis CI API.

Parameters

- **access_token** (*str*) – The access token to use to authenticate ourselves.
- **access_point** (*str*) – The access point to communicate with.

get_access_point () → *str*

Provides the currently set access point.

get_active_from_github_id (*github_id: Union[str, int], *, params: Optional[dict] = None*) → *PyTravisCI.resource_types.active.Active*

Provides the list of all active builds.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/active>

Parameters

- **github_id** – Documentation missing. The GitHub user or organization ID to get the active builds for.
- **params** – The query parameters to append to the URL.

get_active_from_login (*login: str, *, provider: str = 'github', params: Optional[dict] = None*) → *PyTravisCI.resource_types.active.Active*

Provides the list of all active builds for the given login in the given provider.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/active>

Parameters

- **login** – Documentation missing. The Login of to user to fetch the data for.
- **provider** – Documentation missing.
- **params** – The query parameters to append to the URL.

get_broadcasts (*, *params: Optional[dict] = None*) → *PyTravisCI.resource_types.broadcasts.Broadcasts*

Provides the list of broadcasts of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/broadcasts>

Parameters **params** – The query parameters to append to the URL.

get_build (*build_id: Union[int, str], *, params: Optional[dict] = None*) → *PyTravisCI.resource_types.build.Build*

Provides the build information from its ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/build>

Parameters

- **build_id** – Value uniquely identifying the build.
- **params** – The query parameters to append to the URL.

get_builds (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.builds.Builds
Provides the list of builds of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/build>

Parameters **params** – The query parameters to append to the URL.

get_cron (*cron_id: Union[str, int]*, *, *params: Optional[dict] = None*) → PyTravisCI.resource_types.cron.Cron
Provides a cron from its given ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/cron>

Parameters

- **cron_id** – Value uniquely identifying the cron.
- **params** – The query parameters to append to the URL.

get_job (*job_id: Union[str, int]*, *, *params: Optional[dict] = None*) → PyTravisCI.resource_types.job.Job
Provides a job from its given ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/job>

Parameters

- **job_id** – Value uniquely identifying the job.
- **params** – The query parameters to append to the URL.

get_jobs (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.jobs.Jobs
Provides the list of jobs of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/jobs>

Parameters **params** – The query parameters to append to the URL.

get_organization (*organization_id: Union[str, int]*, *, *params: Optional[dict] = None*) → PyTravisCI.resource_types.organization.Organization
Provides an organization from its given ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/organization>

Parameters

- **organization_id** – Value uniquely identifying the organization.
- **params** – The query parameters to append to the URL.

get_organizations (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.organizations.Organizations
Provides the list of organizations of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/organization>

Parameters **params** – The query parameters to append to the URL.

get_repositories (*, *params: Optional[dict] = None*) → Py-TravisCI.resource_types.repositories.Repositories
Provides the list of repositories of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/repositories>

Parameters **params** – The query parameters to append to the URL.

get_repositories_from_github_id (*github_id: Union[str, int], *, params: Optional[dict] = None*) → Py-TravisCI.resource_types.repositories.Repositories
Provides the list of repositories of the given GitHub ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/repositories>

Parameters

- **github_id** – Documentation missing. The GitHub user or organization ID to get the repositories for.
- **params** – The query parameters to append to the URL.

get_repositories_from_login (*login: str, *, provider: str = 'github', params: Optional[dict] = None*) → PyTravisCI.resource_types.repositories.Repositories
Provides the list of repositories for the given login in the given provider.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/repositories>

Parameters

- **login** – Documentation missing. The Login of to user to fetch the data for.
- **provider** – Documentation missing.
- **params** – The query parameters to append to the URL.

get_repository (*repository_id_or_slug: Union[str, int], *, params: Optional[dict] = None*) → Py-TravisCI.resource_types.repository.Repository
Provides the repository from its given ID or slug.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/repository>

Parameters

- **repository_id_or_slug** – Value uniquely identifying the repository.
- **params** – The query parameters to append to the URL.

get_repository_from_provider (*provider: str, repository_id_or_slug: Union[str, int], *, params: Optional[dict] = None*) → Py-TravisCI.resource_types.repository.Repository

Provides the repository from its given provider, ID or slug.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/repository>

Parameters

- **provider** (*str*) – Documentation missing.
- **repository_id_or_slug** – Value uniquely identifying the repository.
- **params** – The query parameters to append to the URL.

get_user (**, params: Optional[dict] = None*) → PyTravisCI.resource_types.user.User
Provides the information of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/user>

Parameters **params** – The query parameters to append to the URL.

get_user_from_id (*user_id: Union[str, int], *, params: Optional[dict] = None*) → Py-TravisCI.resource_types.user.User
Provides the information of a user from its ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/user>

Parameters

- **user_id** – Value uniquely identifying the user.
- **params** – The query parameters to append to the URL.

lint (*subject: Union[_io.TextIOWrapper, bytes, str]*) → PyTravisCI.resource_types.lint.Lint
Lints the given subject.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/lint>

Parameters **subject** – A string, file stream or bytes.

Raises **TypeError** – When then given **subject** is not correct.

set_access_point (*value: str*) → None
Sets the access point to communicate with.

set_access_token (*value: str*) → None
Sets the access token.

CHAPTER 6

Resource Types

Resource types are objects provided by the Travis CI API documentation. For us, as developers they are our direct way to interact with their API.

Please keep in mind, that I built this tool to stay as close as possible from the official documentation located at <https://developer.travis-ci.org>. Most of the resource types are documented correctly but Travis CI will continue its development cycle, there please take a look at their documentation from time to time.

One of the rare things which was left undocumented are the `params`. Therefore, using this tool without the documentation while using the `params` argument may be a lost of time.

Warning: I tried to make the attributes match the documentation. But with the time some may be missing. You are invited to report them per issue or pull request.

6.1 Not implemented

Here is the list of resource types which are not (yet) implemented into this project. Feel free to submit your PR!

- Beta migration_request
- Beta migration_requests
- Email subscription
- Preference
- Preferences

6.2 Base

Just another Travis CI (API) Python interface.

A module which provides the base of all our resource type objects.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.resource_types.base.ComplexJsonEncoder (*, skipkeys=False,
                                                         ensure_ascii=True,
                                                         check_circular=True,
                                                         allow_nan=True,
                                                         sort_keys=False,      in-
                                                         dent=None,           sepa-
                                                         rators=None,         de-
                                                         fault=None)
```

Provides ours complex JSON encoder.

default (o: Any)

Implements our complex conversion to JSON.

```
class PyTravisCI.resource_types.base.ResourceTypesBase (**kwargs)
```

The base of all ressource types.

dict (*, remove_tags: bool = False) → str

Alias of to_dict.

first_page () → Optional[ResourceTypesBase]

Provides the first page.

Raises **FirstPageNotFound** – If the first page was not found.

get_complete () → Optional[ResourceTypesBase]

Provides the complete representation if the current representation is the minimal one.

has_first_page () → bool

Checks if there is a first page to follow.

has_last_page() → bool
Checks if there is a last page to follow.

has_next_page() → bool
Checks if there is a next page to follow.

has_previous_page() → bool
Checks if there is a previous page to follow.

is_incomplete() → bool
Checks if the current object is incomplete.
An object is consired as incomplete if its representation is `minimal`.

json (*, *remove_tags: bool = False*) → str
Alias of `to_json`.

last_page() → Optional[ResourceTypesBase]
Provides the last page.
Raises *LastPageNotFound* – If the last page was not found.

next_page() → Optional[ResourceTypesBase]
Provides the next page.
Raises *NextPageNotFound* – If the next page was not found.

previous_page() → Optional[ResourceTypesBase]
Provides the previous page.
Raises *PreviousPageNotFound* – If the previous page was not found.

to_dict (*, *remove_tags: bool = False*) → PyTravisCI.resource_types.base.ResourceTypesBase.dict
Converts the current object to dict.
Parameters *remove_tags* – Remove all `@tags` and everything related to PyTravisCI.

to_json (*, *remove_tags: bool = False*) → str
Converts the current object to json.
Parameters *remove_tags* – Remove all `@tags` and everything related to PyTravisCI.

6.3 Active

Just another Travis CI (API) Python interface.

A module which provides the “Active” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal

(continues on next page)

(continued from previous page)

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

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

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

class PyTravisCI.resource_types.active.Active(**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides a list of all “active” builds.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/active>

Variables *builds* (List[*Build*]) – The active builds.

6.4 Beta feature

Just another Travis CI (API) Python interface.

A module which provides the “Beta Feature” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

(continues on next page)

(continued from previous page)

```
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

```
class PyTravisCI.resource_types.beta_feature.BetaFeature (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the description of a beta feature.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/beta_feature

Variables

- **id** (*int*) – Value uniquely identifying the beta feature.
- **name** (*str*) – The name of the feature.
- **description** (*str*) – Longer description of the feature.
- **enabled** (*bool*) – Indicates if the user has this feature turned on.
- **feedback_url** (*str*) – Url for users to leave Travis CI feedback on this feature.

```
delete () → Union[bool, PyTravisCI.resource_types.beta_feature.BetaFeature]
    Deletes the current beta feature.
```

Parameters **params** – The query parameters to append to the URL.

```
disable () → bool
    Disables the current beta feature.
```

```
enable () → bool
    Enables the current beta feature.
```

6.5 Beta features

Just another Travis CI (API) Python interface.

A module which provides the “Beta Features” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
```

(continues on next page)

(continued from previous page)

to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

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

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

class PyTravisCI.resource_types.beta_features.**BetaFeatures** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides a list of all beta features

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/beta_features

Variables *beta_features* (List[*BetaFeature*]) – The active builds.

6.6 Branch

Just another Travis CI (API) Python interface.

A module which provides the “Branch” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

(continues on next page)

(continued from previous page)

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

class PyTravisCI.resource_types.branch.**Branch** (**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a branch.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/branch>

Variables

- **name** (*str*) – Name of the git branch.
- **repository** (*Repository*) – GitHub user or organization the branch belongs to.
- **default_branch** (*bool*) – Whether or not this is the repository’s default branch.
- **exists_on_github** (*bool*) – Whether or not the branch still exists on GitHub.
- **last_build** (*Build*) – Last build on the branch.
- **recent_builds** (*List[Build]*) – Last 10 builds on the branch (when `include=branch.recent_builds` is used).

create_cron (*interval: str, *, dont_run_if_recent_build_exists: bool = False, params: Optional[dict] = None*) → *PyTravisCI.resource_types.cron.Cron*
 Creates a new cron associated to the current branch.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/cron>

Parameters

- **interval** – Interval at which the cron will run (can be daily, weekly or monthly).
- **dont_run_if_recent_build_exists** – Whether a cron build should run if there has been a build on this branch in the last 24 hours.
- **params** – The query parameters to append to the URL.

get_cron (**, params: Optional[dict] = None*) → *Optional[PyTravisCI.resource_types.cron.Cron]*
 Provides the cron of the current branch.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/cron>

Parameters **params** – The query parameters to append to the URL.

6.7 Branches

Just another Travis CI (API) Python interface.

A module which provides the “Branches” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.branches.**Branches** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provide the list of branches of a chosen repository.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/branches>

Variables *branches* (List[*Branch*]) – List of branches.

6.8 Broadcast

Just another Travis CI (API) Python interface.

A module which provides the “Broadcast” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

(continues on next page)

(continued from previous page)

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

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

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

class PyTravisCI.resource_types.broadcast.**Broadcast** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a broadcast.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/broadcast>

Variables

- **id** (*int*) – Value uniquely identifying the broadcast.
- **message** (*str*) – Message to display to the user.
- **created_at** (*:py:class`~datetime.datetime`*) – When the broadcast was created.
- **category** (*str*) – Broadcast category (used for icon and color).
- **active** (*bool*) – Whether or not the broadcast should still be displayed.
- **recipient** – Either a user, organization or repository, or null for global.

6.9 Broadcasts

Just another Travis CI (API) Python interface.

A module which provides the “Broadcasts” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

(continues on next page)

(continued from previous page)

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

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

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

class PyTravisCI.resource_types.broadcasts.**Broadcasts** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides a list of broadcasts.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/broadcasts>

Variables ***broadcasts*** (List[*Broadcast*]) – List of broadcasts.

