

---

# litmos-api

*Release 1.3.0*

Nov 04, 2022



---

## Contents

---

<b>1</b>	<b>Litmos API</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Getting started . . . . .	1
1.3	Documentation . . . . .	4
1.4	Development . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Reference</b>	<b>9</b>
4.1	litmos_api . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Bug reports . . . . .	11
5.2	Documentation improvements . . . . .	11
5.3	Feature requests and feedback . . . . .	11
5.4	Development . . . . .	12
<b>6</b>	<b>Authors</b>	<b>13</b>
<b>7</b>	<b>Changelog</b>	<b>15</b>
7.1	1.0.0 (2020-07-30) . . . . .	15
7.2	0.1.1 (2019-08-19) . . . . .	15
7.3	0.1.0 (2016-12-07) . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>



# CHAPTER 1

---

## Litmos API

---

Litmos REST API client for python 3.6 and above

- Free software: BSD license

## 1.1 Installation

```
pip install litmos-api
```

## 1.2 Getting started

```
from litmos import Litmos

API_KEY = 'XXXXXXXXXXXX'
LITMOS_APP_NAME = 'jins.litmos.com'
LITMOS_SERVER_URL = 'https://api.litmos.com/v1.svc' # https://support.litmos.com/hc/
↳ en-us/articles/227734667-Overview-Developer-API
```

(continues on next page)

(continued from previous page)

```

litmos = Litmos(API_KEY, LITMOS_APP_NAME, LITMOS_SERVER_URL)

# --- User ---
# retrieve users
all_users = litmos.User.all()

# retrieve all users (with all information populated - default /users/all API_
↳ endpoint only returns a subset of user fields)
# much longer than .all() as individual requests to /find/{user-id} for every user_
↳ are made
all_users_with_full_details = litmos.User.all(True)

# find user by Id
user = litmos.User.find('rnjx2WaQOa11')

# search for user by username
user = litmos.User.search('beelzebub@pieshop.net')

# update JobTitle & City fields
user.JobTitle = 'Pie eater'
user.City = 'Pieland'

# save user
user.save()

# deactivate user
user.deactivate()

# create user
user = litmos.User.create({
    'UserName': 'jobaba72@pieshop.net',
    'FirstName': 'Jo',
    'LastName': 'Baba72',
    'Email': 'jobaba72@pieshop.net'
})

# get teams for a user
user.teams()

# remove all teams from user
user.remove_teams()

# delete user
# with Id
litmos.User.delete('YmrD112q1m41')

# instance delete
user.destroy()

# set a user's manager
user.set_manager('jdhasdhl1sa')

# or pass in a user object
manager = litmos.User.find('rnjx2WaQOa11')[0]
user.set_manager(manager)

```

(continues on next page)

(continued from previous page)

```

# advanced custom fields
# https://support.litmos.com/hc/en-us/articles/227735427-User-Custom-Fields
user.update_advanced_custom_fields(
    [
        {"fieldname": "value1"},
        {"fieldname": "value2"}
    ]
)

# --- Team ---
# get all teams
all_teams = litmos.Team.all()

# find team by Id
team = litmos.Team.find('rnjx2WaQOa11')

# get team members
users = team.users()

# get team leaders
leaders = team.leaders()

# get team admins
admins = team.admins()

# create team (at root level)
team = litmos.Team.create({'Name': 'A-Team', 'Description': 'I pity the fool!'})

# add sub-team
sub_team = litmos.Team()
sub_team.Name = 'B-Team'
sub_team.Description = 'Woohoo'

sub_team_id = team.add_sub_team(sub_team)

# assign courses to team
course1 = litmos.Course.find('d2cJSDvIU0c1')
course2 = litmos.Course.find('d2cJSDvIU0c2')
team.assign_courses([course1, course2])

# unassign courses to team
course1 = litmos.Course.find('d2cJSDvIU0c1')
course2 = litmos.Course.find('d2cJSDvIU0c2')
team.unassign_courses([course1, course2])

# --- Team members ---

# add users to team
user1 = litmos.User.find('rnjx2WaQOa11')
user2 = litmos.User.find('rnjx2WaQOa12')
team.add_users([user1, user2])

# remove users from team
team.remove_user(user2)

# --- Team leaders ---
# promote user

```

(continues on next page)

(continued from previous page)

```
team.promote_team_leader(user1)

# demote user
team.demote_team_leader(user1)

# --- Team admins ---
# promote user
team.promote_team_admin(user1)

# demote user
team.demote_team_admin(user1)
```

## 1.3 Documentation

<https://python-litmos-api.readthedocs.io/>

## 1.4 Development

To run the all tests run:

```
pip install -r requirements.txt
cd tests
nosetests
```



## CHAPTER 2

---

### Installation

---

At the command line:

```
pip install litmos-api
```



## CHAPTER 3

---

### Usage

---

To use litmos-api in a project:

```
import litmos_api
```



## CHAPTER 4

---

### Reference

---

#### 4.1 litmos\_api



Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

### 5.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 5.2 Documentation improvements

litmos-api could always use more documentation, whether as part of the official litmos-api docs, in docstrings, or even on the web in blog posts, articles, and such.

### 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/charliequinn/python-litmos-api/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

## 5.4 Development

To set up *python-litmos-api* for local development:

1. Fork [python-litmos-api](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/python-litmos-api.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .  
git commit -m "Your detailed description of your changes."  
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)<sup>1</sup>.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

---

<sup>1</sup> If you don’t have all the necessary python versions available locally you can rely on Travis - it will [run the tests](#) for each change you add in the pull request.  
It will be slower though ...



## CHAPTER 6

---

### Authors

---

- Charlie Quinn
- Ignacio Bolonio
- Jeremy Knickerbocker



# CHAPTER 7

---

## Changelog

---

### 7.1 1.0.0 (2020-07-30)

- Add ability to update a user's manager via the manager ID or passing in a user object
- Add ability to handle advanced custom fields on User object <https://support.litmos.com/hc/en-us/articles/227735427-User-Custom-Fields>

### 7.2 0.1.1 (2019-08-19)

- Add assign and unassign courses methods to Team class.

### 7.3 0.1.0 (2016-12-07)

- First release on PyPI.



## CHAPTER 8

---

### Indices and tables

---

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