6.10 Build

Just another Travis CI (API) Python interface.

A module which provides the “Build” resource type.

Author: Nissar Chababy, @funilrys, contactTATAFunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

(continues on next page)

(continued from previous page)

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

class PyTravisCI.resource_types.build.**Build** (**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a build

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/build>

Variables

- **id** (*int*) – Value uniquely identifying the build.
- **number** (*str*) – Incremental number for a repository's builds.
- **state** (*str*) – Current state of the build.
- **duration** (*int*) – Wall clock time in seconds.
- **event_type** (*str*) – Event that triggered the build.
- **previous_state** (*str*) – State of the previous build (useful to see if state changed).
- **pull_request_title** (*str*) – Title of the build's pull request.
- **pull_request_number** (*int*) – Number of the build's pull request.
- **started_at** (*datetime*) – When the build started.
- **finished_at** (*datetime*) – When the build finished.
- **private** (*bool*) – Whether or not the build is private.
- **repository** (*Repository*) – GitHub user or organization the build belongs to.
- **branch** (*Branch*) – The branch the build is associated with.
- **tag** (*str*) – The build's tag.
- **commit** (*Commit*) – The commit the build is associated with.
- **jobs** (*List[Job]*) – List of jobs that are part of the build's matrix.
- **stages** (*List[Stage]*) – The stages of the build.
- **created_by** – The User or Organization that created the build.
- **updated_at** (*datetime*) – The last time the build was updated.

cancel () → *PyTravisCI.resource_types.build.Build*
 Cancels the current build.

Raises *BuildAlreadyStopped* – When the current build was already canceled.

get_jobs (*, *params: Optional[dict] = None*) → *PyTravisCI.resource_types.jobs.Jobs*
 Provides the list of jobs of the current build

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/jobs>

Parameters **params** – The query parameters to append to the URL.

get_stages (*, *params: Optional[dict] = None*) → `PyTravisCI.resource_types.stages.Stages`
Provides the list of stages of the current build

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/stages>

Parameters **params** – The query parameters to append to the URL.

is_active (*, *sync: bool = False*) → `bool`
Checks if the build is active.

Parameters **sync** – Authorizes the synchronization before checking.

is_canceled (*, *sync: bool = False*) → `bool`
Checks if the build canceled.

Parameters **sync** – Authorizes the synchronization before checking.

is_created (*, *sync: bool = False*) → `bool`
Checks if the build is created.

Parameters **sync** – Authorizes the synchronization before checking.

is_errored (*, *sync: bool = False*) → `bool`
Checks if the build errored.

Parameters **sync** – Authorizes the synchronization before checking.

is_failed (*, *sync: bool = False*) → `bool`
Checks if the build failed.

Parameters **sync** – Authorizes the synchronization before checking.

is_passed (*, *sync: bool = False*) → `bool`
Checks if the build is passed.

Parameters **sync** – Authorizes the synchronization before checking.

is_started (*, *sync: bool = False*) → `bool`
Checks if the build is started.

Parameters **sync** – Authorizes the synchronization before checking.

restart () → `PyTravisCI.resource_types.build.Build`
Restarts the current build.

Raises *BuildAlreadyStarted* – When the current build was already started.

sync () → `PyTravisCI.resource_types.build.Build`
Fetches the latest information of the current job.

6.11 Builds

Just another Travis CI (API) Python interface.

A module which provides the “Builds” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.builds.**Builds** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the list of builds.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/builds>

Variables *builds* (List[*Build*]) – List of builds.

6.12 Cache

Just another Travis CI (API) Python interface.

A module which provides the “Cache” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

(continues on next page)

(continued from previous page)

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

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

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

```
class PyTravisCI.resource_types.cache.Cache (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase

    Provides the description of a cache.
```

Warning: This resource type can't be compared with any official resource type. In fact, at the time we write this, there no such thing officially in the Travis CI API documentation.

That's the reason we don't document much more.

```
delete () → bool
    Deletes the current cache.
```

6.13 Caches

Just another Travis CI (API) Python interface.

A module which provides the "Caches" resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

furnished to do so, subject to the following conditions:

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

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

class PyTravisCI.resource_types.caches.**Caches** (**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provide the list of caches of a chosen repository.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/caches>

Variables

- **caches** (List[*Cache*]) – List of caches.
- **repo** (Repository) – An internal representation of the repository linked to the current caches.
- **name** (*str*) – The name of the repository linked to the current caches.

delete () → Union[bool, PyTravisCI.resource_types.caches.Caches]
 Deletes the current cache

6.14 Cron

Just another Travis CI (API) Python interface.

A module which provides the “Cron” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

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

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

```
class PyTravisCI.resource_types.cron.Cron (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the description of a cron.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/cron>

Variables

- **id** (*int*) – Value uniquely identifying the build.
- **repository** (*Repository*) – Github repository to which this cron belongs.
- **branch** (*Branch*) – Git branch of repository to which this cron belongs.
- **interval** (*str*) – Interval at which the cron will run (can be “daily”, “weekly” or “monthly”).
- **dont_run_if_recent_build_exists** (*bool*) – Whether a cron build should run if there has been a build on this branch in the last 24 hours.
- **last_run** (*datetime*) – When the cron ran last.
- **next_run** (*datetime*) – When the cron is scheduled to run next.
- **created_at** (*datetime*) – When the cron was created.
- **active** (*bool*) – The cron’s active.

```
delete () → Union[bool, PyTravisCI.resource_types.cron.Cron]
    Deletes the current cron.
```

6.15 Crons

Just another Travis CI (API) Python interface.

A module which provides the “Crons” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.crons.**Crons** (**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the list of crons.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/crons>

Variables *crons* (List[*Cron*]) – List of crons.

6.16 Env var

Just another Travis CI (API) Python interface.

A module which provides the “Env Var” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

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

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

class PyTravisCI.resource_types.env_var.EnvVar (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of an environment variable.

Official Travis CI API documentation

- https://developer.travis-ci.org/resource/env_var

Variables **id** (*str*) – The environment variable id.

:ivar str name; The environment variable name, e.g. FOO.

Variables

- **value** (*str*) – The environment variable's value, e.g. bar.
- **public** (*bool*) – Whether this environment variable should be publicly visible or not.
- **branch** – The env_var's branch.

delete () → Union[bool, PyTravisCI.resource_types.env_var.EnvVar]
Deletes the current environment variable.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

make_private () → PyTravisCI.resource_types.env_var.EnvVar
Makes this environment variable public.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

make_public () → PyTravisCI.resource_types.env_var.EnvVar
Makes this environment variable public.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

set_branch (*branch_name: str*) → PyTravisCI.resource_types.env_var.EnvVar
Sets the name of the branch which is covered by the current environment variable.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

Parameters **branch_name** – The name of the branch to apply.

set_name (*name: str*) → PyTravisCI.resource_types.env_var.EnvVar
Sets the new name of the current environment variable.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

Parameters **name** – The new name

Raises **TypeError** – Whether name is not `str`.

set_value (*value: str*) → PyTravisCI.resource_types.env_var.EnvVar
Sets the new value of the current environment variable.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

Parameters **value** – The new value.

Raises **TypeError** – Whether value is not `str`.

6.17 Env vars

Just another Travis CI (API) Python interface.

A module which provides the “Env Vars” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.env_vars.**EnvVars** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*
Provides the list of environment variables.

Official Travis CI API documentation

- https://developer.travis-ci.org/resource/env_vars

Variables *env_vars* (List[*PyTravisCI.resource_types.env_var.EnvVar*]) – List of *env_vars*.

6.18 Installation

Just another Travis CI (API) Python interface.

A module which provides the “Installation” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.installation.**Installation** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*
Provides the description of a GitHub App installation

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/installation>

Variables

- **id** (*int*) – Value uniquely identifying the user.

- `github_id` (*int*) – The installation’s id on GitHub.
- `owner` – GitHub user or organization the installation belongs to.

6.19 Job

Just another Travis CI (API) Python interface.

A module which provides the “Job” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.resource_types.job.Job(**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the description of a single build.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/job>

Variables

- `id` (*int*) – Value uniquely identifying the job.
- `allow_failure` (*bool*) – The job’s allow_failure.
- `number` (*str*) – Incremental number for a repository’s builds.
- `state` (*str*) – Current state of the job.
- `started_at` (*datetime*) – When the job started.
- `finished_at` (*datetime*) – When the job finished.

- **build** (*Build*) – The build the job is associated with.
- **queue** (*str*) – Worker queue this job is/was scheduled on.
- **repository** (*Repository*) – GitHub user or organization the job belongs to.
- **commit** (*Commit*) – The commit the job is associated with.
- **owner** – GitHub user or organization the job belongs to.
- **stage** (*List[:class`~PyTravisCI.resource\_types.stage.Stage`]*) – The stages of the job.
- **created_at** (*datetime*) – When the job was created.
- **updated_at** (*datetime*) – When the job was updated.

cancel () → *PyTravisCI.resource_types.job.Job*
Cancels the current job.

debug () → *PyTravisCI.resource_types.job.Job*
Restart the current job in debug mode.

Warning: This method may not work if you are not allowed to restart in debug mode.

get_log (*, *params: Optional[dict] = None*) → *PyTravisCI.resource_types.log.Log*
Provides the logs of the current job.

Parameters **params** – The query parameters to append to the URL.

is_active (*, *sync: bool = False*) → *bool*
Checks if the job is active.

Parameters **sync** – Authorizes the synchronization before checking.

is_canceled (*, *sync: bool = False*) → *bool*
Checks if the job canceled.

Parameters **sync** – Authorizes the synchronization before checking.

is_created (*, *sync: bool = False*) → *bool*
Checks if the job is created.

Parameters **sync** – Authorizes the synchronization before checking.

is_errored (*, *sync: bool = False*) → *bool*
Checks if the job errored.

Parameters **sync** – Authorizes the synchronization before checking.

is_failed (*, *sync: bool = False*) → *bool*
Checks if the job failed.

Parameters **sync** – Authorizes the synchronization before checking.

is_passed (*, *sync: bool = False*) → *bool*
Checks if the job is passed.

Parameters **sync** – Authorizes the synchronization before checking.

is_started (*, *sync: bool = False*) → *bool*
Checks if the job is started.

Parameters **sync** – Authorizes the synchronization before checking.

restart () → PyTravisCI.resource_types.job.Job
Restarts the current job.

Raises *JobAlreadyStarted* – When the current job was already started.

sync () → PyTravisCI.resource_types.job.Job
Fetches the latest information of the current job.

6.20 Jobs

Just another Travis CI (API) Python interface.

A module which provides the “Jobs” resource type.

Author: Nissar Chababy, @funilrys, contactTATAFunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.jobs.**Jobs** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides a list of builds.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/jobs>

Variables **jobs** (List[Job]) – List of jobs.

6.21 Key pair (Generated)

Just another Travis CI (API) Python interface.

A module which provides the “Key Pair (Generated)” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.key_pair_generated.**KeyPairGenerated**(**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a RSA key pair (generated).

Official Travis CI API documentation

- https://developer.travis-ci.org/resource/key_pair_generated

Variables

- **description** (*str*) – A text description.
- **public_key** (*str*) – The public key.
- **fingerprint** (*str*) – The fingerprint.

regenerate () → PyTravisCI.resource_types.key_pair_generated.KeyPairGenerated
Generates a new RSA key pair and return its representation.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair_generated

6.22 Key pair

Just another Travis CI (API) Python interface.

A module which provides the “Key Pair” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.key_pair.**KeyPair** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a key pair.

Official Travis CI API documentation

- https://developer.travis-ci.org/resource/key_pair

Variables

- **description** (*str*) – A text description.
- **public_key** (*str*) – The public key.
- **fingerprint** (*str*) – The fingerprint.

delete () → Union[bool, PyTravisCI.resource_types.key_pair.KeyPair]
Deletes the current key pair.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair

set_description (*description: str*) → PyTravisCI.resource_types.key_pair.KeyPair
Sets the new description of the current key pair.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair

Parameters value – The new value (private key).

Raises **TypeError** – Whether value is not :py:class`str`.

set_value (value: str) → PyTravisCI.resource_types.key_pair.KeyPair
Sets the new value (private key) of the current key pair.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair

Parameters value – The new value (private key).

Raises **TypeError** – Whether value is not :py:class`str` nor :py:class`bytes`.

6.23 Lint

Just another Travis CI (API) Python interface.

A module which provides the “Lint” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.resource_types.lint.Lint (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the description of a linting.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/lint>

Variables warnings (list) – An array of hashes with keys and warnings.

```
class PyTravisCI.resource_types.lint.LintWarning(**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase

    Provides a placeholder for the lintings warnings.
```

6.24 Log

Just another Travis CI (API) Python interface.

A module which provides the “Log” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.resource_types.log.Log(**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase

    Provides the description of a job log.
```

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/log>

Variables

- **id** (*int*) – The log’s id.
- **content** (*str*) – The content of the log.
- **log_parts** (*list*) – The log parts that form the log.

```
delete () → Union[bool, PyTravisCI.resource_types.log.Log]
    Deletes the current log
```

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair

6.25 Message

Just another Travis CI (API) Python interface.

A module which provides the “Message” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.resource_types.message.Message(**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the description of an individual message.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/message>

Variables

- **id** (*int*) – The log’s id.
- **level** (*str*) – The message’s level.
- **key** (*str*) – The message’s key.
- **code** (*str*) – The message’s code.
- **args** (*dict*) – The message’s args.
- **src** – The message’s src.

- **line** – The message’s line.

6.26 Messages

Just another Travis CI (API) Python interface.

A module which provides the “Messages” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.messages.**Messages** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides a list of messages.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/messages>

Variables **messages** (*List[PyTravisCI.resource_types.Message]*) – List of messages.

6.27 Organization

Just another Travis CI (API) Python interface.

A module which provides the “Organization” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.organization.**Organization**(**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provide the description of an organization.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/organization>

Variables

- **id** (*int*) – Value uniquely identifying the user.
- **login** (*str*) – Login set on Github.
- **name** (*str*) – Name set on GitHub.
- **github_id** (*int*) – Id set on GitHub.
- **vcs_id** – The user's vcs_id.
- **vcs_type** – The user's vcs_type.
- **avatar_url** (*str*) – Avatar URL set on GitHub.
- **education** (*bool*) – Whether or not the user has an education account.
- **allow_migration** – The user's allow_migration.
- **repositories** (List[PyTravisCI.resource_types.repository.Repository]) – Repositories belonging to this organization.
- **installation** (PyTravisCI.resource_types.installation.Installation) – Installation belonging to the organization.

get_active (*, *params: Optional[dict] = None*) → `PyTravisCI.resource_types.active.Active`
 Provides the list of active builds of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/active>

Parameters *params* – The query parameters to append to the URL.

6.28 Organizations

Just another Travis CI (API) Python interface.

A module which provides the “Organizations” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class `PyTravisCI.resource_types.organizations.Organizations` (**kwargs)

Bases: `PyTravisCI.resource_types.base.ResourceTypesBase`

Provides a list of organization.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/organizations>

Variables *organizations* (`List[PyTravisCI.resource_types.organization.Organization]`) – A list of organization.

6.29 Repositories

Just another Travis CI (API) Python interface.

A module which provides the “Repositories” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.resource_types.repositories.Repositories(**kwargs)
```

```
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

```
    Provides a list of repository.
```

```
    Official Travis CI API documentation
```

- <https://developer.travis-ci.org/resource/repositories>

```
    Variables repositories (List[PyTravisCI.resource_types.repository.
Repository]) – A list of repository.
```

6.30 Repository

Just another Travis CI (API) Python interface.

A module which provides the “Repository” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.repository.**Repository** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a repository.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/repository>

Variables

- **id** (*int*) – Value uniquely identifying the repository.
- **name** (*str*) – The repository's name on GitHub.
- **slug** (*str*) – Same as {repository.owner.name}/{repository.name}.
- **description** (*str*) – The repository's description from GitHub.
- **github_id** (*int*) – The repository's id on GitHub.
- **vcs_id** – The repository's vcs_id.
- **vcs_type** – The repository's vcs_type.
- **github_language** (*str*) – The main programming language used according to GitHub.
- **active** (*bool*) – Whether or not this repository is currently enabled on Travis CI.
- **private** (*bool*) – Whether or not this repository is private.
- **owner** – GitHub user or organization the repository belongs to.
- **owner_name** – The repository's owner_name.
- **vcs_name** – The repository's vcs_name.
- **default_branch** (*Branch*) – The default branch on GitHub.
- **starred** (*bool*) – Whether or not this repository is starred.

- **managed_by_installation** (*bool*) – Whether or not this repository is managed by a GitHub App installation.
- **active_on_org** (*bool*) – Whether or not this repository runs builds on travis-ci.org (may also be null).
- **migration_status** – The repository’s migration_status.
- **history_migration_status** – The repository’s history_migration_status.
- **shared** – The repository’s shared.
- **config_validation** – The repository’s config_validation.
- **allow_migration** – The repository’s allow_migration.

activate (*, *params: Optional[dict] = None*) → `PyTravisCI.resource_types.repository.Repository`
Activates the current repository, allowing its test to be run on Travis Ci.

Parameters **params** – The query parameters to append to the URL.

create_env_var (*name: str, value: str, *, is_public: bool = False, branch: Optional[str] = None, params: Optional[dict] = None*) → `PyTravisCI.resource_types.env_var.EnvVar`
Creates a new environment variable into the current repository.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_vars

Parameters

- **name** – The environment variable name, e.g. FOO.
- **value** – The environment variable’s value, e.g. bar.
- **is_public** – Whether this environment variable should be publicly visible or not.
- **branch** – The env_var’s branch.

Raises **TypeError** – When the types of `name` and `value` are not `:py:class‘str‘`.

create_key_pair (*description: str, value: Union[str, bytes], *, params: Optional[dict] = None*) → `PyTravisCI.resource_types.key_pair.KeyPair`
Creates a new RSA key pair.

Parameters

- **description** – A text description.
- **value** – The private key.

Raises **TypeError** – When the types of `name` and `value` are not `:py:class‘str‘` nor `:py:class‘bytes‘`.

create_request (*message: str, branch: str, *, config: Optional[dict] = None, params: Optional[dict] = None*) → `PyTravisCI.resource_types.request.Request`
Creates a Request

Parameters

- **message** – Travis-ci status message attached to the request.
- **branch** – Branch requested to be built.
- **config** – Build configuration (as parsed from `.travis.yml`).

Raises `TypeError` – When the types of `name` and `value` are not `:py:class`str`` nor `:py:class`bytes``.

`deactivate` (*, *params: Optional[dict] = None*) → `PyTravisCI.resource_types.repository.Repository`
 Activates the current repository, preventing any tests from runningIs on Travis CI.

Parameters `params` – The query parameters to append to the URL.

`encrypt_env_var` (*env_vars: dict, padding: Optional[str] = 'PKCS1v15'*) → `dict`
 Process the encryption of the given environment variables.

Parameters

- **`env_vars`** – The key-value representing the environment variables to encrypt.

Warning: Spaces in keys will be automatically converted to underscore!

- **`padding`** – The padding to use.

Supported by PyTravisCI:

- `PKCS1v15`
- `OAEP`

Supported by Travis CI:

- `PKCS1v15`

Warning: DO NOT CHANGE THIS UNLESS INVITED TO.

As of today, Travis CI use the `PKCS1v15` padding. But it may be possible that one day they will change it to `OAEP`.

Please report to the following references/discussion:

- <https://github.com/travis-ci/travis-ci/issues/5394>
- <https://github.com/travis-ci/travis-ci/issues/5394#issue-124658348>

Returns

A list representing each encrypted values.

As example, if the following is given:

```
{
  "HELLO": "WORLD",
  "WORLD": "HELLO"
}
```

The response will be:

```
[
  { "secure": "encrypted version of HELLO=WORLD" },
  { "secure": "encrypted version of WORLD=HELLO" }
]
```

`encrypt_file` (*input_file: Union[io.IOBase, str], output_file: Union[io.IOBase, str], *, branch: Optional[str] = None*) → `dict`
 Encrypts the content of the given `input_file` into `output_file`.

Side Effects:

- Generates a new IV key.
- Generates a new encryption key.
- Save the IV key into a (new) repository environment variable.
- Save the encryption key into a (new) repository environment variable.

Parameters

- **input_file** – The (plain) file to read.
If a `str` is given, this method will open and close the file for you.
If a `io.TextIOWrapper` is given, this method expects it to be in `rb` mode.
- **output_file** – The file to write.
If a `str` is given, this method will open and close the file for you.
If a `io.TextIOWrapper` is given, this method expects it to be in `wb` mode.
- **branch** – The branch to save the IV and key for.

Returns

A `dict` which represents the data which are supposed to help the end-user decrypt the encrypted data.

Given an input file `hello` and an output file `hello.enc`, this method will provides the following:

```
{
  "command": "openssl aes-256-cbc -K "
    "$ENCRYPTED_BB6A5397D5B2_KEY -iv $ENCRYPTED_BB6A5397D5B2_IV "
    "-in hello.enc -out hello -d",
  "iv": {
    "ENCRYPTED_BB6A5397D5B2_IV": "hexadecimal representation of_
↪the IV."
  },
  "key": {
    "ENCRYPTED_BB6A5397D5B2_KEY": "hexadecimal representation_
↪of the key."
  },
}
```

encrypt_secrets (*secrets: List[Union[str, bytes]], padding: Optional[str] = 'PKCS1v15'*) → str
Encrypts the given secret.

Parameters

- **secret** – A list of secrets to encrypt.
- **padding** – The padding to use.

Supported by PyTravisCI:

- PKCS1v15
- OAEP

Supported by Travis CI:

- PKCS1v15

Warning: DO NOT CHANGE THIS UNLESS INVITED TO.

As of today, Travis CI use the PKCS1v15 padding. But it may be possible that one day they will change it to OAEP.

Please report to the following references/discussion:

- <https://github.com/travis-ci/travis-ci/issues/5394>
- <https://github.com/travis-ci/travis-ci/issues/5394#issue-124658348>

Returns A list of encrypted secrets.

generate_key_pair (*, *params*: *Optional[dict]* = *None*) → Py-TravisCI.resource_types.key_pair_generated.KeyPairGenerated
Generates a new RSA key pair.

get_branch (*branch_name*: *str*, *, *params*: *Optional[dict]* = *None*) → Py-TravisCI.resource_types.branch.Branch
Provides the information of a given branch.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/branch>

Parameters

- **branch_name** – Name of the git branch.
- **params** – The query parameters to append to the URL.

get_branches (*, *params*: *Optional[dict]* = *None*) → PyTravisCI.resource_types.branches.Branches
Provides the list of branches of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/branches>

Parameters **params** – The query parameters to append to the URL.

get_builds (*, *params*: *Optional[dict]* = *None*) → PyTravisCI.resource_types.builds.Builds
Provides the list of builds of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/builds>

Parameters **params** – The query parameters to append to the URL.

get_caches (*, *params*: *Optional[dict]* = *None*) → PyTravisCI.resource_types.caches.Caches
Provides the list of caches of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/caches>

Parameters **params** – The query parameters to append to the URL.

get_crons (*, *params*: *Optional[dict]* = *None*) → PyTravisCI.resource_types.crons.Crons
Provides the list of crons of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/crons>

Parameters **params** – The query parameters to append to the URL.

get_env_var (*env_var_id*: *str*, *, *params*: *Optional[dict]* = *None*) → *Py-TravisCI.resource_types.env_var.EnvVar*
Provides an environment variable from its ID.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_var

Parameters

- **env_var_id** – The ID of the environment variable to get.
- **params** – The query parameters to append to the URL.

get_env_vars (*, *params*: *Optional[dict]* = *None*) → *PyTravisCI.resource_types.env_vars.EnvVars*
Provides the list of environment variables of the current repository.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/env_vars

Parameters **params** – The query parameters to append to the URL.

get_key_pair (*, *params*: *Optional[dict]* = *None*) → *PyTravisCI.resource_types.key_pair.KeyPair*
Provides the RSA key pair of the current repository.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair

Parameters **params** – The query parameters to append to the URL.

get_key_pair_generated (*, *params*: *Optional[dict]* = *None*) → *Py-TravisCI.resource_types.key_pair_generated.KeyPairGenerated*
Provides the generated RSA key pair of the current repository.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/key_pair_generated

Parameters **params** – The query parameters to append to the URL.

get_request (*request_id*: *Union[str, int]*, *, *params*: *Optional[dict]* = *None*) → *Py-TravisCI.resource_types.request.Request*
Provides a single request from its given ID.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/requests>

Parameters

- **request_id** – The ID of the request to get.
- **params** – The query parameters to append to the URL.

get_requests (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.requests.Requests
Provides the list of requests of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/requests>

Parameters *params* – The query parameters to append to the URL.

get_setting (*setting_name: str*, *, *params: Optional[dict] = None*) → PyTravisCI.resource_types.setting.Setting
Provides a single setting from its given name.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/setting>

Parameters *params* – The query parameters to append to the URL.

get_settings (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.settings.Settings
Provides the list of settings of the current repository.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/settings>

Parameters *params* – The query parameters to append to the URL.

star (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.repository.Repository
Stars the current repository.

Parameters *params* – The query parameters to append to the URL.

unstar (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.repository.Repository
Unstars the current repository.

Parameters *params* – The query parameters to append to the URL.

6.31 Request

Just another Travis CI (API) Python interface.

A module which provides the “Request” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights

(continues on next page)

(continued from previous page)

to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

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

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

class PyTravisCI.resource_types.request.**Request** (**kwargs)
Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a request.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/request>

Variables

- **id** (*int*) – Value uniquely identifying the build.
- **state** (*str*) – Current state of the build.
- **result** (*str*) – The result of the request (eg. rejected or approved).
- **message** (*str*) – Travis-ci status message attached to the request.
- **previous_state** (*str*) – State of the previous build (useful to see if state changed).
- **pull_request_mergeable** – The request's pull_request_mergeable.
- **repository** (*Repository*) – GitHub user or organization the build belongs to.
- **branch_name** (*str*) – Name of the branch requested to be built.
- **commit** (*Commit*) – The commit the request is associated with.
- **builds** (*List[Build]*) – The request's builds.
- **owner** – GitHub user or organization the request belongs to.
- **created_at** (*datetime*) – When the build started.
- **event_type** (*str*) – Origin of request (push, pull request, api).
- **base_commit** (*str*) – The base commit the request is associated with.
- **head_commit** (*str*) – The head commit the request is associated with.
- **messages** – The request's messages.
- **config** – Build configuration (as parsed from .travis.yml).
- **raw_configs** – The request's raw_configs.

6.32 Requests

Just another Travis CI (API) Python interface.

A module which provides the “Requests” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.requests.**Requests** (**kwargs)

Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the list of requests.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/builds>

Variables **requests** – List of requests.

6.33 Setting

Just another Travis CI (API) Python interface.

A module which provides the “Setting” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.resource_types.setting.**Setting**(**kwargs)

Provides an individual repository setting.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/setting>

Variables

- **name** (*str*) – The settings’s name.
- **value** (*str*) – The settings’s value.

set_value (*value: Union[bool, int]*) → PyTravisCI.resource_types.setting.Setting

Set the new value of the current setting.

Parameters value – The new value.

6.34 Settings

Just another Travis CI (API) Python interface.

A module which provides the “Settings” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

(continues on next page)

(continued from previous page)

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

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

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

class PyTravisCI.resource_types.settings.**Settings** (**kwargs)

Provides a list of repository settings.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/settings>

Variables settings (List[PyTravisCI.resource_types.setting.Setting]) – A list of setting.

6.35 Stage

Just another Travis CI (API) Python interface.

A module which provides the “Stage” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

(continues on next page)

(continued from previous page)

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

class PyTravisCI.resource_types.stage.Stage (**kwargs)
 Bases: *PyTravisCI.resource_types.base.ResourceTypesBase*

Provides the description of a stage

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/stage>

Variables

- **id** (*int*) – Value uniquely identifying the stage.
- **number** (*int*) – Incremental number for a stage.
- **name** (*str*) – The name of the stage.
- **state** (*str*) – Current state of the stage.
- **started_at** (*datetime*) – When the stage started.
- **finished_at** (*datetime*) – When the stage finished.
- **jobs** (List[PyTravisCI.resource_types.job.Job]) – The jobs of a stage.

6.36 Stages

Just another Travis CI (API) Python interface.

A module which provides the “Stages” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

(continues on next page)

(continued from previous page)

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

```
class PyTravisCI.resource_types.stages.Stages (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides a list of stage.

Official Travis CI API documentation

- <https://developer.travis-ci.org/resource/stages>

Variables `stages` (List[PyTravisCI.resource_types.stage.Stage]) – A list of stage.

6.37 User

Just another Travis CI (API) Python interface.

A module which provides the “User” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.resource_types.user.User (**kwargs)
    Bases: PyTravisCI.resource_types.base.ResourceTypesBase
```

Provides the information of a user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/user>

Variables

- **id** (*int*) – Value uniquely identifying the user.
- **login** (*str*) – Login set on Github.
- **name** (*str*) – Name set on GitHub.
- **github_id** (*int*) – Id set on GitHub.
- **vcs_id** – The user's vcs_id.
- **vcs_type** – The user's vcs_type.
- **avatar_url** (*str*) – Avatar URL set on GitHub.
- **education** (*bool*) – Whether or not the user has an education account.
- **allow_migration** – The user's allow_migration.
- **email** (*str*) – The user's email.
- **is_syncing** (*bool*) – Whether or not the user is currently being synced with Github.
- **synced_at** (*datetime*) – The last time the user was synced with GitHub.
- **recently_signed_up** – The user's recently_signed_up.
- **secure_user_hash** – The user's secure_user_hash.
- **repositories** (List[Repository]) – Repositories belonging to this user.
- **installation** (Installation) – Installation belonging to the user.
- **emails** – The user's emails.

get_active (*, *params: Optional[dict] = None*) → PyTravisCI.resource_types.active.Active
Provides the list of active builds of the current user.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/active>

Parameters **params** – The query parameters to append to the URL.

get_beta_features (*, *params: Optional[dict] = None*) → Py-TravisCI.resource_types.beta_features.BetaFeatures
Provides the list of beta features of the current user.

Official Travis CI API documentation:

- https://developer.travis-ci.org/resource/beta_features

Parameters **params** – The query parameters to append to the URL.

synchronize (*, *params: Optional[dict] = None*) → bool
Triggers a synchronization between Travis CI and the user Github account.

Official Travis CI API documentation:

- <https://developer.travis-ci.org/resource/user>

Parameters `params` – The query parameters to append to the URL.

The encryption module is there to provide a low-level interface for the encryption of data and files for or within the Travis CI infrastructure.

7.1 Data

Just another Travis CI (API) Python interface.

A module which provides the interface for the encryption of data for or within the Travis CI infrastructure.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

(continues on next page)

(continued from previous page)

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
class PyTravisCI.encryption.data.DataEncryption (*, private_key: Union[str, bytes, None]
                                                = None, public_key: Union[str, bytes,
                                                None] = None)
```

Bases: `object`

Provides the interface for the encryption of information for or within the Travis CI infrastructure.

Parameters

- **private_key** – The private key to use for encryption.
- **public_key** – The public key to use for encryption.

decrypt_ensure_key_exists()

Ensures that the key needed for encryption exists.

Raises **ValueError** – When the private key is not set/given.

encrypt_ensure_key_exists()

Ensures that the key needed for encryption exists.

Raises **ValueError** – When the public key is not set/given.

get_private_key() → Optional[bytes]

Provides the currently loaded private key.

get_public_key() → Optional[bytes]

Provides the currently loaded public key.

set_private_key (value: Union[str, bytes], password: Union[str, bytes, None] = None) → None

Sets the private key to use.

Parameters

- **value** – The private key to set.
- **password** – The password associated with the private key (if needed).

set_public_key (value: Union[str, bytes]) → None

Sets the public key to use.

Parameters **value** – The public key to set.

7.2 File

Just another Travis CI (API) Python interface.

A module which provides the interface for the encryption of data for or within the Travis CI infrastructure.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.encryption.file.FileEncryption(*, key: Union[str, bytes, None] =
                                                None, iv: Union[str, bytes, None] =
                                                None)
```

Bases: `object`

Provides the interface for the encryption of files for or within the Travis CI infrastructure.

Parameters

- **key** – The key to use to encrypt the file.
- **iv** – The IV to use to encrypt the file.

decrypt_ensure_keys_exists()

Ensures that the keys exists before any decryption.

encrypt_ensure_keys_exists()

Ensures that the keys exists before any encryption.

generate_iv(*, save: bool = True) → bytes

Generate a new IV key.

Parameters save – Dis-authorize the saving into the global iv attribute.

generate_key(*, save: bool = True) → bytes

Generates a new key.

Parameters save – Dis-authorize the saving into the global key attribute.

get_iv() → Optional[bytes]

Provides the hexadecimal representation of the currently set iv.

get_key() → Optional[bytes]

Provides the hexadecimal representation of the currently set key.

set_iv(value: Union[str, bytes]) → None

Sets the IV to use.

Raises ValueError – When the size of the given key is not correct.

set_key (*value: Union[str, bytes]*) → None

Sets the Key to use.

Raises **ValueError** – When the size of the given key is not correct.

CHAPTER 8

Exceptions

Just another Travis CI (API) Python interface.

A module which provides all our exceptions.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

exception `PyTravisCI.exceptions.BuildAlreadyStarted`
Informs that the current build was already started.

exception `PyTravisCI.exceptions.BuildAlreadyStopped`
Informs that the current build was already stopped.

exception `PyTravisCI.exceptions.FirstPageNotFound`
Informs that the first page could not be found.

exception `PyTravisCI.exceptions.JobAlreadyStarted`
Informs that the current job was already started.

exception `PyTravisCI.exceptions.JobAlreadyStopped`
Informs that the current job was already stopped.

exception `PyTravisCI.exceptions.LastPageNotFound`
Informs that the last page could not be found.

exception `PyTravisCI.exceptions.NextPageNotFound`
Informs that the next page could not be found.

exception `PyTravisCI.exceptions.NotIncomplete`
Informs that the current object is not incomplete.

exception `PyTravisCI.exceptions.PageNotFound`
Raises an exception which informs the end-user that we could not find a page.

exception `PyTravisCI.exceptions.PreviousPageNotFound`
Informs that the previous page could not be found.

exception `PyTravisCI.exceptions.PyTravisCIException`
The base of all our exceptions.

exception `PyTravisCI.exceptions.TravisCIErrror` (*url: str, error_message: str, error_type: str, response: Union[dict, str] = None*)
Raises an exception with the error we got from the API.

get_error_message()
Provides the error message from upstream.

get_error_type()
Provides the error type from upstream.

get_response()
Provides the complete response from Travis CI.

get_url()
Provides the url from upstream.

message() → str
Provides the message to raise.

Standardization

This module provides everything we consider as Standardization before giving back any resource type.

Just another Travis CI (API) Python interface.

A module which provides the standardizer class. It's basically a way to unify everything before parsing them to their end-location or objects.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.standardization.Standardization (data: Optional[Any] = None)
```

The standardizer. It format the response from the Travis CI API into something useful (for later usage and distribution).

```
    get_data () → Any
```

Provides the currently set data.

```
    get_standardized () → dict
```

Provides the standardized version.

```
    run_standardization ()
```

A decorator which actually do the standardization.

```
    set_data (value: Any) → None
```

Overwrites the data to work with.

```
    standardize_at_tagged (data: dict) → dict
```

Standardize the `@tags` to something universal and reusable.

```
    standardize_datetime (data: dict) → dict
```

Provides a way to unify/convert the dates to something more Python friendly.

CHAPTER 10

Requester

Just another Travis CI (API) Python interface.

A module which provides the requester object. It's the object which communicates with the API endpoints.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.requester.Requester
```

```
    Handles all our communication / requests to the Travis CI API.
```

This only provides lower level implementation. Meaning that you give it raw data and it provides you raw data in exchange. While working with this class, you don't have access to enhanced feature provided by something like Resource Types objects.

Warning: You are not expected or invited to use this class directly if you don't know what it may imply!

bind_endpoint_to_base_url (*endpoint: str*) → str

Binds the endpoint with the base url.

delete (*endpoint: str, **kwargs*) → dict

Sends a DELETE requests and returns the response.

get (*endpoint: str, **kwargs*) → dict

Sends a GET requests and returns the response.

classmethod get_error_message (*api_response: dict, *, already_checked: bool = False*) → Optional[str]

Provides the error message.

classmethod get_error_type (*api_response: dict, *, already_checked: bool = False*) → Optional[str]

Provides the error type.

head (*endpoint: str, **kwargs*) → dict

Sends a HEAD requests and returns the response.

static is_error (*api_response: dict*) → bool

Checks if the API response is or contain an error as defined by the Travis CI API documentation.

options (*endpoint: str, **kwargs*) → dict

Sends a OPTIONS requests and returns the response.

patch (*endpoint: str, **kwargs*) → dict

Sends a PATCH requests and returns the response.

post (*endpoint: str, **kwargs*) → dict

Sends a POST requests and returns the response.

put (*endpoint: str, **kwargs*) → dict

Sends a PUT requests and returns the response.

classmethod raise_if_error (*req: requests.models.Request, api_response: dict*) → None

Raises a [TravisCIError](#) if the given API response contain an error.

request_factory ()

A decorator which acts as an universal request factory.

set_authorization (*value: str*) → None

Sets the authorization header.

Raises **TypeError** – If value is not a string.

set_base_url (*value: str*) → None

Sets the base URL we have to communicate with.

Raises **TypeError** – If value is not a string.

CHAPTER 11

Communicator

Communicators are sets of classes composed of methods which (actually) process the communication between us and the Travis CI API.

They are not well documented on purpose because they are not meant for the “grand public”. Unless you have a good reason, you should never (directly) play with them.

Warning: Do not play with those classes unless you know what you are doing!

11.1 Base

Just another Travis CI (API) Python interface.

A module which provides the base of all communicator.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

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

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

class PyTravisCI.communicator.base.**CommunicatorBase** (req: *Py-TravisCI.requester.Requester*)

Provides the base of all communicator.

complete_response ()

A decorator which complete the responses before giving it back to the end-user.

delete_response (endpoint: str) → Union[dict, bool]

DELETE and provides the response from the API.

classmethod encode_slug (slug: str) → str

Encodes the (repository) slug.

endpoints = {}

Should be a dict in format:

```
{
    "from_id": "/build/%(build_id)s"
}
```

from_id is the name of the method and build/%(build_id)s is the endpoint we need to communicate with. %(build_id)s is what the method will expect as variable name. In fact, if the end-user give us build_id=4 as argument, the system will replace %(build_id)s with 4.

filter_before_action ()

A decorator which filter/format the given arguments before giving it back to the communicator method.

This is useful in order to avoid having the same thing over and over everywhere.

get_and_construct_endpoint (kwargs: dict) → str

Provides the endpoint to call from the given method name.

static get_method_name () → str

Provides the method name.

Warning: You are not invited to use this method outside of PyTravisCI's communicators.

get_response (endpoint: str) → dict

Provides the response from the API.

get_standardized (data: dict) → Any

Provides the standardized version of the given dataset.

static is_digit (data: Union[str, int]) → bool

Checks if the given data is an integer or a digit string.

patch_response (*endpoint: str, data: dict = None*) → dict
PATCH and provides the response from the API.

post_response (*endpoint: str, data: dict = None*) → dict
POST and provides the response from the API.

classmethod propagate_internal_vars (*variables: dict, start_obj: Any*) → object
Propagate the given var to all objects.

Warning: You are not invited to use this method outside of PyTravis's communicators.

11.2 Active

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Active” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.active.**Active** (*req: PyTravisCI.requester.Requester*)
Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Active resource type.

endpoints = {'from_github_id': '/owner/github_id/(github_id)s/active', 'from_login':

from_github_id (***kwargs*) → resource_types.Active

from_login (***kwargs*) → resource_types.Active

11.3 Beta feature

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Beta Feature” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.communicator.beta_feature.BetaFeature (req: Py-
TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase
```

The communicator of the Beta feature resource type.

```
delete (**kwargs) → Union[bool, resource_types.BetaFeature]
```

```
endpoints = {'delete': '/user/{user_id}s/beta_feature/{beta_feature_id}s', 'update'
```

```
update (**kwargs) → resource_types.BetaFeature
```

11.4 Beta features

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Beta Features” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.communicator.beta_features.BetaFeatures (req: Py-TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase
    The communicator of the Beta features resource type.
    endpoints = {'from_user_id': 'user/%(user_id)s/beta_features'}
    from_user_id (**kwargs) → resource_types.BetaFeatures
```

11.5 Branch

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Branch” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

The above copyright notice **and** this permission notice shall be included **in all**

(continues on next page)

(continued from previous page)

```
copies or substantial portions of the Software.
```

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

```
class PyTravisCI.communicator.branch.Branch (req: PyTravisCI.requester.Requester)
```

```
    Bases: PyTravisCI.communicator.base.CommunicatorBase
```

```
    The communicator of the Active resource type.
```

```
    endpoints = {'from_id_or_slug': '/repo/%(repository_id_or_slug)s/branch/%(branch_name
```

```
from_id_or_slug (**kwargs) → resource_types.Branch
```

```
from_provider (**kwargs) → resource_types.Branch
```

11.6 Branches

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Branches” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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



```

class PyTravisCI.communicator.branches.Branches (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Branches resource type.

    endpoints = {'from_id_or_slug': '/repo/(repository_id_or_slug)s/branches', 'from_provider': '/provider/(provider_id_or_slug)s/branches'}

    from_id_or_slug (**kwargs) → resource_types.Branches

    from_provider (**kwargs) → resource_types.Branches

```

11.7 Broadcasts

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Broadcasts” resource type.

Author: Nissar Chababy, @funilrys, contactTATAFunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```

class PyTravisCI.communicator.broadcasts.Broadcasts (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Broadcasts resource type.

    endpoints = {'fetch': '/broadcasts'}

    fetch (**kwargs) → resource_types.Broadcasts

```

11.8 Build

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Build” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.communicator.build.Build (req: PyTravisCI.requester.Requester)
```

```
    Bases: PyTravisCI.communicator.base.CommunicatorBase
```

```
    The communicator of the Build resource type.
```

```
    cancel (**kwargs) → resource_types.Build
```

```
    endpoints = {'cancel': '/build/%(build_id)s/cancel', 'from_id': '/build/%(build_id)s
```

```
    from_id (**kwargs) → resource_types.Build
```

```
    restart (**kwargs) → resource_types.Build
```

11.9 Builds

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Builds” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.builds.**Builds** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Builds resource type.

endpoints = {'**fetch**': '/builds', '**from_id_or_slug**': '/repo/%(repository_id_or_slug)s

fetch (**kwargs) → resource_types.Builds

from_id_or_slug (**kwargs) → resource_types.Builds

from_provider (**kwargs) → resource_types.Builds

11.10 Cache

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Cache” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell

(continues on next page)

(continued from previous page)

copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

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

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

class PyTravisCI.communicator.cache.**Cache** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Cache resource type.

delete (***kwargs*) → Union[bool, resource_types.Caches]

endpoints = {'delete': '/repo/%(repository_id_or_slug)s/caches'}

11.11 Caches

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Caches” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```

class PyTravisCI.communicator.caches.Caches (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Caches resource type.

    delete (**kwargs) → Union[bool, resource_types.Caches]

    endpoints = {'delete': '/repo/(repository_id_or_slug)s/caches', 'from_id_or_slug':

    from_id_or_slug (**kwargs) → resource_types.Caches

    from_provider (**kwargs) → resource_types.Caches

```

11.12 Cron

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Cron” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```

class PyTravisCI.communicator.cron.Cron (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Cron resource type.

    create (**kwargs) → resource_types.Cron

    delete (**kwargs) → Union[bool, resource_types.Cron]

    endpoints = {'create': '/repo/(repository_id_or_slug)s/branch/(branch_name)s/cron',

    from_id (**kwargs) → resource_types.Cron

```

```
from_id_or_slug_and_branch (**kwargs) → resource_types.Cron
```

```
from_provider_and_branch (**kwargs) → resource_types.Cron
```

11.13 Crons

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Crons” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.communicator.crons.Crons (req: PyTravisCI.requester.Requester)
```

```
    Bases: PyTravisCI.communicator.base.CommunicatorBase
```

The communicator of the Crons resource type.

```
    endpoints = {'from_id_or_slug': '/repo/(repository_id_or_slug)s/crons', 'from_provid
```

```
    from_id_or_slug (**kwargs) → resource_types.Crons
```

```
    from_provider (**kwargs) → resource_types.Crons
```

11.14 Env Var

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Env Var” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.env_var.**EnvVar** (req: PyTravisCI.requester.Requester)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the EnvVar resource type.

delete (**kwargs) → Union[bool, resource_types.EnvVar]

endpoints = {'delete': '/repo/(repository_id_or_slug)s/env_var/(env_var_id)s', 'from

from_id_or_slug (**kwargs) → resource_types.EnvVar

from_provider (**kwargs) → resource_types.EnvVar

update (**kwargs) → resource_types.EnvVar

11.15 Env Vars

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Env Vars” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

(continues on next page)

(continued from previous page)

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

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

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

class PyTravisCI.communicator.env_vars.**EnvVars** (req: PyTravisCI.requester.Requester)

Bases: PyTravisCI.communicator.base.CommunicatorBase

The communicator of the EnvVars resource type.

create (**kwargs) → resource_types.EnvVar

endpoints = {'create': '/repo/(repository_id_or_slug)s/env_vars', 'from_id_or_slug':

from_id_or_slug (**kwargs) → resource_types.EnvVars

from_provider (**kwargs) → resource_types.EnvVars

11.16 Job

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Job” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

The above copyright notice **and** this permission notice shall be included **in all**

(continues on next page)

(continued from previous page)

copies **or** substantial portions of the Software.

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

class PyTravisCI.communicator.job.**Job** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Job resource type.

cancel (***kwargs*) → resource_types.Job

debug (***kwargs*) → resource_types.Job

endpoints = {'cancel': '/job/(job_id)s/cancel', 'debug': '/job/(job_id)s/debug', 'from_id': '/job/(job_id)s/from_id', 'restart': '/job/(job_id)s/restart'}

from_id (***kwargs*) → resource_types.Job

restart (***kwargs*) → resource_types.Job

11.17 Jobs

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Jobs” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

(continues on next page)

(continued from previous page)

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
class PyTravisCI.communicator.jobs.Jobs (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Jobs resource type.

    endpoints = {'fetch': '/jobs', 'from_build_id': '/build/%(build_id)s/jobs'}

    fetch (**kwargs) → resource_types.Jobs

    from_build_id (**kwargs) → resource_types.Jobs
```

11.18 Key Pair (Generated)

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Key Pair (Generated)” resource type.

Author: Nissar Chababy, @funilrys, contactTATAFunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.communicator.key_pair_generated.KeyPairGenerated (req: Py-
                                                                    TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Key pair generated resource type.

    create (**kwargs) → resource_types.KeyPairGenerated

    endpoints = {'create': '/repo/%(repository_id_or_slug)s/key_pair/generated', 'from_id
```

`from_id_or_slug(**kwargs) → resource_types.KeyPairGenerated`

`from_provider(**kwargs) → resource_types.KeyPairGenerated`

11.19 Key Pair

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Key Pair” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.key_pair.**KeyPair** (req: PyTravisCI.requester.Requester)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Key pair resource type.

create (**kwargs) → resource_types.KeyPair

delete (**kwargs) → Union[Any, resource_types.KeyPair]

endpoints = {'create': '/repo/(repository_id_or_slug)s/key_pair', 'delete': '/repo/

from_id_or_slug (**kwargs) → resource_types.KeyPair

from_provider (**kwargs) → resource_types.KeyPair

update (**kwargs) → resource_types.KeyPair

11.20 Lint

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Lint” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.communicator.lint.Lint (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Lint resource type.

    endpoints = {'fetch': '/lint'}

    fetch (**kwargs) → resource_types.Lint
```

11.21 Log

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Log” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.log.**Log**(req: *PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Log resource type.

delete (**kwargs) → Union[bool, resource_types.Log]

endpoints = {'delete': '/job/(job_id)s/log', 'from_id': '/job/(job_id)s/log'}

from_id (**kwargs) → resource_types.Log

11.22 Log

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Log” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

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

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

class PyTravisCI.communicator.log.**Log** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Log resource type.

delete (***kwargs*) → Union[bool, resource_types.Log]

endpoints = {'delete': '/job/(job_id)s/log', 'from_id': '/job/(job_id)s/log'}

from_id (***kwargs*) → resource_types.Log

11.23 Messages

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Messages” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```

class PyTravisCI.communicator.messages.Messages (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Messages resource type.

    endpoints = {'from_id_or_slug': '/repo/(repository_id_or_slug)s/request/(request_id_or_slug)'
        from_id_or_slug (**kwargs) → resource_types.Messages
        from_provider (**kwargs) → resource_types.Messages

```

11.24 Organization

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Organization” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```

class PyTravisCI.communicator.organization.Organization (req: Py-
    TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Organization resource type.

    endpoints = {'from_id': '/org/(organization_id)s'}
    from_id (**kwargs) → resource_types.Organization

```

11.25 Organizations

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Organizations” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

```
MIT License
```

```
Copyright (c) 2019, 2020 Nissar Chababy
```

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

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

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

```
class PyTravisCI.communicator.organizations.Organizations (req: Py-
TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase
    The communicator of the Organizations resource type.
    endpoints = {'fetch': '/orgs'}
    fetch (**kwargs) → resource_types.Organizations
```

11.26 Repositories

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Repositories” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.repositories.**Repositories** (req: *Py-TravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Repositories resource type.

endpoints = {'fetch': '/repos', 'from_github_id': '/owner/github_id/(github_id)s/re

fetch (**kwargs) → resource_types.Repositories

from_github_id (**kwargs) → resource_types.Repositories

from_login (**kwargs) → resource_types.Repositories

11.27 Repository

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Repository” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

(continues on next page)

(continued from previous page)

furnished to do so, subject to the following conditions:

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

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

```
class PyTravisCI.communicator.repository.Repository (req: Py-
                                                    TravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase
    The communicator of the Active resource type.
    activate (**kwargs) → resource_types.Repository
    deactivate (**kwargs) → resource_types.Repository
    endpoints = {'activate': '/repo/(repository_id_or_slug)s/activate', 'deactivate': '
    from_id_or_slug (**kwargs) → resource_types.Repository
    from_provider (**kwargs) → resource_types.Repository
    star (**kwargs) → resource_types.Repository
    unstar (**kwargs) → resource_types.Repository
```

11.28 Request

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Request” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

The above copyright notice **and** this permission notice shall be included **in all**

(continues on next page)

(continued from previous page)

copies **or** substantial portions of the Software.

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

class PyTravisCI.communicator.request.**Request** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the Request resource type.

endpoints = {'from_id_or_slug': '/repo/%(repository_id_or_slug)s/request/%(request_id

from_id_or_slug (**kwargs) → resource_types.Request

from_provider (**kwargs) → resource_types.Request

11.29 Requests

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Requests” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.communicator.requests.Requests (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Requests resource type.

    create (**kwargs) → resource_types.Request

    endpoints = {'create': '/repo/%(repository_id_or_slug)s/requests', 'from_id_or_slug':
from_id_or_slug (**kwargs) → resource_types.Requests

from_provider (**kwargs) → resource_types.Requests
```

11.30 Stages

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “Stages” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

```
class PyTravisCI.communicator.stages.Stages (req: PyTravisCI.requester.Requester)
    Bases: PyTravisCI.communicator.base.CommunicatorBase

    The communicator of the Stages resource type.

    endpoints = {'from_build_id': '/build/%(build_id)s/stages'}

from_build_id (**kwargs) → resource_types.Stages
```

11.31 User

Just another Travis CI (API) Python interface.

A module which provides the communicator of the “User” resource type.

Author: Nissar Chababy, @funilrys, contactTATAfunilrysTODTODcom

Project link: <https://github.com/funilrys/PyTravisCI>

Project documentation: <https://pytravisci.readthedocs.io/en/latest/>

License

MIT License

Copyright (c) 2019, 2020 Nissar Chababy

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

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

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

class PyTravisCI.communicator.user.**User** (*req: PyTravisCI.requester.Requester*)

Bases: *PyTravisCI.communicator.base.CommunicatorBase*

The communicator of the User resource type.

endpoints = {'**fetch**': '/user', '**from_user_id**': '/user/%(user_id)s', '**synchronize**':

fetch (**kwargs) → resource_types.User

from_user_id (**kwargs) → resource_types.User

synchronize (**kwargs) → resource_types.User

CHAPTER 12

Indices and tables

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

p

PyTravisCI.communicator.active, 87
PyTravisCI.communicator.base, 85
PyTravisCI.communicator.beta_feature, 88
PyTravisCI.communicator.beta_features, 88
PyTravisCI.communicator.branch, 89
PyTravisCI.communicator.branches, 90
PyTravisCI.communicator.broadcasts, 91
PyTravisCI.communicator.build, 92
PyTravisCI.communicator.builds, 92
PyTravisCI.communicator.cache, 93
PyTravisCI.communicator.caches, 94
PyTravisCI.communicator.cron, 95
PyTravisCI.communicator.crons, 96
PyTravisCI.communicator.env_var, 96
PyTravisCI.communicator.env_vars, 97
PyTravisCI.encryption.data, 75
PyTravisCI.encryption.file, 76
PyTravisCI.exceptions, 79
PyTravisCI.requester, 83
PyTravisCI.resource_types.active, 29
PyTravisCI.resource_types.base, 27
PyTravisCI.resource_types.beta_feature, 30
PyTravisCI.resource_types.beta_features, 31
PyTravisCI.resource_types.branch, 32
PyTravisCI.resource_types.branches, 33
PyTravisCI.resource_types.broadcast, 34
PyTravisCI.resource_types.broadcasts, 35
PyTravisCI.resource_types.build, 36
PyTravisCI.resource_types.builds, 38
PyTravisCI.resource_types.cache, 39
PyTravisCI.resource_types.caches, 40
PyTravisCI.resource_types.cron, 41
PyTravisCI.resource_types.crons, 42
PyTravisCI.resource_types.env_var, 43
PyTravisCI.resource_types.env_vars, 45
PyTravisCI.standardization, 81
PyTravisCI.travis_ci, 21

A

activate() (*PyTravisCI.communicator.repository.Repository* method), 108
 activate() (*PyTravisCI.resource_types.repository.Repository* method), 60
 Active (class in *PyTravisCI.communicator.active*), 87
 Active (class in *PyTravisCI.resource_types.active*), 30

B

BetaFeature (class in *PyTravisCI.communicator.beta_feature*), 88
 BetaFeature (class in *PyTravisCI.resource_types.beta_feature*), 31
 BetaFeatures (class in *PyTravisCI.communicator.beta_features*), 89
 BetaFeatures (class in *PyTravisCI.resource_types.beta_features*), 32
 bind_endpoint_to_base_url() (*PyTravisCI.requester.Requester* method), 84
 Branch (class in *PyTravisCI.communicator.branch*), 90
 Branch (class in *PyTravisCI.resource_types.branch*), 33
 Branches (class in *PyTravisCI.communicator.branches*), 90
 Branches (class in *PyTravisCI.resource_types.branches*), 34
 Broadcast (class in *PyTravisCI.resource_types.broadcast*), 35
 Broadcasts (class in *PyTravisCI.communicator.broadcasts*), 91
 Broadcasts (class in *PyTravisCI.resource_types.broadcasts*), 36
 Build (class in *PyTravisCI.communicator.build*), 92
 Build (class in *PyTravisCI.resource_types.build*), 37
 BuildAlreadyStarted, 79
 BuildAlreadyStopped, 79
 Builds (class in *PyTravisCI.communicator.builds*), 93
 Builds (class in *PyTravisCI.resource_types.builds*), 39

C

Cache (class in *PyTravisCI.communicator.cache*), 94
 Cache (class in *PyTravisCI.resource_types.cache*), 40
 Caches (class in *PyTravisCI.communicator.caches*), 94
 Caches (class in *PyTravisCI.resource_types.caches*), 41
 cancel() (*PyTravisCI.communicator.build.Build* method), 92
 cancel() (*PyTravisCI.communicator.job.Job* method), 99
 cancel() (*PyTravisCI.resource_types.build.Build* method), 37
 cancel() (*PyTravisCI.resource_types.job.Job* method), 48
 CommunicatorBase (class in *PyTravisCI.communicator.base*), 86
 complete_response() (*PyTravisCI.communicator.base.CommunicatorBase* method), 86
 ComplexJsonEncoder (class in *PyTravisCI.resource_types.base*), 28
 create() (*PyTravisCI.communicator.cron.Cron* method), 95
 create() (*PyTravisCI.communicator.env_vars.EnvVars* method), 98
 create() (*PyTravisCI.communicator.key_pair.KeyPair* method), 101
 create() (*PyTravisCI.communicator.key_pair_generated.KeyPairGenerated* method), 100
 create() (*PyTravisCI.communicator.requests.Requests* method), 110
 create_cron() (*PyTravisCI.resource_types.branch.Branch* method), 33
 create_env_var() (*PyTravisCI.resource_types.repository.Repository* method), 60
 create_key_pair() (*PyTravisCI.resource_types.repository.Repository* method), 60

`create_request()` (*PyTravisCI.resource_types.repository.Repository method*), 60

`Cron` (*class in PyTravisCI.communicator.cron*), 95

`Cron` (*class in PyTravisCI.resource_types.cron*), 42

`Crons` (*class in PyTravisCI.communicator.crons*), 96

`Crons` (*class in PyTravisCI.resource_types.crons*), 43

D

`DataEncryption` (*class in PyTravisCI.encryption.data*), 76

`deactivate()` (*PyTravisCI.communicator.repository.Repository method*), 108

`deactivate()` (*PyTravisCI.resource_types.repository.Repository method*), 61

`debug()` (*PyTravisCI.communicator.job.Job method*), 99

`debug()` (*PyTravisCI.resource_types.job.Job method*), 48

`decrypt_ensure_key_exists()` (*PyTravisCI.encryption.data.DataEncryption method*), 76

`decrypt_ensure_keys_exists()` (*PyTravisCI.encryption.file.FileEncryption method*), 77

`default()` (*PyTravisCI.resource_types.base.ComplexJsonEncoder method*), 28

`delete()` (*PyTravisCI.communicator.beta_feature.BetaFeature method*), 88

`delete()` (*PyTravisCI.communicator.cache.Cache method*), 94

`delete()` (*PyTravisCI.communicator.caches.Caches method*), 95

`delete()` (*PyTravisCI.communicator.cron.Cron method*), 95

`delete()` (*PyTravisCI.communicator.env_var.EnvVar method*), 97

`delete()` (*PyTravisCI.communicator.key_pair.KeyPair method*), 101

`delete()` (*PyTravisCI.communicator.log.Log method*), 103, 104

`delete()` (*PyTravisCI.requester.Requester method*), 84

`delete()` (*PyTravisCI.resource_types.beta_feature.BetaFeature method*), 31

`delete()` (*PyTravisCI.resource_types.cache.Cache method*), 40

`delete()` (*PyTravisCI.resource_types.caches.Caches method*), 41

`delete()` (*PyTravisCI.resource_types.cron.Cron method*), 42

`delete()` (*PyTravisCI.resource_types.env_var.EnvVar method*), 44

`delete()` (*PyTravisCI.resource_types.key_pair.KeyPair method*), 51

`delete()` (*PyTravisCI.resource_types.log.Log method*), 53

`delete_response()` (*PyTravisCI.communicator.base.CommunicatorBase method*), 86

`dict()` (*PyTravisCI.resource_types.base.ResourceTypesBase method*), 28

`disable()` (*PyTravisCI.resource_types.beta_feature.BetaFeature method*), 31

E

`enable()` (*PyTravisCI.resource_types.beta_feature.BetaFeature method*), 31

`enable_slug()` (*PyTravisCI.communicator.base.CommunicatorBase class method*), 86

`encrypt_ensure_key_exists()` (*PyTravisCI.encryption.data.DataEncryption method*), 76

`encrypt_ensure_keys_exists()` (*PyTravisCI.encryption.file.FileEncryption method*), 77

`encrypt_env_var()` (*PyTravisCI.resource_types.repository.Repository method*), 61

`encrypt_file()` (*PyTravisCI.resource_types.repository.Repository method*), 61

`encrypt_secrets()` (*PyTravisCI.resource_types.repository.Repository method*), 62

`endpoints` (*PyTravisCI.communicator.active.Active attribute*), 87

`endpoints` (*PyTravisCI.communicator.base.CommunicatorBase attribute*), 86

`endpoints` (*PyTravisCI.communicator.beta_feature.BetaFeature attribute*), 88

`endpoints` (*PyTravisCI.communicator.beta_features.BetaFeatures attribute*), 89

`endpoints` (*PyTravisCI.communicator.branch.Branch attribute*), 90

`endpoints` (*PyTravisCI.communicator.branches.Branches attribute*), 91

`endpoints` (*PyTravisCI.communicator.broadcasts.Broadcasts attribute*), 91

`endpoints` (*PyTravisCI.communicator.build.Build attribute*), 92

`endpoints` (*PyTravisCI.communicator.builds.Builds attribute*), 93

`endpoints` (*PyTravisCI.communicator.cache.Cache attribute*), 94

`endpoints` (*PyTravisCI.communicator.caches.Caches attribute*), 95

endpoints (PyTravisCI.communicator.cron.Cron attribute), 95
 endpoints (PyTravisCI.communicator.crons.Crons attribute), 96
 endpoints (PyTravisCI.communicator.env_var.EnvVar attribute), 97
 endpoints (PyTravisCI.communicator.env_vars.EnvVars attribute), 98
 endpoints (PyTravisCI.communicator.job.Job attribute), 99
 endpoints (PyTravisCI.communicator.jobs.Jobs attribute), 100
 endpoints (PyTravisCI.communicator.key_pair.KeyPair attribute), 101
 endpoints (PyTravisCI.communicator.key_pair_generated.KeyPairGenerated attribute), 100
 endpoints (PyTravisCI.communicator.lint.Lint attribute), 102
 endpoints (PyTravisCI.communicator.log.Log attribute), 103, 104
 endpoints (PyTravisCI.communicator.messages.Messages attribute), 105
 endpoints (PyTravisCI.communicator.organization.Organization attribute), 105
 endpoints (PyTravisCI.communicator.organizations.Organizations attribute), 106
 endpoints (PyTravisCI.communicator.repositories.Repositories attribute), 107
 endpoints (PyTravisCI.communicator.repository.Repository attribute), 108
 endpoints (PyTravisCI.communicator.request.Request attribute), 109
 endpoints (PyTravisCI.communicator.requests.Requests attribute), 110
 endpoints (PyTravisCI.communicator.stages.Stages attribute), 110
 endpoints (PyTravisCI.communicator.user.User attribute), 111
 EnvVar (class in PyTravisCI.communicator.env_var), 97
 EnvVar (class in PyTravisCI.resource_types.env_var), 44
 EnvVars (class in PyTravisCI.communicator.env_vars), 98
 EnvVars (class in PyTravisCI.resource_types.env_vars), 45
 FileEncryption (class in PyTravisCI.encryption.file), 77
 filter_before_action() (PyTravisCI.communicator.base.CommunicatorBase method), 86
 first_page() (PyTravisCI.resource_types.base.ResourceTypesBase method), 80
 FirstPageNotFound, 80
 from_build_id() (PyTravisCI.communicator.jobs.Jobs method), 100
 from_build_id() (PyTravisCI.communicator.stages.Stages method), 110
 from_github_id() (PyTravisCI.communicator.active.Active method), 87
 from_github_id() (PyTravisCI.communicator.repositories.Repositories method), 107
 from_id() (PyTravisCI.communicator.build.Build method), 92
 from_id() (PyTravisCI.communicator.cron.Cron method), 95
 from_id() (PyTravisCI.communicator.job.Job method), 99
 from_id() (PyTravisCI.communicator.log.Log method), 103, 104
 from_id() (PyTravisCI.communicator.organization.Organization method), 105
 from_id_or_slug() (PyTravisCI.communicator.branch.Branch method), 90
 from_id_or_slug() (PyTravisCI.communicator.branches.Branches method), 91
 from_id_or_slug() (PyTravisCI.communicator.builds.Builds method), 93
 from_id_or_slug() (PyTravisCI.communicator.caches.Caches method), 95
 from_id_or_slug() (PyTravisCI.communicator.crons.Crons method), 96
 from_id_or_slug() (PyTravisCI.communicator.jobs.Jobs method), 100

F

<i>TravisCI.communicator.env_var.EnvVar</i> method), 97	<i>TravisCI.communicator.key_pair_generated.KeyPairGenerated</i> method), 101
from_id_or_slug() (Py- <i>TravisCI.communicator.env_vars.EnvVars</i> method), 98	from_provider() (Py- <i>TravisCI.communicator.messages.Messages</i> method), 105
from_id_or_slug() (Py- <i>TravisCI.communicator.key_pair.KeyPair</i> method), 101	from_provider() (Py- <i>TravisCI.communicator.repository.Repository</i> method), 108
from_id_or_slug() (Py- <i>TravisCI.communicator.key_pair_generated.KeyPairGenerated</i> method), 100	from_provider() (Py- <i>TravisCI.communicator.request.Request</i> method), 109
from_id_or_slug() (Py- <i>TravisCI.communicator.messages.Messages</i> method), 105	from_provider() (Py- <i>TravisCI.communicator.requests.Requests</i> method), 110
from_id_or_slug() (Py- <i>TravisCI.communicator.repository.Repository</i> method), 108	from_provider_and_branch() (Py- <i>TravisCI.communicator.cron.Cron</i> method), 96
from_id_or_slug() (Py- <i>TravisCI.communicator.request.Request</i> method), 109	from_user_id() (Py- <i>TravisCI.communicator.beta_features.BetaFeatures</i> method), 89
from_id_or_slug() (Py- <i>TravisCI.communicator.requests.Requests</i> method), 110	from_user_id() (Py- <i>TravisCI.communicator.user.User</i> method), 111
from_id_or_slug_and_branch() (Py- <i>TravisCI.communicator.cron.Cron</i> method), 95	
from_login() (PyTravisCI.communicator.active.Active method), 87	G
from_login() (PyTravisCI.communicator.repositories.Repositories method), 107	generate_iv() (Py- <i>TravisCI.encryption.file.FileEncryption</i> method), 77
from_provider() (Py- <i>TravisCI.communicator.branch.Branch</i> method), 90	generate_key() (Py- <i>TravisCI.encryption.file.FileEncryption</i> method), 77
from_provider() (Py- <i>TravisCI.communicator.branches.Branches</i> method), 91	generate_key_pair() (Py- <i>TravisCI.resource_types.repository.Repository</i> method), 63
from_provider() (Py- <i>TravisCI.communicator.builds.Builds</i> method), 93	get() (PyTravisCI.requester.Requester method), 84
from_provider() (Py- <i>TravisCI.communicator.caches.Caches</i> method), 95	get_access_point() (Py- <i>TravisCI.travis_ci.TravisCI</i> method), 22
from_provider() (Py- <i>TravisCI.communicator.crons.Crons</i> method), 96	get_active() (PyTravisCI.resource_types.organization.Organization method), 56
from_provider() (Py- <i>TravisCI.communicator.env_var.EnvVar</i> method), 97	get_active() (PyTravisCI.resource_types.user.User method), 72
from_provider() (Py- <i>TravisCI.communicator.env_vars.EnvVars</i> method), 98	get_active_from_github_id() (Py- <i>TravisCI.travis_ci.TravisCI</i> method), 22
from_provider() (Py- <i>TravisCI.communicator.key_pair.KeyPair</i> method), 101	get_active_from_login() (Py- <i>TravisCI.travis_ci.TravisCI</i> method), 22
	get_and_construct_endpoint() (Py- <i>TravisCI.communicator.base.CommunicatorBase</i> method), 86
	get_beta_features() (Py- <i>TravisCI.resource_types.user.User</i> method), 72
	get_branch() (PyTravisCI.resource_types.repository.Repository method), 63

<code>get_branches()</code>	(Py-TravisCI.resource_types.repository.Repository method), 63	<code>TravisCI.resource_types.repository.Repository</code> method), 64
<code>get_broadcasts()</code>	(PyTravisCI.travis_ci.TravisCI method), 22	<code>get_log()</code> (PyTravisCI.resource_types.job.Job method), 48
<code>get_build()</code>	(PyTravisCI.travis_ci.TravisCI method), 22	<code>get_method_name()</code> (Py-TravisCI.communicator.base.CommunicatorBase static method), 86
<code>get_builds()</code>	(PyTravisCI.resource_types.repository.Repository method), 63	<code>get_organization()</code> (Py-TravisCI.travis_ci.TravisCI method), 23
<code>get_builds()</code>	(PyTravisCI.travis_ci.TravisCI method), 22	<code>get_organizations()</code> (Py-TravisCI.travis_ci.TravisCI method), 23
<code>get_caches()</code>	(PyTravisCI.resource_types.repository.Repository method), 63	<code>get_private_key()</code> (Py-TravisCI.encryption.data.DataEncryption method), 76
<code>get_complete()</code>	(Py-TravisCI.resource_types.base.ResourceTypesBase method), 28	<code>get_public_key()</code> (Py-TravisCI.encryption.data.DataEncryption method), 76
<code>get_cron()</code>	(PyTravisCI.resource_types.branch.Branch method), 33	<code>get_repositories()</code> (Py-TravisCI.travis_ci.TravisCI method), 24
<code>get_cron()</code>	(PyTravisCI.travis_ci.TravisCI method), 23	<code>get_repositories_from_github_id()</code> (Py-TravisCI.travis_ci.TravisCI method), 24
<code>get_crons()</code>	(PyTravisCI.resource_types.repository.Repository method), 63	<code>get_repositories_from_login()</code> (Py-TravisCI.travis_ci.TravisCI method), 24
<code>get_data()</code>	(PyTravisCI.standardization.Standardization method), 82	<code>get_repository()</code> (PyTravisCI.travis_ci.TravisCI method), 24
<code>get_env_var()</code>	(Py-TravisCI.resource_types.repository.Repository method), 64	<code>get_repository_from_provider()</code> (Py-TravisCI.travis_ci.TravisCI method), 24
<code>get_env_vars()</code>	(Py-TravisCI.resource_types.repository.Repository method), 64	<code>get_request()</code> (Py-TravisCI.resource_types.repository.Repository method), 64
<code>get_error_message()</code>	(Py-TravisCI.exceptions.TravisCIErrors method), 80	<code>get_requests()</code> (Py-TravisCI.resource_types.repository.Repository method), 64
<code>get_error_message()</code>	(Py-TravisCI.requester.Requester class method), 84	<code>get_response()</code> (Py-TravisCI.communicator.base.CommunicatorBase method), 86
<code>get_error_type()</code>	(Py-TravisCI.exceptions.TravisCIErrors method), 80	<code>get_response()</code> (Py-TravisCI.exceptions.TravisCIErrors method), 80
<code>get_error_type()</code>	(PyTravisCI.requester.Requester class method), 84	<code>get_setting()</code> (Py-TravisCI.resource_types.repository.Repository method), 65
<code>get_iv()</code>	(PyTravisCI.encryption.file.FileEncryption method), 77	<code>get_settings()</code> (Py-TravisCI.resource_types.repository.Repository method), 65
<code>get_job()</code>	(PyTravisCI.travis_ci.TravisCI method), 23	<code>get_stages()</code> (PyTravisCI.resource_types.build.Build method), 38
<code>get_jobs()</code>	(PyTravisCI.resource_types.build.Build method), 37	<code>get_standardized()</code> (Py-TravisCI.communicator.base.CommunicatorBase method), 86
<code>get_jobs()</code>	(PyTravisCI.travis_ci.TravisCI method), 23	<code>get_standardized()</code> (Py-TravisCI.standardization.Standardization method), 82
<code>get_key()</code>	(PyTravisCI.encryption.file.FileEncryption method), 77	<code>get_url()</code> (PyTravisCI.exceptions.TravisCIErrors method), 82
<code>get_key_pair()</code>	(Py-TravisCI.resource_types.repository.Repository method), 64	
<code>get_key_pair_generated()</code>	(Py-	

method), 80
get_user() (PyTravisCI.travis_ci.TravisCI method), 25
get_user_from_id() (Py-TravisCI.travis_ci.TravisCI method), 25

H

has_first_page() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 28
has_last_page() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 28
has_next_page() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 29
has_previous_page() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 29
head() (PyTravisCI.requester.Requester method), 84

I

Installation (class in Py-TravisCI.resource_types.installation), 46
is_active() (PyTravisCI.resource_types.build.Build method), 38
is_active() (PyTravisCI.resource_types.job.Job method), 48
is_canceled() (Py-TravisCI.resource_types.build.Build method), 38
is_canceled() (PyTravisCI.resource_types.job.Job method), 48
is_created() (PyTravisCI.resource_types.build.Build method), 38
is_created() (PyTravisCI.resource_types.job.Job method), 48
is_digit() (PyTravisCI.communicator.base.CommunicatorBase static method), 86
is_error() (PyTravisCI.requester.Requester static method), 84
is_errored() (PyTravisCI.resource_types.build.Build method), 38
is_errored() (PyTravisCI.resource_types.job.Job method), 48
is_failed() (PyTravisCI.resource_types.build.Build method), 38
is_failed() (PyTravisCI.resource_types.job.Job method), 48
is_incomplete() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 29
is_passed() (PyTravisCI.resource_types.build.Build method), 38

is_passed() (PyTravisCI.resource_types.job.Job method), 48
is_started() (PyTravisCI.resource_types.build.Build method), 38
is_started() (PyTravisCI.resource_types.job.Job method), 48

J

Job (class in PyTravisCI.communicator.job), 99
Job (class in PyTravisCI.resource_types.job), 47
JobAlreadyStarted, 80
JobAlreadyStopped, 80
Jobs (class in PyTravisCI.communicator.jobs), 100
Jobs (class in PyTravisCI.resource_types.jobs), 49
json() (PyTravisCI.resource_types.base.ResourceTypesBase method), 29

K

KeyPair (class in PyTravisCI.communicator.key_pair), 101
KeyPair (class in PyTravisCI.resource_types.key_pair), 51
KeyPairGenerated (class in Py-TravisCI.communicator.key_pair_generated), 100
KeyPairGenerated (class in Py-TravisCI.resource_types.key_pair_generated), 50

L

last_page() (PyTravisCI.resource_types.base.ResourceTypesBase method), 29
LastPageNotFound, 80
Lint (class in PyTravisCI.communicator.lint), 102
Lint (class in PyTravisCI.resource_types.lint), 52
lint() (PyTravisCI.travis_ci.TravisCI method), 25
LintWarning (class in Py-TravisCI.resource_types.lint), 52
Log (class in PyTravisCI.communicator.log), 103, 104
Log (class in PyTravisCI.resource_types.log), 53

M

make_private() (Py-TravisCI.resource_types.env_var.EnvVar method), 44
make_public() (Py-TravisCI.resource_types.env_var.EnvVar method), 44
Message (class in PyTravisCI.resource_types.message), 54
message() (PyTravisCI.exceptions.TravisCIError method), 80
Messages (class in Py-TravisCI.communicator.messages), 104

Messages (class in Py-TravisCI.resource_types.messages), 55

N

next_page() (PyTravisCI.resource_types.base.ResourceTypesBase method), 29

NextPageNotFound, 80

NotIncomplete, 80

O

options() (PyTravisCI.requester.Requester method), 84

Organization (class in Py-TravisCI.communicator.organization), 105

Organization (class in Py-TravisCI.resource_types.organization), 56

Organizations (class in Py-TravisCI.communicator.organizations), 106

Organizations (class in Py-TravisCI.resource_types.organizations), 57

P

PageNotFound, 80

patch() (PyTravisCI.requester.Requester method), 84

patch_response() (Py-TravisCI.communicator.base.CommunicatorBase method), 86

post() (PyTravisCI.requester.Requester method), 84

post_response() (Py-TravisCI.communicator.base.CommunicatorBase method), 87

previous_page() (Py-TravisCI.resource_types.base.ResourceTypesBase method), 29

PreviousPageNotFound, 80

propagate_internal_vars() (Py-TravisCI.communicator.base.CommunicatorBase class method), 87

put() (PyTravisCI.requester.Requester method), 84

PyTravisCI.communicator.active (module), 87

PyTravisCI.communicator.base (module), 85

PyTravisCI.communicator.beta_feature (module), 88

PyTravisCI.communicator.beta_features (module), 88

PyTravisCI.communicator.branch (module), 89

PyTravisCI.communicator.branches (module), 90

PyTravisCI.communicator.broadcasts (module), 91

PyTravisCI.communicator.build (module), 92

PyTravisCI.communicator.builds (module), 92

PyTravisCI.communicator.cache (module), 93

PyTravisCI.communicator.caches (module), 93

PyTravisCI.communicator.cron (module), 95

PyTravisCI.communicator.crons (module), 96

PyTravisCI.communicator.env_var (module), 96

PyTravisCI.communicator.env_vars (module), 97

PyTravisCI.communicator.job (module), 98

PyTravisCI.communicator.jobs (module), 99

PyTravisCI.communicator.key_pair (module), 101

PyTravisCI.communicator.key_pair_generated (module), 100

PyTravisCI.communicator.lint (module), 102

PyTravisCI.communicator.log (module), 102, 103

PyTravisCI.communicator.messages (module), 104

PyTravisCI.communicator.organization (module), 105

PyTravisCI.communicator.organizations (module), 106

PyTravisCI.communicator.repositories (module), 106

PyTravisCI.communicator.repository (module), 107

PyTravisCI.communicator.request (module), 108

PyTravisCI.communicator.requests (module), 109

PyTravisCI.communicator.stages (module), 110

PyTravisCI.communicator.user (module), 111

PyTravisCI.encryption.data (module), 75

PyTravisCI.encryption.file (module), 76

PyTravisCI.exceptions (module), 79

PyTravisCI.requester (module), 83

PyTravisCI.resource_types.active (module), 29

PyTravisCI.resource_types.base (module), 27

PyTravisCI.resource_types.beta_feature (module), 30

PyTravisCI.resource_types.beta_features (module), 31

PyTravisCI.resource_types.branch (module), 32

PyTravisCI.resource_types.branches (module), 33

PyTravisCI.resource_types.broadcast

(*module*), 34
PyTravisCI.resource_types.broadcasts (*module*), 35
PyTravisCI.resource_types.build (*module*), 36
PyTravisCI.resource_types.builds (*module*), 38
PyTravisCI.resource_types.cache (*module*), 39
PyTravisCI.resource_types.caches (*module*), 40
PyTravisCI.resource_types.cron (*module*), 41
PyTravisCI.resource_types.crons (*module*), 42
PyTravisCI.resource_types.env_var (*module*), 43
PyTravisCI.resource_types.env_vars (*module*), 45
PyTravisCI.resource_types.installation (*module*), 46
PyTravisCI.resource_types.job (*module*), 47
PyTravisCI.resource_types.jobs (*module*), 49
PyTravisCI.resource_types.key_pair (*module*), 50
PyTravisCI.resource_types.key_pair_generated (*module*), 49
PyTravisCI.resource_types.lint (*module*), 52
PyTravisCI.resource_types.log (*module*), 53
PyTravisCI.resource_types.message (*module*), 54
PyTravisCI.resource_types.messages (*module*), 55
PyTravisCI.resource_types.organization (*module*), 55
PyTravisCI.resource_types.organizations (*module*), 57
PyTravisCI.resource_types.repositories (*module*), 58
PyTravisCI.resource_types.repository (*module*), 58
PyTravisCI.resource_types.request (*module*), 65
PyTravisCI.resource_types.requests (*module*), 67
PyTravisCI.resource_types.setting (*module*), 67
PyTravisCI.resource_types.settings (*module*), 68
PyTravisCI.resource_types.stage (*module*), 69
PyTravisCI.resource_types.stages (*module*), 70
PyTravisCI.resource_types.user (*module*), 71
PyTravisCI.standardization (*module*), 81
PyTravisCI.travis_ci (*module*), 21
PyTravisCIException, 80

R

raise_if_error() (*PyTravisCI.requester.Requester class method*), 84
regenerate() (*PyTravisCI.resource_types.key_pair_generated.KeyPair method*), 50
Repositories (*class in Py-TravisCI.communicator.repositories*), 107
Repositories (*class in Py-TravisCI.resource_types.repositories*), 58
Repository (*class in Py-TravisCI.communicator.repository*), 108
Repository (*class in Py-TravisCI.resource_types.repository*), 59
Request (*class in PyTravisCI.communicator.request*), 109
Request (*class in PyTravisCI.resource_types.request*), 66
request_factory() (*Py-TravisCI.requester.Requester method*), 84
Requester (*class in PyTravisCI.requester*), 83
Requests (*class in Py-TravisCI.communicator.requests*), 109
Requests (*class in Py-TravisCI.resource_types.requests*), 67
ResourceTypesBase (*class in Py-TravisCI.resource_types.base*), 28
restart() (*PyTravisCI.communicator.build.Build method*), 92
restart() (*PyTravisCI.communicator.job.Job method*), 99
restart() (*PyTravisCI.resource_types.build.Build method*), 38
restart() (*PyTravisCI.resource_types.job.Job method*), 48
run_standardization() (*Py-TravisCI.standardization.Standardization method*), 82

S

set_access_point() (*Py-TravisCI.travis_ci.TravisCI method*), 25
set_access_token() (*Py-TravisCI.travis_ci.TravisCI method*), 25
set_authorization() (*Py-TravisCI.requester.Requester method*), 84
set_base_url() (*PyTravisCI.requester.Requester method*), 84

set_branch() (PyTravisCI.resource_types.env_var.EnvVar
 method), 44
 set_data() (PyTravisCI.standardization.Standardization
 method), 82
 set_description() (Py-
 TravisCI.resource_types.key_pair.KeyPair
 method), 51
 set_iv() (PyTravisCI.encryption.file.FileEncryption
 method), 77
 set_key() (PyTravisCI.encryption.file.FileEncryption
 method), 77
 set_name() (PyTravisCI.resource_types.env_var.EnvVar
 method), 44
 set_private_key() (Py-
 TravisCI.encryption.data.DataEncryption
 method), 76
 set_public_key() (Py-
 TravisCI.encryption.data.DataEncryption
 method), 76
 set_value() (PyTravisCI.resource_types.env_var.EnvVar
 method), 45
 set_value() (PyTravisCI.resource_types.key_pair.KeyPair
 method), 52
 set_value() (PyTravisCI.resource_types.setting.Setting
 method), 68
 Setting (class in PyTravisCI.resource_types.setting),
 68
 Settings (class in Py-
 TravisCI.resource_types.settings), 69
 Stage (class in PyTravisCI.resource_types.stage), 70
 Stages (class in PyTravisCI.communicator.stages), 110
 Stages (class in PyTravisCI.resource_types.stages), 71
 Standardization (class in Py-
 TravisCI.standardization), 81
 standardize_at_tagged() (Py-
 TravisCI.standardization.Standardization
 method), 82
 standardize_datetime() (Py-
 TravisCI.standardization.Standardization
 method), 82
 star() (PyTravisCI.communicator.repository.Repository
 method), 108
 star() (PyTravisCI.resource_types.repository.Repository
 method), 65
 sync() (PyTravisCI.resource_types.build.Build
 method), 38
 sync() (PyTravisCI.resource_types.job.Job method),
 49
 synchronize() (PyTravisCI.communicator.user.User
 method), 111
 synchronize() (Py-
 TravisCI.resource_types.user.User method),
 72
 to_dict() (PyTravisCI.resource_types.base.ResourceTypesBase
 method), 29
 to_json() (PyTravisCI.resource_types.base.ResourceTypesBase
 method), 29
 TravisCI (class in PyTravisCI.travis_ci), 21
 TravisCIError, 80

U

unstar() (PyTravisCI.communicator.repository.Repository
 method), 108
 unstar() (PyTravisCI.resource_types.repository.Repository
 method), 65
 update() (PyTravisCI.communicator.beta_feature.BetaFeature
 method), 88
 update() (PyTravisCI.communicator.env_var.EnvVar
 method), 97
 update() (PyTravisCI.communicator.key_pair.KeyPair
 method), 101
 User (class in PyTravisCI.communicator.user), 111
 User (class in PyTravisCI.resource_types.user), 71